Inference Server

Release Notes

Table of Contents

Chapter 1. Triton Inference Server Overview

The NVIDIA® Triton™ Inference Server provides a cloud and edge inferencing solution that is optimized for CPUs and GPUs. Triton supports an HTTP/REST and GRPC protocol that allows remote clients to request inferencing for any model that is being managed by the server. For edge deployments, Triton is available as a shared library with a C API that allows the complete Triton functionality to be included directly in an application.

The Triton Inference Server is in the Triton Inference Server container. There are additional C++ and Python client libraries that are external to the container, and you can find additional documentation at GitHub: Inference Server.

This document describes the key features, software enhancements and improvements, known issues, and how to run this container.

Chapter 2. Pulling A Container

About this task

Before you can pull a container from the NGC container registry:

- Install Docker.
  - For NVIDIA DGX™ users, see Preparing to use NVIDIA Containers Getting Started Guide.
  - For non-DGX users, see NVIDIA® GPU Cloud™ (NGC) container registry installation documentation based on your platform.
- Ensure that you have access and can log in to the NGC container registry.

  Refer to NGC Getting Started Guide for more information.

The deep learning frameworks, the NGC Docker containers, and the deep learning framework containers are stored in the `nvcr.io/nvidia` repository.

Chapter 3. Running The Triton Inference Server

About this task

To get set up and start using Triton Inference Server, refer to the Triton Inference Server documentation.

Chapter 4. Triton Inference Server Release 25.04

The Triton Inference Server container image, release 25.04, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 24.04 including Python 3.12
- NVIDIA CUDA 12.9.0
- NVIDIA cuBLAS 12.9.0.2
- cuDNN 9.9.0.52
- NVIDIA NCCL 2.26.3 (optimized for NVIDIA NVLink®)
- NVIDIA TensorRT™ 10.9.0.34
- UCX 1.18.0
- GDRCopy 2.4.1
- NVIDIA HPC-X 2.21
- OpenMPI 4.1.7
- FIL
- NVIDIA DALI® 1.48
- nvImageCodec 0.2.0.7
- ONNX Runtime 1.21.0
- Intel OpenVINO 2025.0.0
- DCGM 3.3.6
- TensorRT-LLM version release/0.18.0

- vLLM version 0.8.1

Driver Requirements

Release 25.04 is based on CUDA 12.9.0 which requires NVIDIA Driver release 575 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, R520, R530, R545 and R555 drivers, which are not forward-compatible with CUDA 12.6. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 25.04 supports CUDA compute capability 7.5 and later. This corresponds to GPUs in the NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, NVIDIA Ada Lovelace, and NVIDIA Blackwell architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Exposed gRPC infer thread count as a server option.
- Improved server stability during the gRPC client cancellation.
- Improved server stability in tracing mode.
- Added BLS decoupled request cancellation in the Python Backend.
- GenAI-Perf now offers a new configuration file alongside the command line.
- GenAI-Perf now supports the Huggingface TGI generated endpoint.
- GenAI-Perf added a Token per second per user (TPS/user) metric.
- GenAI-Perf metric parsing speed was increased by 60%.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
25.04	25.04	24.04	NVIDIA CUDA 12.9.0	TensorRT 10.9.0.34
25.03	2.56.0		NVIDIA CUDA 12.8.1.012	TensorRT 10.9.0.34
25.02	2.55.0		NVIDIA CUDA 12.8.0.38	TensorRT 10.8.0.43
25.01	2.54.0		NVIDIA CUDA 12.8.0	TensorRT 10.8.0.43
24.12	2.53.0		NVIDIA CUDA 12.6.3	TensorRT 10.6.0.26
24.11	2.52.0		NVIDIA CUDA 12.6.3	TensorRT 10.6.0.26
24.10	2.51.0	22.04	NVIDIA CUDA 12.6.2	TensorRT 10.5.0.18
24.09	2.50.0		NVIDIA CUDA 12.6.1	TensorRT 10.4.0.26
24.08	2.49.0		NVIDIA CUDA 12.6	TensorRT 10.3.0.26
24.07	2.48.0		NVIDIA CUDA 12.5.1	TensorRT 10.2.0.19
24.06	2.47.0		NVIDIA CUDA 12.5.0.23	TensorRT 10.1.0.27
24.05	2.46.0		NVIDIA CUDA 12.4.1	TensorRT 10.0.1.6
24.04	2.45.0			TensorRT 8.6.3
24.03	2.44		NVIDIA CUDA 12.4.0.41	
24.02	2.43		NVIDIA CUDA 12.3.2	
24.01	2.42			TensorRT 8.6.1.6
23.12	2.41			
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- vLLM backend for 25.04 might be unstable with the vLLM V1 architecture. We recommend switching to V0 for this release, by setting `VLLM_USE_V1` environment variable to 0.
- The core Python binding may incur an additional D2H and H2D copy if the backend and frontend both specify device memory to be used for response tensors.
- A segmentation fault related to DCGM and NSCQ may be encountered during server shutdown on NVSwitch systems. A possible workaround for this issue is to disable the collection of GPU metrics `tritonserver --allow-gpu-metrics false ...`
- vLLM backend currently does not take advantage of the vLLM v0.6 performance improvement when metrics are enabled.
- When using TensorRT models, if auto-complete configuration is disabled and `is_non_linear_format_io:true` for reformat-free tensors is not provided in the model configuration, the model may not load successfully.
- When using Python models indecoupled mode, users need to ensure that the ResponseSender goes out of scope or is properly cleaned up before unloading the model to guarantee that the unloading process executes correctly.
- Restart support was temporarily removed for Python models.
- The Triton Inference Server with vLLM backend currently does not support running vLLM models with tensor parallelism sizes greater than 1 and default "distributed_executor_backend" setting when using explicit model control mode.

In attempt to load a vllm model (tp > 1) in explicit mode, users could potentially see failure at the `initialize` step: `could not acquire lock for <_io.BufferedWriter name='<stdout>'> at interpreter shutdown, possibly due to daemon threads`. For the default model control mode, after server shutdown, vllm related sub-processes are not killed. Related vllm issue: https://github.com/vllm-project/vllm/issues/6766 . Please specify `distributed_executor_backend:ray` in the `model.json` when deploying vllm models with tensor parallelism > 1.

- When loading models with file override, multiple model configuration files are not supported. Users must provide the model configuration by setting parameter `config : <JSON>` instead of custom configuration file in the following format: `file:configs/<model-config-name>.pbtxt : <base64-encoded-file-content>`.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- The Java CAPI is known to have intermittent segfaults.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. NVIDIA recommends experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.

- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.
- Python backend support for Windows is limited and does not currently support the following features:
  - GPU tensors
  - CPU and GPU-related metrics
  - Custom execution environments
  - The model load/unload APIs

Chapter 5. Triton Inference Server Release 25.03

The Triton Inference Server container image, release 25.03, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 24.04 including Python 3.12
- NVIDIA CUDA 12.8.1.012
- NVIDIA cuBLAS 12.8.4.1
- cuDNN 9.8.0.87
- NVIDIA NCCL 2.25.1 (optimized for NVIDIA NVLink®)
- NVIDIA TensorRT™ 10.9.0.34
- UCX 1.18.0
- GDRCopy 2.4.1
- NVIDIA HPC-X 2.21
- OpenMPI 4.1.7
- FIL
- NVIDIA DALI® 1.47
- nvImageCodec 0.2.0.7
- ONNX Runtime 1.21.0
- Intel OpenVINO 2025.0.0
- DCGM 3.3.6
- TensorRT-LLM version release/0.18.0

- vLLM version 0.7.3

Driver Requirements

Release 25.03 is based on CUDA 12.8.1.012 which requires NVIDIA Driver release 560 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, R520, R530, R545 and R555 drivers, which are not forward-compatible with CUDA 12.6. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 25.03 supports CUDA compute capability 7.5 and later. This corresponds to GPUs in the NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, NVIDIA Ada Lovelace, and NVIDIA Blackwell architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The Tensorflow Backend has been deprecated starting in 25.03. The last release of Triton Inference Server with the Tensorflow Backend is 25.02. Users wishing to continue using the Tensorflow backend in 25.03 and later can build the Tensorflow Backend from the source.
- The “XX.YY-tf2-python-py3” container will no longer be available starting in 25.03. See the Tensorflow Backend deprecation.
- Added generate and generate_stream inference types to SageMaker server. Customers can choose which inference types - infer (default), generate or generate_stream using SAGEMAKER_TRITON_INFERENCE_TYPE environment variable during server launch.
- In an effort to allow quick, on-demand metric retrieval for external load balancers such as the Kubernetes Inference Gateway API, Triton when used with TRT-LLM can include live KV-cache utilization and capacity metrics in the HTTP response header when processing inference requests.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton

Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
25.03	2.56.0	24.04	NVIDIA CUDA 12.8.1.012	TensorRT 10.9.0.34
25.02	2.55.0		NVIDIA CUDA 12.8.0.38	TensorRT 10.8.0.43
25.01	2.54.0		NVIDIA CUDA 12.8.0	TensorRT 10.8.0.43
24.12	2.53.0		NVIDIA CUDA 12.6.3	TensorRT 10.6.0.26
24.11	2.52.0		NVIDIA CUDA 12.6.3	TensorRT 10.6.0.26
24.10	2.51.0	22.04	NVIDIA CUDA 12.6.2	TensorRT 10.5.0.18
24.09	2.50.0		NVIDIA CUDA 12.6.1	TensorRT 10.4.0.26
24.08	2.49.0		NVIDIA CUDA 12.6	TensorRT 10.3.0.26
24.07	2.48.0		NVIDIA CUDA 12.5.1	TensorRT 10.2.0.19
24.06	2.47.0		NVIDIA CUDA 12.5.0.23	TensorRT 10.1.0.27
24.05	2.46.0		NVIDIA CUDA 12.4.1	TensorRT 10.0.1.6
24.04	2.45.0			TensorRT 8.6.3
24.03	2.44		NVIDIA CUDA 12.4.0.41	
24.02	2.43		NVIDIA CUDA 12.3.2	
24.01	2.42			TensorRT 8.6.1.6
23.12	2.41			
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- The core Python binding may incur an additional D2H and H2D copy if the backend and frontend both specify device memory to be used for response tensors.
- A segmentation fault related to DCGM and NSCQ may be encountered during server shutdown on NVSwitch systems. A possible workaround for this issue is to disable the collection of GPU metrics `tritonserver --allow-gpu-metrics false ...`
- vLLM backend currently does not take advantage of the vLLM v0.6 performance improvement when metrics are enabled.
- When using TensorRT models, if auto-complete configuration is disabled and `is_non_linear_format_io:true` for reformat-free tensors is not provided in the model configuration, the model may not load successfully.
- When using Python models indecoupled mode, users need to ensure that the ResponseSender goes out of scope or is properly cleaned up before unloading the model to guarantee that the unloading process executes correctly.
- The Triton Inference Server with vLLM backend currently does not support running vLLM models with tensor parallelism sizes greater than 1 and default "distributed_executor_backend" setting when using explicit model control mode. In attempt to load a vllm model (tp > 1) in explicit mode, users could potentially see failure at the `initialize` step: `could not acquire lock for <_io.BufferedWriter name='<stdout>'> at interpreter shutdown, possibly due to daemon threads`. For the default model control mode, after server shutdown, vllm related sub-

processes are not killed. Related vllm issue: https://github.com/vllm-project/vllm/issues/6766 . Please specify `distributed_executor_backend:ray` in the `model.json` when deploying vllm models with tensor parallelism > 1.

- When loading models with file override, multiple model configuration files are not supported. Users must provide the model configuration by setting parameter `config : <JSON>` instead of custom configuration file in the following format: `file:configs/<model-config-name>.pbtxt : <base64-encoded-file-content>`.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- The Java CAPI is known to have intermittent segfaults.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. NVIDIA recommends experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.

- Python backend support for Windows is limited and does not currently support the following features:
  - GPU tensors
  - CPU and GPU-related metrics
  - Custom execution environments
  - The model load/unload APIs

Chapter 6. Triton Inference Server Release 25.02

The Triton Inference Server container image, release 25.02, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 24.04 including Python 3.12
- NVIDIA CUDA 12.8.38
- NVIDIA cuBLAS 12.8.3.14
- cuDNN 9.7.1.26
- NVIDIA NCCL 2.25.1 (optimized for NVIDIA NVLink®)
- NVIDIA TensorRT™ 10.8.0.43
- OpenUCX 1.15.0
- GDRCopy 2.4.1
- NVIDIA HPC-X 2.21
- OpenMPI 4.1.7
- FIL
- NVIDIA DALI® 1.46
- nvImageCodec 0.2.0.7
- ONNX Runtime 1.20.1
- Intel OpenVINO 2024.05.0
- DCGM 3.3.6
- TensorRT-LLM version release/0.17.0

- vLLM version 0.7.0

Driver Requirements

Release 25.02 is based on CUDA 12.8.0.38 which requires NVIDIA Driver release 560 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, R520, R530, R545 and R555 drivers, which are not forward-compatible with CUDA 12.6. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 25.02 supports CUDA compute capability 7.5 and later. This corresponds to GPUs in the NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, NVIDIA Ada Lovelace, and NVIDIA Blackwell architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Python backend now supports setting and retrieving Inference Response Parameters on InferenceResponse objects on model.py.
- Optimized the core Python binding architecture leading to improved OpenAI frontend performance.
- Added dynamic sampling parameter handling, improving flexibility and consistency across vllm interactions. Added support for “guided_generation” request parameter for efficient constrained decoding workflows.Improved Multi-Lora handling in TRTLLM GRPC Client `end_to_end_grpc_client.py`
- Improved Multi-Lora handling in TRTLLM GRPC Client `end_to_end_grpc_client.py`
- GenAI-Perf added the ability to format output using Jinja2 templates.
- GenAI-Perf telemetry now supports multiple metric endpoints.
- GenAI-Perf now supports increased corpus size, 90x the previously supported size.
- GenAI-Perf now supports keys without values as input.
- GenAI-Perf fixed the OSL issue due to Performance Analyzer not removing the first 4 bytes from output.

- GenAI-Perf added a chat template option for the TRT-LLM engine.
- Performance Analyzer fixed TRITON_ENABLE_GPU compile definition bug.
- Performance Analyzer bumped minimum required C++ version to C++20.
- Performance Analyzer modified to disallow user attempts to use concurrency and warmup with the schedule flag.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
25.02	2.55.0	24.04	NVIDIA CUDA 12.8.0.38	TensorRT 10.8.0.43
25.01	2.54.0		NVIDIA CUDA 12.8.0	TensorRT 10.8.0.43
24.12	2.53.0		NVIDIA CUDA 12.6.3	TensorRT 10.6.0.26
24.11	2.52.0		NVIDIA CUDA 12.6.3	TensorRT 10.6.0.26
24.10	2.51.0	22.04	NVIDIA CUDA 12.6.2	TensorRT 10.5.0.18
24.09	2.50.0		NVIDIA CUDA 12.6.1	TensorRT 10.4.0.26
24.08	2.49.0		NVIDIA CUDA 12.6	TensorRT 10.3.0.26
24.07	2.48.0		NVIDIA CUDA 12.5.1	TensorRT 10.2.0.19
24.06	2.47.0		NVIDIA CUDA 12.5.0.23	TensorRT 10.1.0.27
24.05	2.46.0		NVIDIA CUDA 12.4.1	TensorRT 10.0.1.6
24.04	2.45.0			TensorRT 8.6.3
24.03	2.44		NVIDIA CUDA 12.4.0.41	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
24.02	2.43		NVIDIA CUDA 12.3.2	
24.01	2.42			TensorRT 8.6.1.6
23.12	2.41			
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- The core Python binding may incur an additional D2H and H2D copy if the backend and frontend both specify device memory to be used for response tensors.
- A segmentation fault related to DCGM and NSCQ may be encountered during server shutdown on NVSwitch systems. A possible workaround for this issue is to disable the collection of GPU metrics `tritonserver --allow-gpu-metrics false ...`
- vLLM backend currently does not take advantage of the vLLM v0.6 performance improvement when metrics are enabled.
- Incorrect results are known to occur when using TensorRT (TRT) Backend for inference using int8 data type for I/O on the Blackwell GPU architecture.
- When using TensorRT models, if auto-complete configuration is disabled and `is_non_linear_format_io:true` for reformat-free tensors is not provided in the model configuration, the model may not load successfully.

- When using Python models indecoupled mode, users need to ensure that the ResponseSender goes out of scope or is properly cleaned up before unloading the model to guarantee that the unloading process executes correctly.
- Restart support was temporarily removed for Python models.
- The Triton Inference Server with vLLM backend currently does not support running vLLM models with tensor parallelism sizes greater than 1 and default "distributed_executor_backend" setting when using explicit model control mode. In attempt to load a vllm model (tp > 1) in explicit mode, users could potentially see failure at the `initialize` step: `could not acquire lock for <_io.BufferedWriter name='<stdout>'> at interpreter shutdown, possibly due to daemon threads`. For the default model control mode, after server shutdown, vllm related sub-processes are not killed. Related vllm issue: https://github.com/vllm-project/vllm/issues/6766 . Please specify `distributed_executor_backend:ray` in the `model.json` when deploying vllm models with tensor parallelism > 1.
- When loading models with file override, multiple model configuration files are not supported. Users must provide the model configuration by setting parameter `config : <JSON>` instead of custom configuration file in the following format: `file:configs/<model-config-name>.pbtxt : <base64-encoded-file-content>`.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- The Java CAPI is known to have intermittent segfaults.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. NVIDIA recommends experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The

correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.

- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.
- Python backend support for Windows is limited and does not currently support the following features:
  - GPU tensors
  - CPU and GPU-related metrics
  - Custom execution environments
  - The model load/unload APIs

Chapter 7. Triton Inference Server Release 25.01

The Triton Inference Server container image, release 25.01, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 24.04 including Python 3.12
- NVIDIA CUDA 12.8.0.038
- NVIDIA cuBLAS 12.8.3.14
- cuDNN 9.7.0.66
- NVIDIA NCCL 2.25.1 (optimized for NVIDIA NVLink®)
- NVIDIA TensorRT™ 10.8.0.43
- OpenUCX 1.15.0
- GDRCopy 2.4.1
- NVIDIA HPC-X 2.21
- OpenMPI 4.1.7
- FIL
- NVIDIA DALI® 1.45
- nvImageCodec 0.2.0.7
- ONNX Runtime 1.20.1
- Intel OpenVINO 2024.05.0
- DCGM 3.3.6
- TensorRT-LLM version release/0.17.0

- vLLM version 0.6.3.1

Driver Requirements

Release 25.01 is based on CUDA 12.8.0.038 which requires NVIDIA Driver release 560 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, R520, R530, R545 and R555 drivers, which are not forward-compatible with CUDA 12.6. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 25.01 supports CUDA compute capability 7.5 and later. This corresponds to GPUs in the NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, NVIDIA Ada Lovelace, and NVIDIA Blackwell architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Starting with the 25.01 release, Triton Inference Server supports Blackwell GPU architectures.
- Fixed a bug when passing the correlation ID of string type to python_backend. Added datatype checks to correlation ID values.
- vLLM backend can now take advantage of the vLLM v0.6 performance improvement by communicating with the vLLM engine via ZMQ.
- GenAI-Perf now provides the exact input sequence length requested for synthetic text generation.
- GenAI-Perf supports the creation of a prefix pool to emulate system prompts via --num-system-prompts and --system-prompt-length.
- GenAI-Perf improved error visibility via returning more detailed errors when OpenAI frontends return an error or metric generation fails.
- GenAI-Perf reports time-to-second-token and request count in its metrics.
- GenAI-Perf allows the use of a custom tokenizer in its “compare” subcommand for comparing multiple profiles.
- GenAI-Perf natively supports --request-count for sending a specific number of requests and --header for sending a list of headers with every request.

- Model Analyzer functionality has been migrated to GenAI-Perf via the “analyze” subcommand, enabling the tool to sweep and find the optimal model configuration.
- A bytes appending bug was fixed in GenAI-Perf, resulting in more accurate output sequence lengths for Triton.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
25.01	2.54.0	24.04	NVIDIA CUDA 12.8.0	TensorRT 10.8.0.43
24.12	2.53.0		NVIDIA CUDA 12.6.3	TensorRT 10.6.0.26
24.11	2.52.0		NVIDIA CUDA 12.6.3	TensorRT 10.6.0.26
24.10	2.51.0	22.04	NVIDIA CUDA 12.6.2	TensorRT 10.5.0.18
24.09	2.50.0		NVIDIA CUDA 12.6.1	TensorRT 10.4.0.26
24.08	2.49.0		NVIDIA CUDA 12.6	TensorRT 10.3.0.26
24.07	2.48.0		NVIDIA CUDA 12.5.1	TensorRT 10.2.0.19
24.06	2.47.0		NVIDIA CUDA 12.5.0.23	TensorRT 10.1.0.27
24.05	2.46.0		NVIDIA CUDA 12.4.1	TensorRT 10.0.1.6
24.04	2.45.0			TensorRT 8.6.3
24.03	2.44		NVIDIA CUDA 12.4.0.41	
24.02	2.43		NVIDIA CUDA 12.3.2	
24.01	2.42			TensorRT 8.6.1.6
23.12	2.41			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- A segmentation fault related to DCGM and NSCQ may be encountered during server shutdown on NVSwitch systems. A possible workaround for this issue is to disable the collection of GPU metrics `tritonserver --allow-gpu-metrics false ...`
- vLLM backend currently does not take advantage of the vLLM v0.6 performance improvement when metrics are enabled.
- Please note, that the vllm version provided in 25.01 container is 0.6.3.post1. Due to some issues with vllm library versioning, `vllm.__version__` displays `0.6.3`.
- Incorrect results are known to occur when using TensorRT (TRT) Backend for inference using int8 data type for I/O on the Blackwell GPU architecture.
- When running Torch TRT models, the output may differ from running the same model on a previous release.
- When using TensorRT models, if auto-complete configuration is disabled and `is_non_linear_format_io:true` for reformat-free tensors is not provided in the model configuration, the model may not load successfully.
- When using Python models indecoupled mode, users need to ensure that the ResponseSender goes out of scope or is properly cleaned up before unloading the model to guarantee that the unloading process executes correctly.
- Restart support was temporarily removed for Python models.

- The Triton Inference Server with vLLM backend currently does not support running vLLM models with tensor parallelism sizes greater than 1 and default "distributed_executor_backend" setting when using explicit model control mode. In attempt to load a vllm model (tp > 1) in explicit mode, users could potentially see failure at the `initialize` step: `could not acquire lock for <_io.BufferedWriter name='<stdout>'> at interpreter shutdown, possibly due to daemon threads`. For the default model control mode, after server shutdown, vllm related sub-processes are not killed. Related vllm issue: https://github.com/vllm-project/vllm/issues/6766 . Please specify `distributed_executor_backend:ray` in the `model.json` when deploying vllm models with tensor parallelism > 1.
- When loading models with file override, multiple model configuration files are not supported. Users must provide the model configuration by setting parameter `config : <JSON>` instead of custom configuration file in the following format: `file:configs/<model-config-name>.pbtxt : <base64-encoded-file-content>`.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- The Java CAPI is known to have intermittent segfaults.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. NVIDIA recommends experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.

- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.
- Python backend support for Windows is limited and does not currently support the following features:
  - GPU tensors
  - CPU and GPU-related metrics
  - Custom execution environments
  - The model load/unload APIs

Chapter 8. Triton Inference Server Release 24.12

The Triton Inference Server container image, release 24.12, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 24.04 including Python 3.12
- NVIDIA CUDA 12.6.3
- NVIDIA cuBLAS 12.6.4.1
- cuDNN 9.6.0.74
- NVIDIA NCCL 2.23.4 (optimized for NVIDIA NVLink®)
- NVIDIA TensorRT™ 10.7.0.23
- OpenUCX 1.15.0
- GDRCopy 2.4.1
- NVIDIA HPC-X 2.21
- OpenMPI 4.1.7
- FIL
- NVIDIA DALI® 1.44
- nvImageCodec 0.2.0.7
- ONNX Runtime 1.20.1
- Intel OpenVINO 2024.05.0
- DCGM 3.3.6
- TensorRT-LLM version release/0.16.0

- vLLM version 0.5.5

Driver Requirements

Release 24.12 is based on CUDA 12.6.3 which requires NVIDIA Driver release 560 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, R520, R530, R545 and R555 drivers, which are not forward-compatible with CUDA 12.6. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 24.12 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- vLLM backend health check may be optionally enabled which unloads the model if the vLLM engine health check failed.
- vLLM backend supports sending additional outputs from vLLM if requested.
- Improved server stability during the gRPC client cancellation.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
24.12	2.53.0	24.04	NVIDIA CUDA 12.6.3	TensorRT 10.7.0.23
24.11	2.52.0		NVIDIA CUDA 12.6.3	TensorRT 10.6.0.26

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
24.10	2.51.0	22.04	NVIDIA CUDA 12.6.2	TensorRT 10.5.0.18
24.09	2.50.0		NVIDIA CUDA 12.6.1	TensorRT 10.4.0.26
24.08	2.49.0		NVIDIA CUDA 12.6	TensorRT 10.3.0.26
24.07	2.48.0		NVIDIA CUDA 12.5.1	TensorRT 10.2.0.19
24.06	2.47.0		NVIDIA CUDA 12.5.0.23	TensorRT 10.1.0.27
24.05	2.46.0		NVIDIA CUDA 12.4.1	TensorRT 10.0.1.6
24.04	2.45.0			TensorRT 8.6.3
24.03	2.44		NVIDIA CUDA 12.4.0.41	
24.02	2.43		NVIDIA CUDA 12.3.2	
24.01	2.42			TensorRT 8.6.1.6
23.12	2.41			
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Numpy 2.x is not currently supported for Python Backend models and may cause them to return empty tensors unexpectedly, please use Numpy 1.x until support is added.
- To build the Llama 3.1 engine inside the 24.09-trtllm-python-py3 image, make sure to upgrade the transformer library to 4.43+ due to the bug in 4.43.x. One option to do so is to run `pip install -U transformers`. For more information, please refer to the discussion: https://github.com/NVIDIA/TensorRT-LLM/issues/2121
- Triton vLLM container comes with the vLLM version, which has a known vulnerability: https://github.com/advisories/GHSA-w2r7-9579-27hf. Note, that the affected code is not invoked at runtime, therefore the Triton vLLM container is not affected by this issue.
- When running Torch TRT models, the output may differ from running the same model on a previous release.
- When using TensorRT models, if auto-complete configuration is disabled and `is_non_linear_format_io:true` for reformat-free tensors is not provided in the model configuration, the model may not load successfully.
- When using Python models indecoupled mode, users need to ensure that the ResponseSender goes out of scope or is properly cleaned up before unloading the model to guarantee that the unloading process executes correctly.
- Restart support was temporarily removed for Python models.
- Triton TensorRT-LLM Backend container image uses TensorRT-LLM version 0.16.0 and is built out of nvcr.io/nvidia/tritonserver:24.11-py3-min. Please refer to the Triton TRT-LLM Container Support Matrix section in the GitHub release note for more details.
- The Triton Inference Server with vLLM backend currently does not support running vLLM models with tensor parallelism sizes greater than 1 and default "distributed_executor_backend" setting when using explicit model control mode. In attempt to load a vllm model (tp > 1) in explicit mode, users could potentially see failure at the `initialize` step: `could not acquire lock for <_io.BufferedWriter name='<stdout>'> at interpreter shutdown, possibly due to daemon threads`. For the default model control mode, after server shutdown, vllm related sub-processes are not killed. Related vllm issue: https://github.com/vllm-project/vllm/issues/6766 . Please specify `distributed_executor_backend:ray` in the `model.json` when deploying vllm models with tensor parallelism > 1.
- When loading models with file override, multiple model configuration files are not supported. Users must provide the model configuration by setting parameter `config : <JSON>` instead of custom configuration file in the following format:

`file:configs/<model-config-name>.pbtxt : <base64-encoded-file-content>`.

- Perf Analyzer no longer supports the `--trace-file` option.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- The Java CAPI is known to have intermittent segfaults.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. NVIDIA recommends experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.
- Python backend support for Windows is limited and does not currently support the following features:
  - GPU tensors
  - CPU and GPU-related metrics

- Custom execution environments
- The model load/unload APIs

Chapter 9. Triton Inference Server Release 24.11

The Triton Inference Server container image, release 24.11, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 24.04 including Python 3.12
- NVIDIA CUDA 12.6.3
- NVIDIA cuBLAS 12.6.4.1
- cuDNN 9.5.1.17
- NVIDIA NCCL 2.23.4 (optimized for NVIDIA NVLink®)
- NVIDIA TensorRT™ 10.6.0.26
- OpenUCX 1.15.0
- GDRCopy 2.4.1
- NVIDIA HPC-X 2.21
- OpenMPI 4.1.7
- FIL
- NVIDIA DALI® 1.43
- nvImageCodec 0.2.0.7
- ONNX Runtime 1.19.2
- Intel OpenVINO 2024.4.0
- DCGM 3.2.6
- TensorRT-LLM version release/0.15.0

- vLLM version 0.5.5 post 1

Driver Requirements

Release 24.11 is based on CUDA 12.6.3 which requires NVIDIA Driver release 560 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, R520, R530, R545 and R555 drivers, which are not forward-compatible with CUDA 12.6. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 24.11 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Conceptual Guides were enhanced with a comprehensive tutorial on Semantic Caching optimization for LLM workloads.
- Triton Metrics
  - Added a new histogram metric “Request to First Response Time” to decoupled models. Enabled by setting `--metrics-config histogram_latencies=true`.Added a new histogram metric `Request to First Response Time` to decoupled models. Enabled by setting `--metrics-config summary_latencies=true`. See the docs here:https://github.com/triton-inference-server/server/blob/r24.11/docs/user_guide/metrics.md#histograms
  - A new model configuration field `model_metrics` that allows overriding default buckets for histogram metric families.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
24.11	2.52.0	24.04	NVIDIA CUDA 12.6.3	TensorRT 10.6.0.26
24.10	2.51.0		NVIDIA CUDA 12.6.2	TensorRT 10.5.0.18
24.09	2.50.0		NVIDIA CUDA 12.6.1	TensorRT 10.4.0.26
24.08	2.49.0		NVIDIA CUDA 12.6	TensorRT 10.3.0.26
24.07	2.48.0		NVIDIA CUDA 12.5.1	TensorRT 10.2.0.19
24.06	2.47.0		NVIDIA CUDA 12.5.0.23	TensorRT 10.1.0.27
24.05	2.46.0		NVIDIA CUDA 12.4.1	TensorRT 10.0.1.6
24.04	2.45.0			TensorRT 8.6.3
24.03	2.44		NVIDIA CUDA 12.4.0.41	
24.02	2.43		NVIDIA CUDA 12.3.2	
24.01	2.42			TensorRT 8.6.1.6
23.12	2.41			
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- TensorFlow backend may leak memory due to a known issue with the cuDNN library shipped with the container.
- Triton vLLM container comes with the vLLM version, which has a known vulnerability: https://github.com/advisories/GHSA-w2r7-9579-27hf. Note, that the affected code is not invoked at runtime, therefore the Triton vLLM container is not affected by this issue.
- When running Torch TRT models, the output may differ from running the same model on a previous release.
- When using TensorRT models, if auto-complete configuration is disabled and `is_non_linear_format_io:true` for reformat-free tensors is not provided in the model configuration, the model may not load successfully.
- When using Python models indecoupled mode, users need to ensure that the ResponseSender goes out of scope or is properly cleaned up before unloading the model to guarantee that the unloading process executes correctly.
- Restart support was temporarily removed for Python models.
- Triton TensorRT-LLM Backend container image uses TensorRT-LLM version 0.15.0 and is built out of nvcr.io/nvidia/tritonserver:24.10-py3-min. Please refer to the Triton TRT-LLM Container Support Matrix section in the GitHub release note for more details.
- The Triton Inference Server with vLLM backend currently does not support running vLLM models with tensor parallelism sizes greater than 1 and default "distributed_executor_backend" setting when using explicit model control mode. In attempt to load a vllm model (tp > 1) in explicit mode, users could potentially see failure at the `initialize` step: `could not acquire lock for <_io.BufferedWriter name='<stdout>'> at interpreter shutdown, possibly due to daemon threads`. For the default model control mode, after server shutdown, vllm related sub-processes are not killed. Related vllm issue: https://github.com/vllm-project/vllm/issues/6766 . Please specify `distributed_executor_backend:ray` in the `model.json` when deploying vllm models with tensor parallelism > 1.
- When loading models with file override, multiple model configuration files are not supported. Users must provide the model configuration by setting parameter `config : <JSON>` instead of custom configuration file in the following format:

`file:configs/<model-config-name>.pbtxt : <base64-encoded-file-content>`.

- Perf Analyzer no longer supports the `--trace-file` option.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- The Java CAPI is known to have intermittent segfaults.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. NVIDIA recommends experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.
- Python backend support for Windows is limited and does not currently support the following features:
  - GPU tensors
  - CPU and GPU-related metrics

  - Custom execution environments
  - The model load/unload APIs
- The latest GenAI-Perf package on pypi.org is version 0.0.9dev while the latest Triton SDK container (24.11) contains GenAI-Perf version 0.0.8.

Chapter 10. Triton Inference Server Release 24.10

The Triton Inference Server container image, release 24.10, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 22.04 including Python 3.10
- NVIDIA CUDA 12.6.2
- NVIDIA cuBLAS 12.6.3.3
- cuDNN 9.5.0.50
- NVIDIA NCCL 2.22.3 (optimized for NVIDIA NVLink®)
- NVIDIA TensorRT™ 10.5.0.18
- OpenUCX 1.15.0
- GDRCopy 2.3
- NVIDIA HPC-X 2.20
- OpenMPI 4.1.7
- FIL
- NVIDIA DALI® 1.42
- nvImageCodec 0.2.0.7
- ONNX Runtime 1.19.2
- Intel OpenVINO 2024.0.0
- DCGM 3.2.6
- TensorRT-LLM version release/0.14.0

- vLLM version 0.5.3 post 1

Driver Requirements

Release 24.10 is based on CUDA 12.6.2 which requires NVIDIA Driver release 560 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, R520, R530, R545 and R555 drivers, which are not forward-compatible with CUDA 12.6. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 24.10 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Optimized vLLM performance with custom metrics.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
24.10	2.51.0	22.04	NVIDIA CUDA 12.6.2	TensorRT 10.5.0.18
24.09	2.50.0		NVIDIA CUDA 12.6.1	TensorRT 10.4.0.26
24.08	2.49.0		NVIDIA CUDA 12.6	TensorRT 10.3.0.26

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
24.07	2.48.0		NVIDIA CUDA 12.5.1	TensorRT 10.2.0.19
24.06	2.47.0		NVIDIA CUDA 12.5.0.23	TensorRT 10.1.0.27
24.05	2.46.0		NVIDIA CUDA 12.4.1	TensorRT 10.0.1.6
24.04	2.45.0			TensorRT 8.6.3
24.03	2.44		NVIDIA CUDA 12.4.0.41	
24.02	2.43		NVIDIA CUDA 12.3.2	
24.01	2.42			TensorRT 8.6.1.6
23.12	2.41			
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Numpy 2.x is not currently supported for Python Backend models and may cause them to return empty tensors unxpectedly, please use Numpy 1.x until support is added.
- Triton vLLM container comes with the vLLM version, which has a known vulnerability: https://github.com/advisories/GHSA-w2r7-9579-27hf. Note, that the affected code is not invoked at runtime, therefore the Triton vLLM container is not affected by this issue.
- When running Torch TRT models, the output may differ from running the same model on a previous release.
- When using TensorRT models, if auto-complete configuration is disabled and `is_non_linear_format_io:true` for reformat-free tensors is not provided in the model configuration, the model may not load successfully.
- When using Python models indecoupled mode, users need to ensure that the ResponseSender goes out of scope or is properly cleaned up before unloading the model to guarantee that the unloading process executes correctly.
- Restart support was temporarily removed for Python models.
- Triton TensorRT-LLM Backend container image uses TensorRT-LLM version 0.14.0 and built out of nvcr.io/nvidia/tritonserver:24.07-py3-min. Please refer to the Triton TRT-LLM Container Support Matrix section in the GitHub release note for more details.
- The Triton Inference Server with vLLM backend currently does not support running vLLM models with tensor parallelism sizes greater than 1 and default "distributed_executor_backend" setting when using explicit model control mode. In attempt to load a vllm model (tp > 1) in explicit mode, users could potentially see failure at the `initialize` step: `could not acquire lock for <_io.BufferedWriter name='<stdout>'> at interpreter shutdown, possibly due to daemon threads`. For the default model control mode, after server shutdown, vllm related sub-processes are not killed. Related vllm issue: https://github.com/vllm-project/vllm/issues/6766 . Please specify `distributed_executor_backend:ray` in the `model.json` when deploying vllm models with tensor parallelism > 1.
- When loading models with file override, multiple model configuration files are not supported. Users must provide the model configuration by setting parameter `config : <JSON>` instead of custom configuration file in the following format: `file:configs/<model-config-name>.pbtxt : <base64-encoded-file-content>`.
- Perf Analyzer no longer supports the `--trace-file` option.
- TensorRT-LLM backend provides limited support of Triton extensions and features.

- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- The Java CAPI is known to have intermittent segfaults.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. NVIDIA recommends experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.
- Python backend support for Windows is limited and does not currently support the following features:
  - GPU tensors
  - CPU and GPU-related metrics
  - Custom execution environments
  - The model load/unload APIs

Chapter 11. Triton Inference Server Release 24.09

The Triton Inference Server container image, release 24.09, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 22.04 including Python 3.10
- NVIDIA CUDA 12.6.1
- NVIDIA cuBLAS 12.6.3.1
- cuDNN 9.4.0.58
- NVIDIA NCCL 2.22.3 (optimized for NVIDIA NVLink®)
- NVIDIA TensorRT™ 10.4.0.26
- OpenUCX 1.15.0
- GDRCopy 2.3
- NVIDIA HPC-X 2.20
- OpenMPI 4.1.7
- FIL
- NVIDIA DALI® 1.41
- nvImageCodec 0.2.0.7
- ONNX Runtime 1.19.2
- Intel OpenVINO 2024.0.0
- DCGM 3.2.6
- TensorRT-LLM version release/0.13.0

- vLLM version 0.5.3 post 1

Driver Requirements

Release 24.09 is based on CUDA 12.6 which requires NVIDIA Driver release 560 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, R520, R530, R545 and R555 drivers, which are not forward-compatible with CUDA 12.6. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 24.09 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Our tutorials were updated with 2 extensive guides on constrained decoding implementation in TensorRT-LLM python backend and function/tool calling. Guides can be found here.
- Our tutorials were also updated for Kubernetes Multi-Node and Multi-Instance Scaling with Triton and TRT-LLM; they can be found here.
- vLLM backend now supports these additional metrics. For more information, see vllm_backend.
  - vllm:e2e_request_latency_seconds
  - vllm:request_prompt_tokens
  - vllm:request_generation_tokens
  - vllm:request_params_best_of
  - vllm:request_params_n

  To enable the vLLM model's metrics reporting, add these lines to config.pbtxt:

  ```bash
  parameters: {
    key: "REPORT_CUSTOM_METRICS"
    value: {
      string_value:"yes"
    }
  ```

```
}
```

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
24.09	2.50.0	22.04	NVIDIA CUDA 12.6.1	TensorRT 10.4.0.26
24.08	2.49.0		NVIDIA CUDA 12.6	TensorRT 10.3.0.26
24.07	2.48.0		NVIDIA CUDA 12.5.1	TensorRT 10.2.0.19
24.06	2.47.0		NVIDIA CUDA 12.5.0.23	TensorRT 10.1.0.27
24.05	2.46.0		NVIDIA CUDA 12.4.1	TensorRT 10.0.1.6
24.04	2.45.0			TensorRT 8.6.3
24.03	2.44		NVIDIA CUDA 12.4.0.41	
24.02	2.43		NVIDIA CUDA 12.3.2	
24.01	2.42			TensorRT 8.6.1.6
23.12	2.41			
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	TensorRT 8.0.2.2 for ARM SBSA Linux
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- To build the Llama 3.1 engine inside the 24.09-trtllm-python-py3 image, make sure to upgrade the transformer library to 4.43+ due to the bug in 4.43.x. One option to do so is to run `pip install -U transformers`. For more information, please refer to the discussion: https://github.com/NVIDIA/TensorRT-LLM/issues/2121
- Triton vLLM container comes with the vLLM version, which has a known vulnerability: https://github.com/advisories/GHSA-w2r7-9579-27hf. Note, that the affected code is not invoked at runtime, therefore the Triton vLLM container is not affected by this issue.
- When running Torch TRT models, the output may differ from running the same model on a previous release.
- When using TensorRT models, if auto-complete configuration is disabled and `is_non_linear_format_io:true` for reformat-free tensors is not provided in the model configuration, the model may not load successfully.
- When using Python models indecoupled mode, users need to ensure that the ResponseSender goes out of scope or is properly cleaned up before unloading the model to guarantee that the unloading process executes correctly.
- Restart support was temporarily removed for Python models.
- Triton TensorRT-LLM Backend container image uses TensorRT-LLM version 0.13.0 and built out of nvcr.io/nvidia/tritonserver:24.07-py3-min. Please refer to the Triton TRT-LLM Container Support Matrix section in the GitHub release note for more details.
- The Triton Inference Server with vLLM backend currently does not support running vLLM models with tensor parallelism sizes greater than 1 and default "distributed_executor_backend" setting when using explicit model control mode. In attempt to load a vllm model (tp > 1) in explicit mode, users could potentially see failure at the `initialize` step: `could not acquire lock for <_io.BufferedWriter name='<stdout>'> at interpreter shutdown, possibly due to daemon threads`. For the default model control mode, after server shutdown, vllm related sub-

processes are not killed. Related vllm issue: https://github.com/vllm-project/vllm/issues/6766 . Please specify `distributed_executor_backend:ray` in the `model.json` when deploying vllm models with tensor parallelism > 1.

- When loading models with file override, multiple model configuration files are not supported. Users must provide the model configuration by setting parameter `config : <JSON>` instead of custom configuration file in the following format: `file:configs/<model-config-name>.pbtxt : <base64-encoded-file-content>`.
- Perf Analyzer no longer supports the `--trace-file` option.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- The Java CAPI is known to have intermittent segfaults.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. NVIDIA recommends experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud

model's folder in the temporary directory, which is deleted upon server's shutdown.

- Python backend support for Windows is limited and does not currently support the following features:
  - GPU tensors
  - CPU and GPU-related metrics
  - Custom execution environments
  - The model load/unload APIs

Chapter 12. Triton Inference Server Release 24.08

The Triton Inference Server container image, release 24.08, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 22.04 including Python 3.10
- NVIDIA CUDA 12.6
- NVIDIA cuBLAS 12.6.0.22
- cuDNN 9.3.0.75
- NVIDIA NCCL 2.22.3 (optimized for NVIDIA NVLink®)
- NVIDIA TensorRT™ 10.3.0.26
- OpenUCX 1.15.0
- GDRCopy 2.3
- NVIDIA HPC-X 2.19
- OpenMPI 4.1.7
- FIL
- NVIDIA DALI® 1.40
- nvImageCodec 0.2.0.7
- ONNX Runtime 1.18.1
- Intel OpenVINO 2024.0.0
- DCGM 3.2.6
- TensorRT-LLM version release/0.12.0

- vLLM version 0.5.3 post 1

Driver Requirements

Release 24.08 is based on CUDA 12.6 which requires NVIDIA Driver release 560 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, R520, R530, R545 and R555 drivers, which are not forward-compatible with CUDA 12.6. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 24.08 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- OpenAI-compatible embeddings and Hugging Face TEI re-ranker API-compatible rankings can now be profiled via GenAI-Perf.
- GenAI-Perf can now receive multiple user-specified prompts via --input-file.
- The request-rate for async requests have been updated in the OpenAI and HTTP clients to send requests at exactly that rate. Users submitting more requests than their models can handle can see increased latency.
  - The stabilization metric for Perf Analyzer has been updated due to these changes, so if latency does not stabilize for async models, a warning will be printed but Perf Analyzer will still complete.
- Perf Analyzer will not validate any user-supplied inputs and outputs, returning an error if the model does not contain them.
- Python backend now supports BF16 tensors via DLPack
- vLLM backend now supports reporting metrics:
  - vllm:prompt_tokens_total
  - vllm:generation_tokens_total
  - vllm:time_to_first_token_seconds
  - Vllm:time_to_first_token_seconds

To enable the vLLM model's metrics reporting, add these lines to config.pbtxt:

```bash
parameters: {
  key: "REPORT_CUSTOM_METRICS"
  value: {
    string_value:"yes"
  }
}
```

- TensorRT-LLM backend now supports specifying GPU device IDs per instance using the `gpu_device_ids` field.
- After the model config is updated to load new model versions, any loaded model versions whose model files are unmodified will not be reloaded.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
24.08	2.49.0	22.04	NVIDIA CUDA 12.6	TensorRT 10.3.0.26
24.07	2.48.0		NVIDIA CUDA 12.5.1	TensorRT 10.2.0.19
24.06	2.47.0		NVIDIA CUDA 12.5.0.23	TensorRT 10.1.0.27
24.05	2.46.0		NVIDIA CUDA 12.4.1	TensorRT 10.0.1.6
24.04	2.45.0			TensorRT 8.6.3
24.03	2.44		NVIDIA CUDA 12.4.0.41	
24.02	2.43		NVIDIA CUDA 12.3.2	
24.01	2.42			TensorRT 8.6.1.6
23.12	2.41			
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- When running Torch TRT models, the output may differ from running the same model on a previous release.
- When using TensorRT models, if auto-complete configuration is disabled and `is_non_linear_format_io:true` for reformat-free tensors is not provided in the model configuration, the model may not load successfully.
- When using Python models indecoupled mode, users need to ensure that the ResponseSender goes out of scope or is properly cleaned up before unloading the model to guarantee that the unloading process executes correctly.
- Restart support was temporarily removed for Python models.
- Triton TensorRT-LLM Backend container image uses TensorRT-LLM version 0.12.0 and built out of nvcr.io/nvidia/tritonserver:24.07-py3-min. Please refer to the Triton TRT-LLM Container Support Matrix section in the GitHub release note for more details.
- The Triton Inference Server with vLLM backend currently does not support running vLLM models with tensor parallelism sizes greater than 1 and default "distributed_executor_backend" setting when using explicit model control mode. In attempt to load a vllm model (tp > 1) in explicit mode, users could potentially see failure at the `initialize` step: `could not acquire lock for <_io.BufferedWriter name='<stdout>'> at interpreter shutdown, possibly due to daemon threads`. For the default model control mode, after server shutdown, vllm related sub-processes are not killed. Related vllm issue: https://github.com/vllm-project/

vllm/issues/6766 . Please specify `distributed_executor_backend:ray` in the `model.json` when deploying vllm models with tensor parallelism > 1.

- When loading models with file override, multiple model configuration files are not supported. Users must provide the model configuration by setting parameter `config : <JSON>` instead of custom configuration file in the following format: `file:configs/<model-config-name>.pbtxt : <base64-encoded-file-content>`.
- Perf Analyzer no longer supports the `--trace-file` option.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- The Java CAPI is known to have intermittent segfaults.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. NVIDIA recommends experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.

- Python backend support for Windows is limited and does not currently support the following features:
  - GPU tensors
  - CPU and GPU-related metrics
  - Custom execution environments
  - The model load/unload APIs
- Starting in 24.06, if you use Triton's iGPU container you might encounter this error message when loading TensorRT models built with the 24.06 TensorRT iGPU container:

```
"Serialization (Serialization assertion stdVersionRead ==
 serializationVersion failed.Version tag does not match. Note: Current
 Version: 236, Serialized Engine Version: 237)."
```

  If this happens you can rebuild your iGPU models with the 24.04 TensorRT iGPU container and then run them in the Triton 24.06 iGPU container.

Chapter 13. Triton Inference Server Release 24.07

The Triton Inference Server container image, release 24.07, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 22.04 including Python 3.10
- NVIDIA CUDA 12.5.1
- NVIDIA cuBLAS 12.5.3.2
- cuDNN 9.2.1.18
- NVIDIA NCCL 2.22.3 (optimized for NVIDIA NVLink®)
- NVIDIA TensorRT™ 10.2.0.19
- OpenUCX 1.15.0
- GDRCopy 2.3
- NVIDIA HPC-X 2.19
- OpenMPI 4.1.7
- FIL
- NVIDIA DALI® 1.39
- nvImageCodec 0.2.0.7
- ONNX Runtime 1.18.1
- Intel OpenVINO 2024.0.0
- DCGM 3.2.6
- TensorRT-LLM version release/0.11

- vLLM version 0.5.0 post 1

Driver Requirements

Release 24.07 is based on CUDA 12.5.1 which requires NVIDIA Driver release 555 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, R520 and R545 drivers, which are not forward-compatible with CUDA 12.5. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 24.07 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- OpenAI-compatible embeddings and Hugging Face TEI re-ranker API-compatible rankings can now be profiled via GenAI-Perf.
- GenAI-Perf can now receive multiple user-specified prompts via --input-file.
- The request-rate for async requests have been updated in the OpenAI and HTTP clients to send requests at exactly that rate. Users submitting more requests than their models can handle can see increased latency.
  - The stabilization metric for Perf Analyzer has been updated due to these changes, so if latency does not stabilize for async models, a warning will be printed but Perf Analyzer will still complete.
- Perf Analyzer will not validate any user-supplied inputs and outputs, returning an error if the model does not contain them.
- Triton now supports tracing custom activities in the backend. For more information please refer to documentation.
- Enhanced Failure Count Metrics to reflect failure reason of inference request.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
27.07	2.48.0	22.04	NVIDIA CUDA 12.5.1	TensorRT 10.2.0.19
24.06	2.47.0		NVIDIA CUDA 12.5.0.23	TensorRT 10.1.0.27
24.05	2.46.0		NVIDIA CUDA 12.4.1	TensorRT 10.0.1.6
24.04	2.45.0			TensorRT 8.6.3
24.03	2.44		NVIDIA CUDA 12.4.0.41	
24.02	2.43		NVIDIA CUDA 12.3.2	
24.01	2.42			TensorRT 8.6.1.6
23.12	2.41			
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- When using Python models indecoupled mode, users need to ensure that the ResponseSender goes out of scope or is properly cleaned up before unloading the model to guarantee that the unloading process executes correctly.
- Restart support was temporarily removed for Python models.
- Triton TensorRT-LLM Backend container image uses TensorRT-LLM version 0.11.0 and built out of nvcr.io/nvidia/pytorch:24.05-py3.
- The Triton Inference Server with vLLM backend (vllm version v0.5.0.post1),when using explicit model control mode, does not support running vLLM models with the default `distributed_executor_backend` and tensor parallelism sizes greater than 1. Attempting to load a vLLM model in explicit mode with tensor parallelism >1 will result in failure at the initialize step: `could not acquire lock for <_io.BufferedWriter name=<stdout>>` at interpreter shutdown, possibly due to daemon threads. Please specify `distributed_executor_backend:ray` in the `model.json` when deploying vllm models with tensor parallelism > 1.

  The Triton Inference Server with vLLM backend currently does not support running vLLM models with tensor parallelism sizes greater than 1 and default "distributed_executor_backend" setting when using explicit model control mode. In attempt to load a vllm model (tp > 1) in explicit mode, users expect to see failure at the `initialize` step: `could not acquire lock for <_io.BufferedWriter name='<stdout>'> at interpreter shutdown, possibly due to daemon threads`. Please specify `distributed_executor_backend:ray` in the `model.json` when deploying vllm models with tensor parallelism > 1.
- When loading models with file override, multiple model configuration files are not supported. Users must provide the model configuration by setting parameter `config : <JSON>` instead of custom configuration file in the following format: `file:configs/<model-config-name>.pbtxt : <base64-encoded-file-content>`.
- Perf Analyzer no longer supports the `--trace-file` option.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- The Java CAPI is known to have intermittent segfaults.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. NVIDIA recommends experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.

- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.
- Python backend support for Windows is limited and does not currently support the following features:
  - GPU tensors
  - CPU and GPU-related metrics
  - Custom execution environments
  - The model load/unload APIs
- Starting in 24.06, if you use Triton's iGPU container you might encounter this error message when loading TensorRT models built with the 24.06 TensorRT iGPU container:

```
"Serialization (Serialization assertion stdVersionRead ==
 serializationVersion failed.Version tag does not match. Note: Current
 Version: 236, Serialized Engine Version: 237)."
```

  If this happens you can rebuild your iGPU models with the 24.04 TensorRT iGPU container and then run them in the Triton 24.06 iGPU container.

Chapter 14. Triton Inference Server Release 24.06

The Triton Inference Server container image, release 24.06, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 22.04 including Python 3.10
- NVIDIA CUDA 12.5.0.23
- NVIDIA cuBLAS 12.5.2.13
- cuDNN 9.1.0.70
- NVIDIA NCCL 2.21.5 (optimized for NVIDIA NVLink®)
- NVIDIA TensorRT™ 10.1.0.27
- OpenUCX 1.15.0
- GDRCopy 2.3
- NVIDIA HPC-X 2.19
- OpenMPI 4.1.4+
- FIL
- NVIDIA DALI® 1.38
- ONNX Runtime 1.18.0
- Intel OpenVINO 2024.0
- DCGM 3.2.6
- TensorRT-LLM version release/0.10
- vLLM version 0.4.3

Driver Requirements

Release 24.06 is based on NVIDIA CUDA 12.5.0.23, which requires NVIDIA Driver release 545 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, and R520 drivers, which are not forward-compatible with CUDA 12.3. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 24.06 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The TensorRT Backend now supports the BF16 datatype.
- A new tutorial on auto-scaling and load balancing TensorRT-LLM model deployments with Triton Inference Server has been released and is located here: https://github.com/triton-inference-server/tutorials/tree/main/Deployment/Kubernetes/TensorRT-LLM_Autoscaling_and_Load_Balancing.
- A compare subcommand has been added to GenAi-Perf to allow comparison across multiple runs.
- Multi-LoRA and multi-model support in GenAI-Perf.
- Custom visualizations in GenAI-Perf.
- A fixed request count can now be requested from Perf Analyzer.
- Ensemble top-level response caching support in Perf Analyzer.
- Added –enable-peer-access to control trying to enable GPU peer access on triton startup. Default is TRUE.
- Python models in default mode may send its response using the InferenceResponseSender similarly to models in decoupled mode.
- Addressed an issue where Triton would cease processing gRPC requests after receiving multiple cancellation requests.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
24.06	2.47.0	22.04	NVIDIA CUDA 12.5.0.23	TensorRT 10.1.0.27
24.05	2.46.0		NVIDIA CUDA 12.4.1	TensorRT 10.0.1.6
24.04	2.45.0			TensorRT 8.6.3
24.03	2.44		NVIDIA CUDA 12.4.0.41	
24.02	2.43		NVIDIA CUDA 12.3.2	
24.01	2.42			TensorRT 8.6.1.6
23.12	2.41			
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- When using Python models indecoupled mode, users need to ensure that the ResponseSender goes out of scope or is properly cleaned up before unloading the model to guarantee that the unloading process executes correctly.
- Restart support was temporarily removed for Python models.
- TensorRT v10 does not support implicit batching. As a result, Triton no longer supports TensorRT models with implicit batch dimensions.
- Since TensorRT v10 no longer supports implicit batch, Tritonserver will not be able to load existing TF-TRT models that use implicit batch. Therefore, we need to build TF-TRT models withdynamic batch support.
- Multiple model configuration files are not supported by loading models with file override. Users still need to provide the model configuration by setting parameter "config" : "<JSON>" instead of custom configuration file "file:configs/<model-config-name>.pbtxt" : "<base64-encoded-file-content>".
- Perf Analyzer no longer supports the --trace-file option.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- The Java CAPI is known to have intermittent segfaults.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. NVIDIA recommends experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.

- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.
- Python backend support for Windows is limited and does not currently support the following features:
  - GPU tensors
  - CPU and GPU-related metrics
  - Custom execution environments
  - The model load/unload APIs
- Starting in 24.06, if you use Triton's iGPU container you might encounter this error message when loading TensorRT models built with the 24.06 TensorRT iGPU container:

```
"Serialization (Serialization assertion stdVersionRead ==
 serializationVersion failed.Version tag does not match. Note: Current
 Version: 236, Serialized Engine Version: 237)."
```

  If this happens you can rebuild your iGPU models with the 24.04 TensorRT iGPU container and then run them in the Triton 24.06 iGPU container.

Chapter 15. Triton Inference Server Release 24.05

The Triton Inference Server container image, release 24.05, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 22.04 including Python 3.10
- NVIDIA CUDA 12.4.1
- NVIDIA cuBLAS 12.4.5.8
- cuDNN 9.1.0.70
- NVIDIA NCCL 2.21.5 (optimized for NVIDIA NVLink®)
- NVIDIA TensorRT™ 10.0.1.6
- OpenUCX 1.15.0
- GDRCopy 2.3
- NVIDIA HPC-X 2.19
- OpenMPI 4.1.4+
- FIL
- NVIDIA DALI® 1.37.1
- ONNX Runtime 1.18.0
- Intel OpenVINO 2023.3.0
- DCGM 3.2.6
- TensorRT-LLM version release/0.10
- vLLM version 0.4.0 post1

Driver Requirements

Release 24.05 is based on NVIDIA CUDA 12.4, which requires NVIDIA Driver release 545 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, and R520 drivers, which are not forward-compatible with CUDA 12.3. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 24.05 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Added "namespace" label in metrics if the server is launched with "--model-namespacing=true". The label can now be used to distinguish metrics from two model with same name belonging to different namespace.
- Response cachinghas been extended to top-level requests to ensemble models.
- Improved the performance of Python HTTPClient library.
- Model repository can now includemultiple model configuration filesfor a given model. The specific model configuration to use can be selected when launching the server with "--model-config-name" option.
- `INTER_OP_THREAD_COUNT` and `INTRA_OP_THREAD_COUNT` parametercan now be set in `config.pbtxt` for PyTorch Backend to control thread counts in PyTorch model execution.
- FIL backend is now included in Triton's ARM-SBSA container image.
- Triton's vLLM Backend now support deployment of models with multiple LoRA adapters. See this tutorial to learn more.
- GenAi-Perf added a new compare subcommand to enable generating visual comparisons of different profile runs.
- GenAI-Perf can now accept an input file containing a single prompt string to populate input generation.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
24.05	2.46.0	22.04	NVIDIA CUDA 12.4.1	TensorRT 10.0.1.6
24.04	2.45.0			TensorRT 8.6.3
24.03	2.44		NVIDIA CUDA 12.4.0.41	
24.02	2.43		NVIDIA CUDA 12.3.2	
24.01	2.42			TensorRT 8.6.1.6
23.12	2.41			
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- When using Python models indecoupled mode, users need to ensure that the ResponseSender goes out of scope or is properly cleaned up before unloading the model to guarantee that the unloading process executes correctly.
- TensorRT v10 does not support implicit batching. As a result, Triton no longer supports TensorRT models with implicit batch dimensions.
- Since TensorRT v10 no longer supports implicit batch, Tritonserver will not be able to load existing TF-TRT models that use implicit batch. Therefore, we need to build TF-TRT models withdynamic batch support.
- Multiple model configuration files are not supported by loading models with file override. Users still need to provide the model configuration by setting parameter "config" : "<JSON>" instead of custom configuration file "file:configs/<model-config-name>.pbtxt" : "<base64-encoded-file-content>".
- Perf Analyzer no longer supports the --trace-file option.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- The Java CAPI is known to have intermittent segfaults.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. NVIDIA recommends experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.

- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.
- Python backend support for Windows is limited and does not currently support the following features:
  - GPU tensors
  - CPU and GPU-related metrics
  - Custom execution environments
  - The model load/unload APIs

Chapter 16. Triton Inference Server Release 24.04

The Triton Inference Server container image, release 24.04, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 22.04 including Python 3.10
- NVIDIA CUDA 12.4
- NVIDIA cuBLAS 12.4.5.8
- cuDNN 9.1.0.70
- NVIDIA NCCL 2.21.5 (optimized for NVIDIA NVLink®)
- NVIDIA TensorRT™ 8.6.3
- OpenUCX 1.15.0
- GDRCopy 2.3
- NVIDIA HPC-X 2.18
- OpenMPI 4.1.4+
- FIL
- NVIDIA DALI® 1.36
- ONNX Runtime 1.17.3
- Intel OpenVINO 2023.3.0
- DCGM 3.2.6
- TensorRT-LLM version release/0.9
- vLLM version 0.4.0 post1

- nvImageCodec 0.2.0.7

Driver Requirements

Release 24.04 is based on NVIDIA CUDA 12.4, which requires NVIDIA Driver release 545 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, and R520 drivers, which are not forward-compatible with CUDA 12.3. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 24.04 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Beta support for asyncio in decoupled mode in Python backend.
- Enhancements to server shutdown to take into account both HTTP live connections and inflight inferences.
- Python backend shared memory region naming has been updated to use UUIDs. This allows multiple servers to run on the machine without requiring different shared memory region prefixes.
- Support retrieving OpenTelemetry trace settings from the gRPC/HTTP endpoints.
- Log file and trace file locations can no longer be updated using the gRPC/HTTP endpoints.
- Added an iterative scheduling tutorial to demonstrate how to use iterative scheduling with a GPT2 model.
- Trace settings API now returns trace_mode and trace_config information.
- The TensorRT-LLM container now includes the tensorrt_llm Python package for creating engines.
- Added Python Client API docs to the documentation website.
- Added metric visualizations to GenAI-Perf.

- Added support to Model Analyzer for profiling LLMs with GenAI-Perf.
- Added the ability to select an output token distribution in GenAI-Perf.
- Some arguments have been renamed in the latest version of GenAI-Perf.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
24.04	2.45.0	22.04	NVIDIA CUDA 12.4	TensorRT 8.6.3
24.03	2.44		NVIDIA CUDA 12.4.0.41	
24.02	2.43		NVIDIA CUDA 12.3.2	
24.01	2.42			TensorRT 8.6.1.6
23.12	2.41			
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- The TensorRT-LLM container uses nvcr.io/nvidia/pytorch:24.02-py3 as the base image.
- Perf Analyzer no longer supports --trace-file option.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- When using decoupled models, there is a possibility that response order as sent from the backend may not match with the order in which these responses are received by the streaming gRPC client. Note that this only applies to responses from different requests. Any responses corresponding to the same request will still be received in their expected order, relative to each other.
- The Java CAPI is known to have intermittent segfaults we're looking for a root cause.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. We recommend experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.

- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.
- Python backend support for Windows is limited and does not currently support the following features:
  - GPU tensors
  - CPU and GPU-related metrics
  - Custom execution environments
  - The model load/unload APIs

Chapter 17. Triton Inference Server Release 24.03

The Triton Inference Server container image, release 24.03, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 22.04 including Python 3.10
- NVIDIA CUDA 12.4.0.41
- NVIDIA cuBLAS 12.4.2.65
- cuDNN 9.0.0.306
- NVIDIA NCCL 2.20 (optimized for NVIDIA NVLink®)
- Amazon Labs Sockeye sequence-to-sequence framework 2.3.14 (for machine translation)
- NVIDIA TensorRT™ 8.6.3
- OpenUCX 1.15.0
- GDRCopy 2.3
- NVIDIA HPC-X 2.18
- OpenMPI 4.1.4+
- FIL
- NVIDIA DALI® 1.35
- Intel OpenVINO 2023.3.0
- DCGM 3.2.6
- TensorRT-LLM version release/0.8

- vLLM version 0.3.0

Driver Requirements

Release 24.03 is based on NVIDIA CUDA 12.4.0.41, which requires NVIDIA Driver release 545 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, and R520 drivers, which are not forward-compatible with CUDA 12.3. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 24.03 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- OpenTelemetry context for a trace started on the triton server side is now accessible from the Python Backend.
- Python Backend now supports correlation strings in BLS models.
- Triton now case-insensitively matches HTTP headers when using the header forwarding feature.
- Triton's backend API now allows users to collect per-response metrics.
- Triton now publishes request cancellations in the response statistics.
- GenAI-Perf is a new tool that facilitates LLM benchmarking and is currently available as an alpha release.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
24.03	2.44	22.04	NVIDIA CUDA 12.4.0.41	TensorRT 8.6.3
24.02	2.43		NVIDIA CUDA 12.3.2	
24.01	2.42			TensorRT 8.6.1.6
23.12	2.41			
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- There is a known issue with ONNX Runtime with TensorRT Execution Provider which causes segmentation faults when attempting to load multiple instances of a model on the same GPU. This issue is being tracked here: https://github.com/microsoft/onnxruntime/issues/20089. As a work around, users can serially load models and ensure only one model instance per gpu.
- TensorRT-LLM backend is installed with Triton 24.01 base container due to incompatibility reasons.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- When using decoupled models, there is a possibility that response order as sent from the backend may not match with the order in which these responses are

received by the streaming gRPC client. Note that this only applies to responses from different requests. Any responses corresponding to the same request will still be received in their expected order, relative to each other.

- The Java CAPI is known to have intermittent segfaults we're looking for a root cause.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. We recommend experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.
- Python backend support for Windows is limited and does not currently support the following features:
  - GPU tensors
  - CPU and GPU-related metrics
  - Custom execution environments
  - The model load/unload APIs

Chapter 18. Triton Inference Server Release 24.02

The Triton Inference Server container image, release 24.02, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 22.04 including Python 3.10
- NVIDIA CUDA 12.3.2
- NVIDIA cuBLAS 12.3.4.1
- cuDNN 9.0.0.306
- NVIDIA NCCL 2.19.4 (optimized for NVIDIA NVLink®)
- Amazon Labs Sockeye sequence-to-sequence framework 2.3.14 (for machine translation)
- NVIDIA TensorRT™ 8.6.3
- OpenUCX 1.15.0
- GDRCopy 2.3
- NVIDIA HPC-X 2.16rc4
- OpenMPI 4.1.4+
- FIL
- NVIDIA DALI® 1.34
- Intel OpenVINO 2023.3.0
- DCGM 3.2.6
- TensorRT-LLM version release/0.8

- vLLM version 0.3.0

Driver Requirements

Release 24.02 is based on CUDA 12.3.2, which requires NVIDIA Driver release 545 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, and R520 drivers, which are not forward-compatible with CUDA 12.3. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 24.02 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Added base python backend functionality for Windows.
- Removed the `Wait/Read(avg)` and `Overhead` metrics for gRPC in the Trace Summary Tool to avoid displaying inaccurate readings.
- OpenTelemetry trace mode switched to Batch Span Processor, which batches completed spans and sends them in bulk. This processor supports both size and time based batching. Size-based batching is controlled by 2 parameters: `bsp_max_export_batch_size` and `bsp_max_queue_size`, while time-based batching is controlled by `bsp_schedule_delay`.
- Refer to the appropriate column of the Frameworks Support Matrix for container image versions on which the inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
24.02	2.43	22.04	NVIDIA CUDA 12.3.2	TensorRT 8.6.3
24.01	2.42			TensorRT 8.6.1.6
23.12	2.41			
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- ONNX Runtime backend is not included with 24.02 release due to incompatibility reasons. However iGPU and Windows build assets shipped with ONNX Runtime backend.
- TensorRT-LLM backend is installed with Triton 24.01 base container due to incompatibility reasons.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- When using decoupled models, there is a possibility that response order as sent from the backend may not match with the order in which these responses are received by the streaming gRPC client. Note that this only applies to responses

from different requests. Any responses corresponding to the same request will still be received in their expected order, relative to each other.

- The Java CAPI is known to have intermittent segfaults we're looking for a root cause.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. We recommend experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.
- Python backend support for Windows is limited and does not currently support the following features:
  - GPU tensors
  - CPU and GPU-related metrics
  - Custom execution environments
  - The model load/unload APIs

Chapter 19. Triton Inference Server Release 24.01

The Triton Inference Server container image, release 24.01, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 22.04 including Python 3.10
- NVIDIA CUDA 12.3.2
- NVIDIA cuBLAS 12.3.4.1
- cuDNN 8.9.7.29
- NVIDIA NCCL 2.19.4 (optimized for NVIDIA NVLink®)
- Amazon Labs Sockeye sequence-to-sequence framework 2.3.14 (for machine translation)
- NVIDIA TensorRT™ 8.6.1.6
- OpenUCX 1.15.0
- GDRCopy 2.3
- NVIDIA HPC-X 2.16rc4
- OpenMPI 4.1.4+
- FIL
- NVIDIA DALI® 1.33
- ONNX Runtime 1.16.3
- Intel OpenVINO 2023.0.0
- DCGM 3.2.6

- TensorRT-LLM version release/0.7.1
- vLLM version 0.2.3

Driver Requirements

Release 24.01 is based on CUDA 12.3.2, which requires NVIDIA Driver release 545 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 470.57 (or later R470), 525.85 (or later R525), 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R450, R460, R510, and R520 drivers, which are not forward-compatible with CUDA 12.3. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 24.01 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Added Triton Python API for in-process integration in Python environment.
- Added command line option to retry loading failed model in the number of attempts specified.
- Added support for Context Propagation in OpenTelemetry trace mode.
- Added Triton pinned memory pool usage in reporting metrics.
- Improved error response in HTTP endpoint that HTTP status codes different than 400 may be returned to align with the error type.
- Added experimental support for serving PyTorch 2.0 models.
- The FasterTransformer backend has been deprecated as of 24.01 and will no longer be supported or released with this and future versions of Triton.
- The Model Analyzer now correctly loads and optimizes ensemble models.
- The Model Analyzer now handles the case of optimizing a model on a remote Triton server without requiring a local GPU.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
24.01	2.42	22.04	NVIDIA CUDA 12.3.2	TensorRT 8.6.1.6
23.12	2.41		NVIDIA CUDA 12.3.2	
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- When using decoupled models, there is a possibility that response order as sent from the backend may not match with the order in which these responses are

received by the streaming gRPC client. Note that this only applies to responses from different requests. Any responses corresponding to the same request will still be received in their expected order, relative to each other.

- The Java CAPI is known to have intermittent segfaults we're looking for a root cause.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. We recommend experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.

Chapter 20. Triton Inference Server Release 23.12

The Triton Inference Server container image, release 23.12, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 22.04 including Python 3.10
- NVIDIA CUDA 12.3.2
- NVIDIA cuBLAS 12.3.4.1
- cuDNN 8.9.7.29
- NVIDIA NCCL 2.19.3 (optimized for NVIDIA NVLink®)
- Amazon Labs Sockeye sequence-to-sequence framework 2.3.14 (for machine translation)
- NVIDIA TensorRT™ 8.6.1.6
- OpenUCX 1.15.0
- GDRCopy 2.3
- NVIDIA HPC-X 2.16
- OpenMPI 4.1.4+
- FIL
- NVIDIA DALI® 1.32
- ONNX Runtime 1.16.3
- Intel OpenVINO 2023.0.0
- DCGM 3.2.6

- TensorRT-LLM version release/0.7.0
- vLLM version 0.2.3

Driver Requirements

Release 23.12 is based on CUDA 12.3.2, which requires NVIDIA Driver release 545 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), or 525.85 (or later R525) 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.3. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 23.12 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Added metrics support to TRTLLM backend when running within Triton.
- Request ID will be included in opentelemetry tracing.
- For Jetson devices which support Jetpack 6.0 and above, Triton now publishes containers, based on the latest version of Jetpack, on NGC with the suffix "-igpu". These containers are:
  - **XX.YY-py3-igpu**: Much like the XX.YY-py3 container, this contains tritonserver and all supported backends for Jetson devices.
  - **XX.YY-py3-sdk-igpu**: Much like the XX.YY-py3-sdk container, this contains the Tritonclient and Triton Tools supported on Jetson devices.
- For Jetson devices which support Jetpack 5.1.2, an image has been included in this release that backports a CVE patch for the Triton model load API. This image is based off the r23.06 release and can be built from source using the 'r23.06-update-1-jp' tag. Refer to the 23.12 column of theFrameworks Support Matrix for container image versions on which the 23.12 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.12	2.41	22.04	NVIDIA CUDA 12.3.2	TensorRT 8.6.1.6
23.11	2.40		NVIDIA CUDA 12.3.0	
23.10	2.39.0		NVIDIA CUDA 12.2.2	
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Reuse-grpc-port and reuse-http-port are now properly parsed as booleans. 0 and 1 will continue to work as values. Any other integers will throw an error.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- When using decoupled models, there is a possibility that response order as sent from the backend may not match with the order in which these responses are received by the streaming gRPC client. Note that this only applies to responses

from different requests. Any responses corresponding to the same request will still be received in their expected order, relative to each other.

- The FasterTransformer backend is only officially supported for 22.12, though it can be built for Triton container versions up to 23.07.
- The Java CAPI is known to have intermittent segfaults we're looking for a root cause.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. We recommend experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.
- Model Analyzer is not able to analyze and optimize ensemble model configs due to a bug in the way composing models are loaded.

Chapter 21. Triton Inference Server Release 23.11

The Triton Inference Server container image, release 23.11, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For a complete list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

The container also includes the following:

- Ubuntu 22.04 including Python 3.10
- NVIDIA CUDA 12.3.0
- NVIDIA cuBLAS 12.3.2.1
- cuDNN 8.9.6
- NVIDIA NCCL 2.19.3 (optimized for NVIDIA NVLink®)
- Amazon Labs Sockeye sequence-to-sequence framework 2.3.14 (for machine translation)
- NVIDIA TensorRT™ 8.6.1.6
- OpenUCX 1.15.0
- GDRCopy 2.3
- NVIDIA HPC-X 2.16
- OpenMPI 4.1.4+
- FIL
- NVIDIA DALI® 1.31.0
- ONNX Runtime 1.16.3
- Intel OpenVINO 2023.0.0
- DCGM 3.2.6

- TensorRT-LLM version release/0.6.0
- vLLM version 0.2.1.post1

Driver Requirements

Release 23.11 is based on CUDA 12.3.0, which requires NVIDIA Driver release 545 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), or 525.85 (or later R525) 535.86 (or later R535), or 545.23 (or later R545).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.3. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 23.11 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Starting with the 23.11 release, Triton containers supporting iGPU architectures are published, and run on Jetson devices. Please refer to the Frameworks Support Matrix for information regarding which iGPU hardware/software is supported by which container.
- Implicit state management has been enhanced to support growing buffers and use a single buffer for both input and output states.
- Sequence batcher has been enhanced to support iterative scheduling.
- The backend API has been enhanced to support rescheduling a request.Currently, only Python backend and Custom C++ backends support request rescheduling.
- TRT-LLM backend now supports request cancellation.
- Configuration of a vLLM backend model can now be auto-completed by Triton. The user just needs to pass backend: "vllm" to leverage the auto-complete feature.
- Python backend now supports parameters in BLS requests.
- Python backend GPU tensor support has been improved to provide better performance.

- A new tutorial demonstrating how to deploy LLaMa2 using TRT-LLM has been added.
- Added benchmarking script for profiling LLMs using Perf Analyzer
- The HTTP endpoint has been enhanced to support access restriction.
- Secure Deployment Guide has been added to provide guidance on deploying Triton securely.
- The client model loading API no longer allows uploading files outside the model repository.
- DCGM version has been upgraded to 3.2.6.
- The Kubernetes Deploy example now supports Kubernetes' new StartupProbe to allow Triton pods time to finish startup before running health probes.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.11	2.40	22.04	NVIDIA CUDA 12.3.0	TensorRT 8.6.1.6
23.10	2.39.0		NVIDIA CUDA 12.2.2	
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- When using the generate streaming endpoint, Triton will segfault if the client closes the connection before all responses have been generated. The fix will be available in the next release.
- Reuse-grpc-port and reuse-http-port are now properly parsed as booleans. 0 and 1 will continue to work as values. Any other integers will throw an error.
- TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- When using decoupled models, there is a possibility that response order as sent from the backend may not match with the order in which these responses are received by the streaming gRPC client. Note that this only applies to responses from different requests. Any responses corresponding to the same request will still be received in their expected order, relative to each other.
- The FasterTransformer backend is only officially supported for 22.12, though it can be built for Triton container versions up to 23.07.
- The Java CAPI is known to have intermittent segfaults we're looking for a root cause.
- Some systems which implement malloc() may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. We recommend experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Triton Client PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA. The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. Refer to pytorch/pytorch#66930 for more information.

- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.

Chapter 22. Triton Inference Server Release 23.10

The Triton Inference Server container image, release 23.10, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 23.10 is based on CUDA 12.2.2, which requires NVIDIA Driver release 535 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), or 525.85 (or later R525), or 535.86 (or later R535).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.2. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 23.10 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Added support for handling client-side request cancellation in Triton server and backends. (server docs, client docs).
- Triton can deploy supported models on the vLLM engine using the new vLLM backend. A new container with vLLM backend is available on NGC for 23.10.
- Triton now supports the TensorRT-LLM backend. This backend uses the Nvidia TensorRT-LLM, which replaces the Fastertransformer backend. A new container with TensorRT-LLM backend is available on NGC for 23.10.
- AddedGenerate extension (beta) which provides better REST APIs for inference on Large Language Models.
- New tutorials with respect to how to run vLLM with the new REST API, how to run Llama2 with TensorRT-LLM backend, and how to run with HuggingFace models in the tutorial repo.
- Support Scalar I/O in ONNXRuntime backend.
- Added support for writing custom backends in python, a.k.a. Python-based backends.
- Refer to the 23.10 column of the Frameworks Support Matrix for container image versions on which the 23.10 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.10	2.39.0	22.04	NVIDIA CUDA 12.2.2	TensorRT 8.6.1.6
23.09	2.38.0		NVIDIA CUDA 12.2.1	
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	TensorRT 8.0.2.2 for ARM SBSA Linux
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- For its initial release, the TensorRT-LLM backend provides limited support of Triton extensions and features.
- The TensorRT-LLM backend may core dump on server shutdown. This impacts server teardown only and will not impact inferencing.
- When a model uses a backend which is not found, Triton would reference the missing backend as `backend_name /model.py" in the error message. This is already fixed for future releases.
- When using decoupled models, there is a possibility that response order as sent from the backend may not match with the order in which these responses are received by the streaming gRPC client. Note that this only applies to responses from different requests. Any responses corresponding to the same request will still be received in their expected order, relative to each other.
- The FasterTransformer backend is only officially supported for 22.12, though it can be built for Triton container versions up to 23.07.
- The Java CAPI is known to have intermittent segfaults.
- Some systems which implement `malloc()` may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. We recommend experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273

- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.

Chapter 23. Triton Inference Server Release 23.09

The Triton Inference Server container image, release 23.09, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 23.09 is based on CUDA 12.2.1, which requires NVIDIA Driver release 535 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), or 525.85 (or later R525), or 535.86 (or later R535).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.2. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 23.09 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Triton now has Python bindings for the C API. Please refer to this PR for usage.
- Triton now forwards request parameters to each of the composing models of an ensemble model.
- The Filesystem API now supports named temporary cache directories when downloading models using the repository agent.
- Added the number of requests currently in the queue to the metrics API. Documentation can be found here.
- Python backend models can now respond with error codes in addition to error messages.
- TensorRT backend now supports TensortRT version compatibility across models generated with the same major version of TensorRT. Use the `--backend-config=tensorrt,--version-compatible=true` flag to enable this feature.
- Triton's backend API now supports accessing the inference response outputs by name or by index. See the new API here.
- The Python backend now supports loading Pytorch models directly. This feature is experimental and should be treated as Beta.
- Fixed an issue where if the user didn't call `SetResponseReleaseCallback`, canceling a new request could cancel the old response factory as well. Now when canceling a request which is being re-used, a new response factory is created for each inference.
- Refer to the 23.09 column of the Frameworks Support Matrix for container image versions on which the 23.09 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.09	2.38.0	22.04	NVIDIA CUDA 12.2.1	TensorRT 8.6.1.6
23.08	2.37.0			
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- When using decoupled models, there is a possibility that response order as sent from the backend may not match with the order in which these responses are received by the streaming gRPC client. Note that this only applies to responses from different requests. Any responses corresponding to the same request will still be received in their expected order, relative to each other.
- The FasterTransformer backend is only officially supported for 22.12, though it can be built for Triton container versions up to 23.07.
- The Java CAPI is known to have intermittent segfaults.
- Some systems which implement `malloc()` may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. We recommend experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273
- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.

- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- When cloud storage (AWS, GCS, AZURE) is used as a model repository and a model has multiple versions, Triton creates an extra local copy of the cloud model's folder in the temporary directory, which is deleted upon server's shutdown.

Chapter 24. Triton Inference Server Release 23.08

The Triton Inference Server container image, release 23.08, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 23.08 is based on CUDA 12.2.1, which requires NVIDIA Driver release 535 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), or 525.85 (or later R525), or 535.86 (or later R535).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.2. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 23.08 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Triton can load model instances in parallel for supporting backends. See TRITONBACKEND_BackendAttributeSetParallelModelInstanceLoading for more details. As of 23.08, only python and onnxruntime backends support loading model instances in parallel.
- Python backend models can capture trace for composing child models when executing BLS requests.
- Triton OpenTelemetry Tracing exposes resource settings which can be used to configure the service name and version.
- Python backend supports directly loading and serving PyTorch models with torch.compile.
- Exposed preserve_ordering field to oldest strategy sequence batcher.
- Refer to the 23.08 column of the Frameworks Support Matrix for container image versions on which the 23.08 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.08	2.37.0	22.04	NVIDIA CUDA 12.2.1	TensorRT 8.6.1.6
23.07	2.36.0		NVIDIA CUDA 12.1.1	
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Triton uses OpenTelemetry CPP library version, which can cause Triton to crash, when OpenTelemetry's exporter timeouts.
- When using decoupled models, there is a possibility that response order as sent from the backend may not match with the order in which these responses are received by the streaming gRPC client.
- The FasterTransformer backend is only officially supported for 22.12, though it can be built for Triton container versions up to 23.07.
- The Java CAPI is known to have intermittent segfaults.
- Some systems which implement `malloc()` may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. We recommend experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273
- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.

Chapter 25. Triton Inference Server Release 23.07

The Triton Inference Server container image, release 23.07, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 23.07 is based on CUDA 12.1.1, which requires NVIDIA Driver release 530 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), 515.65 (or later R515), 525.85 (or later R525), or 530.30 (or later R530).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.1. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 23.07 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- PyTorch backend supports implicit state management.
- Python backend supports direct serving of TensorFlow SavedModel.
- Python backend supports unpacked Conda execution environment.
- Python backend added the model loading APIs for BLS usage.
- Triton OpenTelemetry trace mode supports ensemble model tracing.
- Triton Python client supports DLPack tensors in CUDA shared memory utilities.
- Triton supports the S3 model repository that contains more than 1000 files.
- Added Java binding of the Triton in-process C++ API.
- Refer to the 23.07 column of the Frameworks Support Matrix for container image versions on which the 23.07 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.07	2.36.0	22.04	NVIDIA CUDA 12.1.1	TensorRT 8.6.1.6
23.06	2.35.0			
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- The FasterTransformer backend build only works with Triton 23.04 and older releases.

- Some systems which implement `malloc()` may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. We recommend experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273
- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.

Chapter 26. Triton Inference Server Release 23.06

The Triton Inference Server container image, release 23.06, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 23.06 is based on CUDA 12.1.1, which requires NVIDIA Driver release 530 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), 515.65 (or later R515), 525.85 (or later R525), or 530.30 (or later R530).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.1. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 23.06 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Support for KIND_MODEL instance type has been extended to PyTorch backend.
- The gRPC clients can now indicate whether they want to receive the flags associated with each response. This can help the clients to programmatically determine when all the responses for a given request have been received on the client side for decoupled models.
- Added beta support for using Redis as a cache for inference requests.
- The statistics extension now includes the memory usage of the loaded models This statistics is currently implemented only for TensorRT and ONNXRuntime backends.
- Added support for batch inputs in ragged batching for PyTorch backend.
- Added serial sequences for Perf Analyzer.
- Refer to the 23.06 column of the Frameworks Support Matrix for container image versions on which the 23.06 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.06	2.35.0	22.04	NVIDIA CUDA 12.1.1	TensorRT 8.6.1.6
23.05	2.34.0			TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- The FasterTransformer backend build only works with Triton 23.04 and older releases.

- OpenVINO 2022.1 is used in the OpenVINO backend and the OpenVINO execution provider for the Onnxruntime Backend. OpenVINO 2022.1 is not officially supported on Ubuntu 22.04 and should be treated as beta.
- Some systems which implement `malloc()` may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. We recommend experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273
- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.

Chapter 27. Triton Inference Server Release 23.05

The Triton Inference Server container image, release 23.05, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 23.05 is based on CUDA 12.1.1, which requires NVIDIA Driver release 530 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), 515.65 (or later R515), 525.85 (or later R525), or 530.30 (or later R530).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.1. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 23.05 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Python backend supports Custom Metrics allowing users to define and report counters and gauges similar to the C API.
- Python Triton Client defines the Triton Client Plugin API allowing users to register custom plugins to add or modify request headers. This feature is in beta and is subject to change in future releases.
- Improved performance of model instance creation/removal. When the model instance group is the only model configuration change, Triton will update the model with the number of instances needed rather than reloading the model. This feature is limited to non-sequence models only. Read more about this feature here in bullet point four.
- Added new command line option `--metrics-address=<address>` allowing the metrics server to bind to a different address than the default `0.0.0.0`.
- Reduced the default number of model load threads from 2*(number of CPU cores) to 4. This eliminates Triton hitting resource limits on systems with large CPU core counts. Use the `--model-load-thread-count` command line option to change this default.
- Added support for DLPack Python specification in Python backend.
- Refer to the 23.05 column of the Frameworks Support Matrix for container image versions on which the 23.05 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.05	2.34.0	22.04	NVIDIA CUDA 12.1.1	TensorRT 8.6.1.2
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Tensorflow backend no longer supports TensorFlow version 1.
- OpenVINO 2022.1 is used in the OpenVINO backend and the OpenVINO execution provider for the Onnxruntime Backend. OpenVINO 2022.1 is not officially supported on Ubuntu 22.04 and should be treated as beta.
- Some systems which implement `malloc()` may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc and jemalloc are installed in the Triton container and can be used by specifying the library in LD_PRELOAD. We recommend experimenting with both tcmalloc and jemalloc to determine which one works better for your use case.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273
- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.

Chapter 28. Triton Inference Server Release 23.04

The Triton Inference Server container image, release 23.04, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 23.04 is based on CUDA 12.1.0, which requires NVIDIA Driver release 530 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), 515.65 (or later R515), 525.85 (or later R525), or 530.30 (or later R530).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.1. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 23.04 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, NVIDIA Hopper™, and NVIDIA Ada Lovelace architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Triton can now load models concurrently reducing the server start-up times. Sequence batcher with direct scheduling strategy now includes experimental support for schedule policy.
- Triton's ragged batching support has been extended to the PyTorch backend.
- Triton can now forward HTTP/GRPC headers as inference request parameters to the backend.
- Triton python backend's business logic scripting now allows developers to select a specific device to receive output tensors from a BLS call.
- Triton latency metrics can now be obtained as configurable quantiles over a sliding time window using experimental metrics summary support.
- Users can now restrict the access of protocols on a given Triton endpoint.
- Triton now provides limited support for tracing inference requests using OpenTelemetry Trace APIs.
- Model Analyzer now supports BLS Models.
- Refer to the 23.04 column of the Frameworks Support Matrix for container image versions on which the 23.04 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.04	2.33.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.6.1
23.03	2.32.0			TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Tensorflow backend no longer supports TensorFlow version 1.
- Triton Inferentia guide is out of date. Some users have reported issues with running Triton on AWS Inferentia instances.
- Some systems which implement `malloc()` may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc is installed in the Triton container and can be used by specifying the library in LD_PRELOAD.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273
- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.

Chapter 29. Triton Inference Server Release 23.03

The Triton Inference Server container image, release 23.03, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 23.03 is based on CUDA 12.1.0, which requires NVIDIA Driver release 530 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), 515.65 (or later R515), 525.85 (or later R525), or 530.30 (or later R530).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.1. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 23.03 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, and NVIDIA Hopper™ architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Added the Parameters Extension which allows an inference request to provide custom parameters that cannot be provided as inputs. These parameters can be used in the python backend as described here.
- Added support for models that use decoupled API for Business Scripting Logic (BLS) in Python backend. Examples can be found here.
- The same model name can be used across different repositories if the `--model-namespacing` flag is set.
- Triton's Response Cache feature has been converted internally to a shared library implementation of the new TRITONCACHE APIs, similar to how backends and repo agents are implemented today. The default cache implementation is local_cache, which is equivalent to the fixed-size in-memory buffer implementation used before. The `--response-cache-byte-size` flag will continue to function in the same way, but the `--cache-config` flag will be the preferred method of cache configuration moving forward. For more information, see the cache documentation here.
- Triton's trace tool now supports tracing for `request_id`.
- Refer to the 23.03 column of the Frameworks Support Matrix for container image versions on which the 23.03 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.03	2.32.0	20.04	NVIDIA CUDA 12.1.0	TensorRT 8.5.3
23.02	2.31.0		NVIDIA CUDA 12.0.1	
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Support for TensorFlow1 will be removed starting from 23.04.
- Triton Inferentia guide is out of date. Some users have reported issues with running Triton on AWS Inferentia instances.
- Some systems which implement `malloc()` may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc is installed in the Triton container and can be used by specifying the library in LD_PRELOAD.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273
- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.

Chapter 30. Triton Inference Server Release 23.02

The Triton Inference Server container image, release 23.02, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 23.02 is based on CUDA 12.0.1, which requires NVIDIA Driver release 525 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), 515.65 (or later R515), or 525.85 (or later R525).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, R460, and R520 drivers, which are not forward-compatible with CUDA 12.0. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 23.02 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, and NVIDIA Hopper™ architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Support for ensemble models in Model Analyzer.
- Support for GRPC Standard Health Check Protocol.
- Fixed intermittent hangs during model loading for Python backend.
- Refer to the 23.02 column of the Frameworks Support Matrix for container image versions on which the 23.02 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.02	2.31.0	20.04	NVIDIA CUDA 12.0.1	TensorRT 8.5.3
23.01	2.30.0			TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

Here are the known issues in this release:

- In some rare cases Triton's TensorRT backend might overwrite input tensors while they are still in use which leads to corrupt input data being used for inference with TensorRT models. If you encounter accuracy issues with your TensorRT model, you can work-around the issue byenabling the output_copy_stream option in your model's configuration.
- Some systems which implement `malloc()` may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc is installed in the Triton container and can be used by specifying the library in LD_PRELOAD.
- When using a custom operator for the PyTorch backend, the operator may not be loaded due to undefined Python library symbols. This can be work-around by specifying Python library in LD_PRELOAD.

- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273
- Perf Analyzer stability criteria has been changed which may result in reporting instability for scenarios that were previously considered stable. This change has been made to improve the accuracy of Perf Analyzer results. If you observe this message, it can be resolved by increasing the `--measurement-interval` in the time windows mode or `--measurement-request-count` in the count windows mode.
- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.

Chapter 31. Triton Inference Server Release 23.01

The Triton Inference Server container image, release 23.01, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 23.01 is based on CUDA 12.0.1, which requires NVIDIA Driver release 525 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), 515.65 (or later R515), or 525.85 (or later R525).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 12.0. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 23.01 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, and NVIDIA Hopper™ architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Refer to the 23.01 column of the Frameworks Support Matrix for container image versions on which the 23.01 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
23.01	2.30.0	20.04	NVIDIA CUDA 12.0.1	TensorRT 8.5.2.2
22.12	2.29.0		NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

Here are the known issues in this release:

- In some rare cases Triton's TensorRT backend might overwrite input tensors while they are still in use which leads to corrupt input data being used for inference with TensorRT models. If you encounter accuracy issues with your TensorRT model, you can work-around the issue byenabling the output_copy_stream option in your model's configuration.
- Some systems which implement `malloc()` may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc is installed in the Triton container and can be used by specifying the library in LD_PRELOAD.
- When using a custom operator for the PyTorch backend, the operator may not be loaded due to undefined Python library symbols. This can be work-around by specifying Python library in LD_PRELOAD.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches

what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273

- Perf Analyzer stability criteria has been changed which may result in reporting instability for scenarios that were previously considered stable. This change has been made to improve the accuracy of Perf Analyzer results. If you observe this message, it can be resolved by increasing the `--measurement-interval` in the time windows mode or `--measurement-request-count` in the count windows mode.
- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.

Chapter 32. Triton Inference Server Release 22.12

The Triton Inference Server container image, release 22.12, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 22.12 is based on CUDA 11.8.0, which requires NVIDIA Driver release 520 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), or 515.65 (or later R515).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.8. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 22.12 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, and NVIDIA Hopper™ architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Improvements to container and non-container builds on Windows.
- Concurrent calls to the model load API will be processed in parallel improving model load times.
- Refer to the 22.12 column of the Frameworks Support Matrix for container image versions on which the 22.12 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.12	2.29.0	20.04	NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.11	2.28.0			
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

Here are the known issues in this release:

- In some rare cases Triton might overwrite input tensors while they are still in use which leads to corrupt input data being used for inference with TensorRT models. If you encounter accuracy issues with your TensorRT model, you can work-around the issue byenabling the output_copy_stream option in your model's configuration.
- Some systems which implement `malloc()` may not release memory back to the operating system right away causing a false memory leak. This can be mitigated by using a different malloc implementation. Tcmalloc is installed in the Triton container and can be used by specifying the library in LD_PRELOAD.
- When using a custom operator for the PyTorch backend, the operator may not be loaded due to undefined Python library symbols. This can be work-around by specifying Python library in LD_PRELOAD.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273

- Perf Analyzer stability criteria has been changed which may result in reporting instability for scenarios that were previously considered stable. This change has been made to improve the accuracy of Perf Analyzer results. If you observe this message, it can be resolved by increasing the `--measurement-interval` in the time windows mode or `--measurement-request-count` in the count windows mode.
- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.

Chapter 33. Triton Inference Server Release 22.11

The Triton Inference Server container image, release 22.11, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 22.11 is based on CUDA 11.8.0, which requires NVIDIA Driver release 520 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), or 515.65 (or later R515).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.8. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 22.11 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, and NVIDIA Hopper™ architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Support for new TensorRT 8.5 features, including:
  - UINT8 I/O
  - “Data dependent dynamic shapes" operators (i.e. ONNX NMS and NonZero operations)
- Support for execution environment paths outside model directory. This can be done via the `EXECUTION_ENV_PATH` parameter in `config.pbtxt`. Refer to the python backend README for known limitations.
- Refer to the 22.11 column of the Frameworks Support Matrix for container image versions on which the 22.11 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.11	2.28.0	20.04	NVIDIA CUDA 11.8.0	TensorRT 8.5.1
22.10	2.27.0			TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

Here are the known issues in this release:

- Triton will not release for Jetson in 22.11. Please use the latest version, 22.10 (https://github.com/triton-inference-server/server/releases/tag/v2.27.0), instead.
- In some rare cases Triton might overwrite input tensors while they are still in use which leads to corrupt input data being used for inference with TensorRT models. If you encounter accuracy issues with your TensorRT model, you can work-around the issue byenabling the output_copy_stream option in your model's configuration.
- When using a custom operator for the PyTorch backend, the operator may not be loaded due to undefined Python library symbols. This can be work-around by specifying Python library in LD_PRELOAD.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about

the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273

- Perf Analyzer stability criteria has been changed which may result in reporting instability for scenarios that were previously considered stable. This change has been made to improve the accuracy of Perf Analyzer results. If you observe this message, it can be resolved by increasing the `--measurement-interval` in the time windows mode or `--measurement-request-count` in the count windows mode.
- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.

Chapter 34. Triton Inference Server Release 22.10

The Triton Inference Server container image, release 22.10, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 22.10 is based on CUDA 11.8.0, which requires NVIDIA Driver release 520 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), or 515.65 (or later R515).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.8. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 22.10 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, and NVIDIA Hopper™ architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Added an example to demonstrate the use of JAX in Python models.
- Improved and enhanced Server Wrapper API to include missing features such as decoupled model and tracing support.
- Multiple concurrent models can be profiled and analyzed by Model Analyzer. Refer to Multi-Model Search Mode for additional details.
- Refer to the 22.10 column of the Frameworks Support Matrix for container image versions on which the 22.10 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.10	2.27.0	20.04	NVIDIA CUDA 11.8.0	TensorRT 8.5 EA
22.09	2.26.0			
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

Here are the known issues in this release:

- In some rare cases, Triton might overwrite input tensors while they are still in use which leads to corrupt input data being used for inference with TensorRT models. If you encounter accuracy issues with your TensorRT model, you can work around the issue byenabling the output_copy_stream option in your model's configuration.
- When using a custom operator for the PyTorch backend, the operator may not be loaded due to undefined Python library symbols. This can be work-around by specifying Python library in LD_PRELOAD.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273
- Perf Analyzer stability criteria has been changed which may result in reporting instability for scenarios that were previously considered stable. This change has been made to improve the accuracy of Perf Analyzer results. If you observe this message, it can be resolved by increasing the `--measurement-interval` in the

time windows mode or `--measurement-request-count` in the count windows mode.

- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.

Chapter 35. Triton Inference Server Release 22.09

The Triton Inference Server container image, release 22.09, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 22.09 is based on CUDA 11.8.0, which requires NVIDIA Driver release 520 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), 510.47 (or later R510), or 515.65 (or later R515).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.8. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 22.09 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, NVIDIA Ampere architecture, and NVIDIA Hopper™ architecture families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Added developer tools Github repository that provides a simplified interface for users to interact with the Triton Core shared library. These developer tools are in beta and are subject to change.
- Added CPU metrics reporting in Triton's Prometheus metrics endpoint.
- Added logging protocol extension for users to change logging configuration dynamically.
- Users can specify the custom plugins to be loaded for TensorRT backend through command line option in addition to `LD_PRELOAD`.
- Enabled auto-completion for OpenVINO backend.
- Enabled Python backend to log messages through Triton's logger.
- Added quick search algorithm to Model Analyzer to drastically reduce search time.
- Added GPU metrics gathering to Perf Analyzer, which is also used by Model Analyzer to improve accuracy of those metrics.
- Refer to the 22.09 column of the Frameworks Support Matrix for container image versions on which the 22.09 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.09	2.26.0	20.04	NVIDIA CUDA 11.8.0	TensorRT 8.5 EA
22.08	2.25.0		NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

Here are the known issues in this release:

- In certain rare cases with specific backends, Triton server may crash with segmentation fault when exiting. Preliminary analysis shows that there might be a race condition in clean up of backend/model/instance state objects. Exact root cause is still unknown.
- Triton's TensorRT support depends on the CUDA event synchronization. In some rare cases the events may be triggered earlier than expected, causing Triton to overwrite input tensors while they are still in use and leading to corrupt input data being used for inference. If you encounter accuracy issues with your TensorRT model, you can work-around the issue by enabling the output_copy_stream option in your model's configuration.

- When using a custom operator for the PyTorch backend, the operator may not be loaded due to undefined Python library symbols. This can be work-around by specifying Python library in LD_PRELOAD.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273
- Perf Analyzer stability criteria has been changed which may result in reporting instability for scenarios that were previously considered stable. This change has been made to improve the accuracy of Perf Analyzer results. If you observe this message, it can be resolved by increasing the `--measurement-interval` in the time windows mode or `--measurement-request-count` in the count windows mode.
- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.

Chapter 36. Triton Inference Server Release 22.08

The Triton Inference Server container image, release 22.08, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 22.08 is based on CUDA 11.7 Update 1, which requires NVIDIA Driver release 515 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), or 510.47 (or later R510).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.7. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 22.08 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, and NVIDIA Ampere Architecture GPU families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- New support for multiple cloud credentials has been enabled. This feature is in beta and is subject to change.
- Models using custom backends which implement auto-complete configuration, can be loaded without an explicit `config.pbtxt` file if they are named in the form `<model_name>.<backend_name>`.
- Users can specify a maximum memory limit when loading models onto the GPU with the new `--model-load-gpu-limit` tritonserver option and the `TRITONSERVER_ServerOptionsSetModelLoadDeviceLimit` C API function.
- Added new documentation, Performance Tuning, with a step by step guide to optimize models for production.
- From this release onwards, Triton will default to TensorFlow version 2.X. TensorFlow version 1.X can still be manually specified using backend config.
- PyTorch backend has improved performance by using a separate CUDA Stream for each model instance when the instance kind is GPU.
- Model Analyzer's `profile` subcommand now analyzes the results after Profile is completed. Usage of the `Analyze` subcommand is deprecated. Refer to the Model Analyzer documentation for further details.
- Refer to the 22.08 column of the Frameworks Support Matrix for container image versions on which the 22.08 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.08	2.25.0	20.04	NVIDIA CUDA 11.7.1	TensorRT 8.4.2.4
22.07	2.24.0		NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

Here are the known issues in this release:

- There is no Jetpack release for 22.08; the latest JetPack release is 22.07.
- Auto-complete may cause an increase in server start time. To avoid a start time increase, users can provide the full model configuration and launch the server with `--disable-auto-complete-config`.
- When auto-completing some model configs, backends may generate a model config even though there is not enough metadata (for example, Graphdef models for TensorFlow Backend). The user will see the model successfully load but fail to inference. In this case, the user should provide the full model configuration for these models or use the `--disable-auto-complete-config` CLI option to show which models fail to load.
- Auto-complete does not support PyTorch models due to lack of metadata in the model. It can only verify that the number of inputs and the input names matches

what is specified in the model configuration. There is no model metadata about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273

- Running inference on multiple TensorRT model instances in Triton may fail with signal(6). The issue is expected to be fixed in a future release. Details can be found in https://github.com/triton-inference-server/server/issues/4566.
- Perf Analyzer stability criteria has been changed which may result in reporting instability for scenarios that were previously considered stable. This change has been made to improve the accuracy of Perf Analyzer results. If you observe this message, it can be resolved by increasing the `--measurement-interval` in the time windows mode or `--measurement-request-count` in the count windows mode.
- Unlike previously noted, 22.07 is the last release that defaults to TensorFlow version 1. From 22.08 onwards Triton will change the default TensorFlow version to 2.x.
- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- Model Analyzer reported values for GPU utilization and GPU power are known to be inaccurate and generally lower than reality.

Chapter 37. Triton Inference Server Release 22.07

The Triton Inference Server container image, release 22.07, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 22.07 is based on CUDA 11.7 Update 1 Preview, which requires NVIDIA Driver release 515 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), or 510.47 (or later R510).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.7. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 22.07 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, and NVIDIA Ampere Architecture GPU families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Auto-Complete is enabled by default. The `--strict-model-config` CLI option has been soft deprecated; use the new `--disable-auto-complete-config` option instead.
- New example backend demonstrating Business Logic Scripting in C++.
- Users can provide values for `init_ops` variables in TensorFlow TF1.x GraphDef models through JSON file.
- New asyncio compatible API to the Python GRPC/HTTP APIs.
- Added a thread pool to reduce service downtime for concurrently loading models. The thread pool size is configurable with the new `--model-load-thread-count` tritonserver option. You can find more information here.
- Model Analyzer now doesn't require `config.pbtxt` file for models that can be auto-completed in Triton.
- Refer to the 22.07 column of the Frameworks Support Matrix for container image versions on which the 22.07 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.07	2.24.0	20.04	NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.4.1
22.06	2.23.0			TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

Here are the known issues in this release:

- JetPack release will be published later in the month in order to align with Jetpack SDK public availability.
- Auto-complete could cause an increase in server start time. To avoid a start time increase, users should provide the full model configuration
- When auto-completing some model configs, backends may generate a model config even though there is not enough metadata (for example, Graphdef models for TensorFlow Backend). The user will see the model successfully load but fail to inference. In this case, the user should provide the full model configuration for these models or use the `--disable-auto-complete-config` CLI option to show which models fail to load.
- Can't do autocomplete for PyTorch models, due to not enough metadata.Can only verify that the number of inputs is correct and the input names match what is specified in the model configuration. There is no info about the number of outputs and datatypes. Related PyTorch bug:https://github.com/pytorch/pytorch/issues/38273.
- Running inference on multiple TensorRT model instances in Triton may fail with signal(6). The issue is expected to be fixed in a future release. Details can be found in https://github.com/triton-inference-server/server/issues/4566.
- Perf Analyzer stability criteria has been changed which may result in reporting instability for scenarios that were previously considered stable. This change has

been made to improve the accuracy of Perf Analyzer results. If you observe this message, it can be resolved by increasing the `--measurement-interval` in the time windows mode or `--measurement-request-count` in the count windows mode.

- Unlike previously noted, 22.07 is the last release that defaults to TensorFlow version 1. From 22.08 onwards Triton will change the default TensorFlow version to 2.x.
- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- Model Analyzer reported values for GPU utilization and GPU power are known to be inaccurate and generally lower than reality.

Chapter 38. Triton Inference Server Release 22.06

The Triton Inference Server container image, release 22.06, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 22.06 is based on CUDA 11.7 Update 1 Preview, which requires NVIDIA Driver release 515 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), or 510.47 (or later R510).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.7. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 22.06 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, and NVIDIA Ampere Architecture GPU families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Triton Arm (SBSA) containers are now out of beta.
- Auto-generated model configuration enables dynamic batching in supported models by default.
- Python backend models now support auto-generated model configuration.
- Decoupled API support in Python Backend model is out of beta.
- Updated I/O tensors naming convention for serving TorchScript models via PyTorch backend.
- Improvements to Perf Analyzer stability and profiling logic.
- Refer to the 22.06 column of the Frameworks Support Matrix for container image versions on which the 22.06 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.06	2.23.0	20.04	NVIDIA CUDA 11.7 Update 1 Preview	TensorRT 8.2.5
22.05	2.22.0		NVIDIA CUDA 11.7.0	
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

Here are the known issues in this release:

- Perf Analyzer stability criteria has been changed which may result in reporting instability for scenarios that were previously considered stable. This change has been made to improve the accuracy of Perf Analyzer results. If you observe this message, it can be resolved by increasing the `--measurement-interval` in the time windows mode or `--measurement-request-count` in the count windows mode.
- 22.06 is the last release that defaults to TensorFlow version 1. From 22.07 onwards Triton will change the default TensorFlow version to 2.X.
- Triton Client PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton Client library for Arm SBSA.

  The correct client wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- Running a PyTorch TorchScript model using the PyTorch backend, where multiple instances of a model are configured can lead to a slowdown in model execution due to the following PyTorch issue:

https://github.com/pytorch/pytorch/issues/27902

- Starting from 22.02, the Triton container, which uses the 22.02 or above PyTorch container, will report an error during model loading in the PyTorch backend when using scripted models that were exported in the legacy format (using our 19.09 or previous PyTorch NGC containers corresponding to PyTorch 1.2.0 or previous releases).

  To load the model successfully in Triton, you need to export the model again by using a recent version of PyTorch.

Chapter 39. Triton Inference Server Release 22.05

The Triton Inference Server container image, release 22.05, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 22.05 is based on CUDA 11.7, which requires NVIDIA Driver release 515 or later. However, if you are running on a data center GPU (for example, T4 or any other data center GPU), you can use NVIDIA driver release 450.51 (or later R450), 470.57 (or later R470), or 510.47 (or later R510).

The CUDA driver's compatibility package only supports particular drivers. Thus, users should upgrade from all R418, R440, and R460 drivers, which are not forward-compatible with CUDA 11.7. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 22.05 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, and NVIDIA Ampere Architecture GPU families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Triton In-Process API is now available in Java.
- Python backend supports the decoupled API as BETA release.
- Models can now load from file content provided during the Triton Server API invocation.
- BF16 data type is now supported.
- PyTorch backend now supports 1-dimensional String I/O.
- In model control mode EXPLICIT, loading all models at startup is supported.
- You may specify customized GRPC channel settings in the GRPC client library.
- Triton In-Process API supports dynamic model repository registration.
- Improved build pipeline in `build.py` and generate build scripts used for pipeline examination.
- ONNX Runtime backend is updated to ONNX Runtime version 1.11.1 in both Ubuntu and Windows versions of Triton.
- Refer to the 22.05 column of the Frameworks Support Matrix for container image versions on which the 22.05 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.05	2.22.0	20.04	NVIDIA CUDA 11.7.0	TensorRT 8.2.5.1
22.04	2.21.0		NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and for x86 Linux and SBSA

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

Here are the known issues in this release:

- A protobuf python package version that satisfies protobuf>=3.5.0,<3.20 must be installed before installing the Triton Arm SBSA wheels or any tritonclient version of 2.22.0 or earlier. Tritonclient versions of 2.22.3 or newer for Jetson, x86, and Windows will work normally.
- Triton PIP wheels for Arm SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton for Arm SBSA.

  The correct wheel file can be pulled directly from the Arm SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- Running a PyTorch TorchScript model using the PyTorch backend, where multiple instances of a model are configured can lead to a slowdown in model execution due to the following PyTorch issue:

  pytorch/pytorch#27902

- Starting in 22.02, the Triton container, which uses the 22.04 PyTorch container, will report an error during model loading in the PyTorch backend when using scripted models that were exported in the legacy format (using our 19.09 or previous PyTorch NGC containers corresponding to PyTorch 1.2.0 or previous releases).

  To load the model successfully in Triton, you need to export the model again by using a recent version of PyTorch.

Chapter 40. Triton Inference Server Release 22.04

The Triton Inference Server container image, release 22.04, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 22.04 is based on NVIDIA CUDA® 11.6.2, which requires NVIDIA Driver release 510 or later. However, if you are running on a Data Center GPU (for example, T4 or any other Tesla board), use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51 (or later R450), 460.27 (or later R460), or 470.57 (or later R470). The CUDA driver's compatibility package only supports specific drivers. For a complete list of supported drivers, see CUDA Application Compatibility. For more information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 22.04 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, and NVIDIA Ampere Architecture GPU families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- You can now specify a customized temp directory with the --tmp-dir argument to build.py during the container build.
- You can now send a raw binary request to eliminate the need for the inference heder specification.
- Ensembles now recognize optional inputs.
- You can now add custom metrics to the existing Triton metrics endpoint in their custom backends and applications using the Triton C API. Documentation can be found here.
- Official support for multiple cloud repositories.

  This support includes the same and different cloud storage providers, for example an instance of Triton can load models from two S3 buckets, two GCS buckets, and two Azure Storage containers.
- ONNX Runtime backend now uses execution providers when available when autocomplete is enabled.

  This enhancement fixes the previous behavior where the backend always used the CPU execution provider.
- The build.py and compose.py now support PyTorch and TensorFlow 1 backends for the CPU-only builds.
- Refer to the 22.04 column of the Frameworks Support Matrix for container image versions on which the 22.04 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.04	2.21.0	20.04	NVIDIA CUDA 11.6.2	TensorRT 8.2.4.2 and for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.03	2.20.0		NVIDIA CUDA 11.6.1	TensorRT 8.2.3 and

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
				for x86 Linux and SBSA TensorRT 8.4.0 for JetPack/ Jetson
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

Here are the known issues in this release:

- Triton PIP wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton for ARM SBSA.

  The correct wheel file can be pulled directly from the ARM SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to pytorch/pytorch#66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- Running a PyTorch TorchScript model using the PyTorch backend, where multiple instances of a model are configured can lead to a slowdown in model execution due to the following PyTorch issue:

  pytorch/pytorch#27902

- Starting in 22.02, the Triton container, which uses the 22.04 PyTorch container, will report an error during model loading in the PyTorch backend when using scripted models that were exported in the legacy format (using our 19.09 or previous PyTorch NGC containers corresponding to PyTorch 1.2.0 or previous releases).

  To successfully load the model in Triton, you need to export the model again by using a recent version of PyTorch.
- Starting in 22.04, Model Analyzer will not sort results correctly when running analysis. To work around this issue, re-profile on main (or an earlier version) and then re-run the analysis step.

Chapter 41. Triton Inference Server Release 22.03

The Triton Inference Server container image, release 22.03, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the container includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 22.03 is based on NVIDIA CUDA® 11.6.1, which requires NVIDIA Driver release 510 or later. However, if you are running on a Data Center GPU (for example, T4 or any other Tesla board), use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51 (or later R450), 460.27 (or later R460), or 470.57 (or later R470). The CUDA driver's compatibility package only supports specific drivers. For a complete list of supported drivers, see CUDA Application Compatibility. For more information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 22.03 supports CUDA compute capability 6.0 and later. This corresponds to GPUs in the NVIDIA Pascal, NVIDIA Volta™, NVIDIA Turing™, and NVIDIA Ampere Architecture GPU families. For a list of GPUs to which this compute capability corresponds, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Models can now load from a serialized `model_config` message with the Triton Server API.
- ONNX Runtime, TensorRT, and Tensorflow backends now support server-side, multi-dimensional ragged batching.
- Cache miss statistics have been added to the Prometheus metrics.
- Trace settings can be configured with the Triton Server Trace Protocol.
- Refer to the 22.03 column of the Frameworks Support Matrix for container image versions on which the 22.03 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and NVIDIA TensorRT™ are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.03	2.20.0	20.04	NVIDIA CUDA 11.6.1	TensorRT 8.2.3
22.02	2.19.0		NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Starting in 22.02, the Triton container, which uses the 22.02 PyTorch container, reports an error in the PyTorch backend when using scripted models that were exported in the legacy format (using our 19.09 or previous PyTorch NGC containers corresponding to PyTorch 1.2.0 or previous releases).

  To avoid this error and successfully load the model in Triton, you need to export the model again by using a later version of PyTorch.

- Triton pip wheels for ARM SBSA are not available from PyPI, so pip will install an incorrect Jetson version of Triton for ARM SBSA.

  The correct wheel file can be pulled directly from the ARM SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU.

  Refer to https://github.com/pytorch/pytorch/issues/66930 for more information.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices such as A100 and A30.
- Triton metrics might not work if the host machine is running a separate DCGM agent on bare-metal or in a container.
- Running a PyTorch TorchScript model by using the PyTorch backend, where multiple instances of a model are configured, can lead to a slowdown in model execution because of the following PyTorch issue:

  https://github.com/pytorch/pytorch/issues/27902

Chapter 42. Triton Inference Server Release 22.02

The Triton Inference Server container image, release 22.02, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the contatiner includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 22.02 is based on NVIDIA CUDA 11.6.0, which requires NVIDIA Driver release 510 or later. However, if you are running on a Data Center GPU (for example, T4 or any other Tesla board), you may use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51 (or later R450), 460.27 (or later R460), or 470.57 (or later R470). The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 22.02 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Enabled full-fledged SSL support in HTTP C++ client library.

- New cache metrics added to Prometheus metrics.
- PyTorch Backend now supports passing inputs in the form of a dictionary of tensors.
- Refer to the 22.02 column of the Frameworks Support Matrix for container image versions on which the 22.02 inference server container is based.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.02	2.19.0	20.04	NVIDIA CUDA 11.6.0	TensorRT 8.2.3
22.01	2.18.0			TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Starting in 22.02, the Triton container (which uses the 22.02 PyTorch container) will report an error in the PyTorch backend when using scripted models that were exported in the legacy format (using our 19.09 or previous PyTorch NGC containers corresponding to PyTorch 1.2.0 or previous releases). To avoid this error, you will need to re-export the model using a recent version of PyTorch to be able to load the model successfully in Triton.
- Addition of cache metrics may affect 3rd party tools/calculations for inference/compute latencies in models with caching enabled not accounting for cache hit requests that don't require inference.
- Triton pip wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton for ARM SBSA. The correct wheel file can be pulled directly from the ARM SBSA SDK image and manually installed.

- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. See https://github.com/pytorch/pytorch/issues/66930.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics may not work if the host machine is running a separate DCGM agent, either on bare-metal or in a container.
- Running a PyTorch TorchScript model using the PyTorch backend, where multiple instances of a model are configured can lead to a slowdown in model execution due to the following PyTorch issue: https://github.com/pytorch/pytorch/issues/27902.

Chapter 43. Triton Inference Server Release 22.01

The Triton Inference Server container image, release 22.01, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

For the list of what the contatiner includes, refer to Deep Learning Frameworks Support Matrix.

Driver Requirements

Release 22.01 is based on NVIDIA CUDA 11.6.0, which requires NVIDIA Driver release 510 or later. However, if you are running on a Data Center GPU (for example, T4 or any other Tesla board), you may use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51 (or later R450), 460.27 (or later R460), or 470.57 (or later R470). The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 22.01 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Implicit state management can be used for ONNX Runtime and TensorRT backends.

- State initialization from a constant is now supported in Implicit State management.
- PyTorch and TensorFlow models now support batching on Inferentia.
- PyTorch and Python backends are now supported on Jetson.
- ARM Support has been added for the Performance Analyzer and Model Analyzer.
- Refer to the 22.01 column of the Frameworks Support Matrix for container image versions that the 22.01 inference server container is based on.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
22.01	2.18.0	20.04	NVIDIA CUDA 11.6.0	TensorRT 8.2.2
21.12	2.17.0		NVIDIA CUDA 11.5.0	TensorRT 8.2.1.8
21.11	2.16.0			TensorRT 8.2.1.8 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Triton pip wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton for ARM SBSA. The correct wheel file can be pulled directly from the ARM SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. See https://github.com/pytorch/pytorch/issues/66930.
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics may not work if the host machine is running a separate DCGM agent, either on bare-metal or in a container.

- Running a PyTorch TorchScript model using the PyTorch backend, where multiple instances of a model are configured can lead to a slowdown in model execution due to the following PyTorch issue: https://github.com/pytorch/pytorch/issues/27902.

Chapter 44. Triton Inference Server Release 21.12

The Triton Inference Server container image, release 21.12, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.5.0
- cuBLAS 11.7.3.1
- NVIDIA cuDNN 8.3.1.22
- NVIDIA NCCL 2.11.4 (optimized for NVLink™)
- rdma-core 36.0
- OpenMPI 4.1.1+
- OpenUCX 1.11.0rc1+
- GDRCopy 2.3
- NVIDIA HPC-X 2.9
- TensorRT 8.2.1.8
- SHARP 2.5

Driver Requirements

Release 21.12 is based on NVIDIA CUDA 11.5.0, which requires NVIDIA Driver release 495 or later. However, if you are running on a Data Center GPU (for example, T4 or any other Tesla board), you may use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51 (or later R450), 460.27 (or later R460), or 470.57 (or later R470). The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility

topic. For more information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 21.12 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Improved Inferentia support to use Neuron Runtime 2.x and multiple instances.
- Models from MLflow can now be deployed to Triton with the MLflow plugin.
- The preview release of TorchTRT models is now supported. PyTorch models optimized using TensorRT can now be loaded into Triton in the same way as other PyTorch models.
- At the end of each Model Analyzer phase, an example command line will be printed to run the next phase.
- Refer to the 21.12 column of the Frameworks Support Matrix for container image versions that the 21.12 inference server container is based on.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

<table>
<tr><th>Container Version</th><th>Triton Inference Server</th><th>Ubuntu</th><th>CUDA Toolkit</th><th>TensorRT</th></tr>
<tr><td>21.12</td><td>2.17.0</td><td rowspan="5">20.04</td><td rowspan="2">NVIDIA CUDA 11.5.0</td><td>TensorRT 8.2.1.8</td></tr>
<tr><td>21.11</td><td>2.16.0</td><td rowspan="2">TensorRT 8.2.1.8 for x64 Linux
TensorRT 8.0.2.2 for ARM SBSA Linux</td></tr>
<tr><td>21.10</td><td>2.15.0</td><td>NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2</td></tr>
<tr><td>21.09</td><td>2.14.0</td><td>NVIDIA CUDA 11.4.2</td><td>TensorRT 8.0.3</td></tr>
<tr><td>21.08</td><td>2.13.0</td><td>NVIDIA CUDA 11.4.1</td><td>TensorRT 8.0.1.6</td></tr>
</table>

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- There was a bug in the GRPC protobuf implementation that was resolved by https://github.com/triton-inference-server/common/pull/34. If the client code uses the `byte_contents` field, the code must be updated to instead use 'bytes_contents'.
- Triton pip wheels for ARM SBSA are not available from PyPI and pip will install an incorrect Jetson version of Triton for ARM SBSA. The correct wheel file can be pulled directly from the ARM SBSA SDK image and manually installed.
- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. See https://github.com/pytorch/pytorch/issues/66930.
- Triton's TensorRT support depends on the input-consumed feature of TensorRT. In some rare cases using TensorRT 8.0 and earlier versions, the input-consumed event fires earlier than expected, causing Triton to overwrite input tensors while they are still in use and leading to corrupt input data being used for inference. This situation occurs when the inputs feed directly into a TensorRT layer that is optimized into a ForeignNode in the builder log. If you encounter accuracy issues with your TensorRT model, you can work around the issue by enabling the `output_copy_stream` option in your model's configuration (https://github.com/triton-inference-server/common/blob/main/protobuf/model_config.proto#L816).
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics may not work if the host machine is running a separate DCGM agent, either on bare-metal or in a container.
- Running a PyTorch TorchScript model using the PyTorch backend, where multiple instances of a model are configured can lead to a slowdown in model execution due to the following PyTorch issue: https://github.com/pytorch/pytorch/issues/27902.

Chapter 45. Triton Inference Server Release 21.11

The Triton Inference Server container image, release 21.11, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.5.0
- cuBLAS 11.7.3.1
- NVIDIA cuDNN 8.3.0.96
- NVIDIA NCCL 2.11.4 (optimized for NVLink™)
- rdma-core 36.0
- OpenMPI 4.1.1+
- OpenUCX 1.11.0rc1+
- GDRCopy 2.3
- NVIDIA HPC-X 2.9
- TensorRT 8.0.3.4 for x64 Linux
- TensorRT 8.0.2.2 for ARM SBSA Linux
- SHARP 2.5

Driver Requirements

Release 21.11 is based on NVIDIA CUDA 11.5.0, which requires NVIDIA Driver release 495 or later. However, if you are running on a Data Center GPU (for example, T4 or any other Tesla board), you may use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51 (or later R450), 460.27 (or later R460), or 470.57 (or later R470). The CUDA driver's compatibility package only supports particular drivers.

For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 21.11 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Added support for LightGBM models with categorical features in FIL backend.
- Added Jetson examples in documentation.
- Completed proof of concept of Inferentia support.
- Added ARM Support for Model Analyzer.
- Refer to the 21.11 column of the Frameworks Support Matrix for container image versions that the 21.11 inference server container is based on.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.11	2.16.0	20.04	NVIDIA CUDA 11.5.0	TensorRT 8.0.3.4 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.10	2.15.0		NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. See https://github.com/pytorch/pytorch/issues/66930.
- Triton's TensorRT support depends on the input-consumed feature of TensorRT. In some rare cases using TensorRT 8.0 and earlier versions, the input-consumed event fires earlier than expected, causing Triton to overwrite input tensors while they are still in use and leading to corrupt input data being used for inference. This situation occurs when the inputs feed directly into a TensorRT layer that is optimized into a ForeignNode in the builder log. If you encounter accuracy issues with your TensorRT model, you can work around the issue by enabling the `output_copy_stream` option in your model's configuration (https://github.com/triton-inference-server/common/blob/main/protobuf/model_config.proto#L816).
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics may not work if the host machine is running a separate DCGM agent, either on bare-metal or in a container.
- There is a known issue in TensorRT 8.0 regarding accuracy for a certain case of int8 inferencing on A40 and similar GPUs. The version of TF-TRT in TF2 21.09 includes a feature that works around this issue, but TF1 21.09 does not include that feature and therefore Triton users may experience the accuracy drop for a small subset of model/data type/batch size combinations on A40 when TF-TRT is used through the TF1 backend. This will be fixed in the next version of TensorRT.
- Running a PyTorch TorchScript model using the PyTorch backend, where multiple instances of a model are configured can lead to a slowdown in model execution due to the following PyTorch issue: https://github.com/pytorch/pytorch/issues/27902.

Chapter 46. Triton Inference Server Release 21.10

The Triton Inference Server container image, release 21.10, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2
- NVIDIA cuDNN 8.2.4.15
- NVIDIA NCCL 2.11.4 (optimized for NVLink™)
- rdma-core 36.0
- OpenMPI 4.1.1+
- OpenUCX 1.11.0rc1+
- GDRCopy 2.3
- NVIDIA HPC-X 2.9
- TensorRT 8.0.3.4 for x64 Linux
- TensorRT 8.0.2.2 for ARM SBSA Linux
- SHARP 2.5

Driver Requirements

Release 21.10 is based on NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2, which requires NVIDIA Driver release 470 or later. However, if you are running on Data Center GPUs (formerly Tesla), for example, T4, you may use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51 (or later R450), or 460.27 (or later R460). The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more

information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 21.10 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Rate limiter is now available and manages the rate at which requests are scheduled on model instances by Triton.
- Starting with the 21.10 release, a beta version of the Triton Inference Server container is available for the ARM SBSA platform.
- Windows Triton build now supports HTTP protocol.
- Triton added support for caching responses to inference requests.
- Sequence IDs can now accept strings.
- Container composer tool can generate CPU-only Triton containers.
- Refer to the 21.10 column of the Frameworks Support Matrix for container image versions that the 21.10 inference server container is based on.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.10	2.15.0	20.04	NVIDIA CUDA 11.4.2 with cuBLAS 11.6.5.2	TensorRT 8.0.3.4 for x64 Linux TensorRT 8.0.2.2 for ARM SBSA Linux
21.09	2.14.0		NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Traced models in PyTorch seem to create overflows when int8 tensor values are transformed to int32 on the GPU. See https://github.com/pytorch/pytorch/issues/66930.
- Triton's TensorRT support depends on the input-consumed feature of TensorRT. In some rare cases using TensorRT 8.0 and earlier versions, the input-consumed event fires earlier than expected, causing Triton to overwrite input tensors while they are still in use and leading to corrupt input data being used for inference. This situation occurs when the inputs feed directly into a TensorRT layer that is optimized into a ForeignNode in the builder log. If you encounter accuracy issues with your TensorRT model, you can work around the issue by enabling the `output_copy_stream` option in your model's configuration (https://github.com/triton-inference-server/common/blob/main/protobuf/model_config.proto#L816).
- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics may not work if the host machine is running a separate DCGM agent, either on bare-metal or in a container.
- There is a known issue in TensorRT 8.0 regarding accuracy for a certain case of int8 inferencing on A40 and similar GPUs. The version of TF-TRT in TF2 21.09 includes a feature that works around this issue, but TF1 21.09 does not include that feature and therefore Triton users may experience the accuracy drop for a small subset of model/data type/batch size combinations on A40 when TF-TRT is used through the TF1 backend. This will be fixed in the next version of TensorRT.
- Running a PyTorch TorchScript model using the PyTorch backend, where multiple instances of a model are configured can lead to a slowdown in model execution due to the following PyTorch issue: https://github.com/pytorch/pytorch/issues/27902.

Chapter 47. Triton Inference Server Release 21.09

The Triton Inference Server container image, release 21.09, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.4.2
- cuBLAS 11.6.1.51
- NVIDIA cuDNN 8.2.4.15
- NVIDIA NCCL 2.11.4 (optimized for NVLink™)
- rdma-core 36.0
- OpenMPI 4.1.1+
- OpenUCX 1.11.0rc1+
- GDRCopy 2.3
- NVIDIA HPC-X 2.9
- TensorRT 8.0.3

Driver Requirements

Release 21.09 is based on NVIDIA CUDA 11.4.2, which requires NVIDIA Driver release 470 or later. However, if you are running on Data Center GPUs (formerly Tesla), for example, T4, you may use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51 (or later R450), or 460.27 (or later R460). The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more

information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 21.09 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Full-featured, beta version of Business Logic Scripting released.
- Beta version for basic JAVA Client released. See https://github.com/triton-inference-server/client/tree/r21.09/src/java for a list of supported features.
- A stack trace is now printed when Triton crashes to aid in debugging.
- The Triton Client SDK wheel file is now available directly from PyPI for both Ubuntu and Windows.
- The TensorRT backend is now an optional part of Triton just like all the other backends. The compose utility can be used to create a Triton container that does not contain the TensorRT backend.
- Model Analyzer can profile with perf_analyzer's C-API.
- Model Analyzer can use the CUDA Device Index in addition to the GPU UUID in the `-gpus` flag.
- Refer to the 21.09 column of the Frameworks Support Matrix for container image versions that the 21.09 inference server container is based on.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.09	2.14.0	20.04	NVIDIA CUDA 11.4.2	TensorRT 8.0.3
21.08	2.13.0		NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Triton's TensorRT support depends on the input-consumed feature of TensorRT. In some rare cases using TensorRT 8.0 and earlier versions, the input-consumed

event fires earlier than expected, causing Triton to overwrite input tensors while they are still in use and leading to corrupt input data being used for inference. This situation occurs when the inputs feed directly into a TensorRT layer that is optimized into a ForeignNode in the builder log. If you encounter accuracy issues with your TensorRT model, you can work around the issue by enabling the `output_copy_stream` option in your model's configuration (https://github.com/triton-inference-server/common/blob/main/protobuf/model_config.proto#L816).

- Triton cannot retrieve GPU metrics with MIG-enabled GPU devices (A100 and A30).
- Triton metrics may not work if the host machine is running a separate DCGM agent, either on bare-metal or in a container.
- There is a known issue in TensorRT 8.0 regarding accuracy for a certain case of int8 inferencing on A40 and similar GPUs. The version of TF-TRT in TF2 21.09 includes a feature that works around this issue, but TF1 21.09 does not include that feature and therefore Triton users may experience the accuracy drop for a small subset of model/data type/batch size combinations on A40 when TF-TRT is used through the TF1 backend. This will be fixed in the next version of TensorRT.
- Running a PyTorch TorchScript model using the PyTorch backend, where multiple instances of a model are configured can lead to a slowdown in model execution due to the following PyTorch issue: https://github.com/pytorch/pytorch/issues/27902.

Chapter 48. Triton Inference Server Release 21.08

The Triton Inference Server container image, release 21.08, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.4.1
- cuBLAS 11.5.4
- NVIDIA cuDNN 8.2.2.6
- NVIDIA NCCL 2.10.3 (optimized for NVLink™)
- rdma-core 36.0
- OpenMPI 4.1.1+
- OpenUCX 1.11.0rc1+
- GDRCopy 2.2
- NVIDIA HPC-X 2.9
- TensorRT 8.0.1.6

Driver Requirements

Release 21.08 is based on NVIDIA CUDA 11.4.1, which requires NVIDIA Driver release 470 or later. However, if you are running on Data Center GPUs (formerly Tesla), for example, T4, you may use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51 (or later R450), or 460.27 (or later R460). The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more

information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 21.08 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Initial Beta release for Business Logic Scripting, a new set of utility functions that allow the execution of inference requests on other models being served by Triton as part of executing a Python model.
- Released new Container Composition Utility which can be used to create custom Triton containers with specific backends and repository agents.
- Starting in 21.08, Triton will release two new containers on NGC.
  - `nvcr.io/nvidia/tritonserver:21.08-tf-python-py3` - GPU enabled Triton server with only the TensorFlow 2.x and Python backends.
  - `nvcr.io/nvidia/tritonserver:21.08-pyt-python-py3` - GPU enabled Triton server with only the PyTorch and Python backends.
- Added Model Analyzer support for models with custom operations.
- Refer to the 21.08 column of the Frameworks Support Matrix for container image versions that the 21.08 inference server container is based on.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.08	2.13.0	20.04	NVIDIA CUDA 11.4.1	TensorRT 8.0.1.6
21.07	2.12.0		NVIDIA CUDA 11.4.0	
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Triton's TensorRT support depends on the input-consumed feature of TensorRT. In some rare cases using TensorRT 8.0 and earlier versions, the input-consumed event fires earlier than expected, causing Triton to overwrite input tensors while they are still in use and leading to corrupt input data being used for inference.

This situation occurs when the inputs feed directly into a TensorRT layer that is optimized into a ForeignNode in the builder log. If you encounter accuracy issues with your TensorRT model, you can work around the issue by enabling the `output_copy_stream` option in your model's configuration (https://github.com/triton-inference-server/common/blob/main/protobuf/model_config.proto#L816).

- Loading models in ONNX Runtime on the Windows build of Triton may be slow due to the JIT compiler being invoked for newer CUDA architectures. For more information, refer to https://github.com/triton-inference-server/onnxruntime_backend/issues/58/.
- There is a known issue in TensorRT 8.0 regarding accuracy for a certain case of int8 inferencing on A40 and similar GPUs. The version of TF-TRT in TF2 21.08 includes a feature that works around this issue, but TF1 21.08 does not include that feature and therefore Triton users may experience the accuracy drop for a small subset of model/data type/batch size combinations on A40 when TF-TRT is used through the TF1 backend. This will be fixed in the next version of TensorRT.
- Running a PyTorch TorchScript model using the PyTorch backend, where multiple instances of a model are configured can lead to a slowdown in model execution due to the following PyTorch issue: https://github.com/pytorch/pytorch/issues/27902.
- Compared with the 21.02 and earlier releases, there are backwards incompatible changes in the example Python client shared-memory support library when that library is used for tensors of type BYTES. The `utils.serialize_byte_tensor()` and `utils.deserialize_byte_tensor()` functions now return np.object_ numpy arrays where previously they returned `np.bytes_ numpy` arrays. Code depending on `np.bytes_` must be updated. This change was necessary because the np.bytes_ type removes all trailing zeros from each array element and so binary sequences ending in zero(s) could not be represented with the old behavior. Correct usage of the Python client shared-memory support library is shown inhttps://github.com/triton-inference-server/server/blob/r21.03/src/clients/python/examples/simple_http_shm_string_client.py.

Chapter 49. Triton Inference Server Release 21.07

The Triton Inference Server container image, release 21.07, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.4.0
- cuBLAS 11.5.2.43
- NVIDIA cuDNN 8.2.2.6
- NVIDIA NCCL 2.10.3 (optimized for NVLink™)
- rdma-core 32.1
- OpenMPI 4.1.1rc1
- OpenUCX 1.10.1
- GDRCopy 2.2
- NVIDIA HPC-X 2.8.2rc3
- TensorRT 8.0.1.6

Driver Requirements

Release 21.07 is based on NVIDIA CUDA 11.4.0, which requires NVIDIA Driver release 470 or later. However, if you are running on Data Center GPUs (formerly Tesla), for example, T4, you may use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51 (or later R450), or 460.27 (or later R460). The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more

information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 21.07 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Added support for CPU in RAPIDS FIL Backend.
- Inference requests using the C API are now allowed to provide multiple copies of an input tensor in different memories. Triton will choose the most performant copy to use depending on where the inference request is executed.
- For ONNX models using TensorRT acceleration, the `tensorrt_accelerator` option in the model configuration can now specify precision and workspace size.https://github.com/triton-inference-server/server/blob/main/docs/user_guide/optimization.md#framework-specific-optimization.
- Model Analyzer added an offline mode, which prioritizes throughput over latency for offline inferencing scenarios. A new set of reports and graphs are created to better analyze the offline use case.
- Refer to the 21.07 column of the Frameworks Support Matrix for container image versions that the 21.07 inference server container is based on.
- Ubuntu 20.04 with June 2021 updates.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.07	2.12.0	20.04	NVIDIA CUDA 11.4.0	TensorRT 8.0.1.6
21.06.1	2.11.0		NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Triton's TensorRT support depends on the input-consumed feature of TensorRT. In some rare cases using TensorRT 8.0 and earlier versions, the input-consumed event fires earlier than expected, causing Triton to overwrite input tensors while they are still in use and leading to corrupt input data being used for inference. This situation occurs when the inputs feed directly into a TensorRT layer that is optimized into a ForeignNode in the builder log. If you encounter accuracy

issues with your TensorRT model, you can work around the issue by enabling the `output_copy_stream` option in your model's configuration (https://github.com/triton-inference-server/common/blob/main/protobuf/model_config.proto#L816).

- The 21.07 release includes libsystemd and libudev versions that have a known vulnerability that was discovered late in our QA process. See CVE-2021-33910 for details. This will be fixed in the next release.
- ONNX Runtime TRT support was removed due to incompatibility with TensorRT 8.0.
- There is a known issue in TensorRT 8.0 regarding accuracy for a certain case of int8 inferencing on A40 and similar GPUs. The version of TF-TRT in TF2 21.07 includes a feature that works around this issue, but TF1 21.07 does not include that feature and therefore Triton users may experience the accuracy drop for a small subset of model/data type/batch size combinations on A40 when TF-TRT is used through the TF1 backend. This will be fixed in the next version of TensorRT.
- Running a PyTorch TorchScript model using the PyTorch backend, where multiple instances of a model are configured can lead to a slowdown in model execution due to the following PyTorch issue: https://github.com/pytorch/pytorch/issues/27902.
- Compared with the 21.02 and earlier releases, there are backwards incompatible changes in the example Python client shared-memory support library when that library is used for tensors of type BYTES. The `utils.serialize_byte_tensor()` and `utils.deserialize_byte_tensor()` functions now return np.object_ numpy arrays where previously they returned `np.bytes_ numpy` arrays. Code depending on `np.bytes_` must be updated. This change was necessary because the np.bytes_ type removes all trailing zeros from each array element and so binary sequences ending in zero(s) could not be represented with the old behavior. Correct usage of the Python client shared-memory support library is shown inhttps://github.com/triton-inference-server/server/blob/r21.03/src/clients/python/examples/simple_http_shm_string_client.py.

Chapter 50. Triton Inference Server Release 21.06.1

The Triton Inference Server container image, release 21.06.1, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.3.1
- cuBLAS 11.5.1.109
- NVIDIA cuDNN 8.2.1
- NVIDIA NCCL 2.9.9 (optimized for NVLink™)
- rdma-core 32.1
- OpenMPI 4.1.1rc1
- OpenUCX 1.10.1
- GDRCopy 2.2
- NVIDIA HPC-X 2.8.2rc3
- TensorRT 7.2.3.4

Driver Requirements

Release 21.06.1 is based on NVIDIA CUDA 11.3.1, which requires NVIDIA Driver release 465.19.01 or later. However, if you are running on Data Center GPUs (formerly Tesla), for example, T4, you may use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51 (or later R450), or 460.27 (or later R460). The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more

information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 21.06.1 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The Forest Inference Library (FIL) backend is added to Triton. The FIL backend allows forest models trained by several popular machine learning frameworks (including XGBoost, LightGBM, Scikit-Learn, and cuML) to be deployed in a Triton.
- Windows version of Triton now includes the OpenVino backend.
- The Performance Analyzer (perf_analyzer) now supports testing against the Triton C API.
- The Python backend now allows the use of conda to create a unique execution environment for your Python model. See https://github.com/triton-inference-server/python_backend#using-custom-python-execution-environments.
- Python models that crash or exit unexpectedly are now automatically restarted by Triton.
- Model repositories in S3 storage can now be accessed using HTTPS protocol. See https://github.com/triton-inference-server/server/blob/main/docs/user_guide/model_repository.md for more information.
- Triton now collects GPU metrics for MIG partitions.
- Passive model instances can now be specified in the model configuration. A passive model instance will be loaded and initialized by Triton, but no inference requests will be sent to the instance. Passive instances are typically used by a custom backend that uses its own mechanisms to distribute work to the passive instances. See the ModelInstanceGroup section of model_config.proto for the setting.
- NVDLA support is added to the TensorRT backend.
- ONNX Runtime version updated to 1.8.0.
- Windows build documentation simplified and improved.
- Improved detailed and summary reports in Model Analyzer.
- Added an offline mode to Model Analyzer.
- Refer to the 21.06 column of the Frameworks Support Matrix for container image versions that the 21.05 inference server container is based on.

- Ubuntu 20.04 with May 2021 updates.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.06.1	2.11.0	20.04	NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.06				
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Triton's TensorRT support depends on the input-consumed feature of TensorRT. In some rare cases using TensorRT 8.0 and earlier versions, the input-consumed event fires earlier than expected, causing Triton to overwrite input tensors while they are still in use and leading to corrupt input data being used for inference. This situation occurs when the inputs feed directly into a TensorRT layer that is optimized into a ForeignNode in the builder log. If you encounter accuracy issues with your TensorRT model, you can work around the issue by enabling the `output_copy_stream` option in your model's configuration (https://github.com/triton-inference-server/common/blob/main/protobuf/model_config.proto#L816).
- Running a PyTorch TorchScript model using the PyTorch backend, where multiple instances of a model are configured can lead to a slowdown in model execution due to the following PyTorch issue: https://github.com/pytorch/pytorch/issues/27902.
- Compared with the 21.02 and earlier releases, there are backwards incompatible changes in the example Python client shared-memory support library when that library is used for tensors of type BYTES. The `utils.serialize_byte_tensor()` and `utils.deserialize_byte_tensor()` functions now return np.object_ numpy arrays where previously they returned `np.bytes_ numpy` arrays. Code depending on `np.bytes_` must be updated. This change was necessary because the np.bytes_ type removes all trailing zeros from each array element and so binary sequences ending in zero(s) could not be represented with the old behavior. Correct usage of the Python client shared-memory support library is shown inhttps://github.com/triton-inference-server/server/blob/r21.03/src/clients/python/examples/simple_http_shm_string_client.py.

Chapter 51. Triton Inference Server Release 21.06

The Triton Inference Server container image, release 21.06, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.3.1
- cuBLAS 11.5.1.109
- NVIDIA cuDNN 8.2.1
- NVIDIA NCCL 2.9.9 (optimized for NVLink™)
- rdma-core 32.1
- OpenMPI 4.1.1rc1
- OpenUCX 1.10.1
- GDRCopy 2.2
- NVIDIA HPC-X 2.8.2rc3
- TensorRT 7.2.3.4

Driver Requirements

Release 21.06 is based on NVIDIA CUDA 11.3.1, which requires NVIDIA Driver release 465.19.01 or later. However, if you are running on Data Center GPUs (formerly Tesla), for example, T4, you may use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51 (or later R450), or 460.27 (or later R460). The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more

information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 21.06 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Windows version of Triton now includes the OpenVino backend.
- The Performance Analyzer (perf_analyzer) now supports testing against the Triton C API.
- The Python backend now allows the use of conda to create a unique execution environment for your Python model. See https://github.com/triton-inference-server/python_backend#using-custom-python-execution-environments.
- Python models that crash or exit unexpectedly are now automatically restarted by Triton.
- Model repositories in S3 storage can now be accessed using HTTPS protocol. See https://github.com/triton-inference-server/server/blob/main/docs/user_guide/model_repository.md for more information.
- Triton now collects GPU metrics for MIG partitions.
- Passive model instances can now be specified in the model configuration. A passive model instance will be loaded and initialized by Triton, but no inference requests will be sent to the instance. Passive instances are typically used by a custom backend that uses its own mechanisms to distribute work to the passive instances. See the ModelInstanceGroup section of model_config.proto for the setting.
- NVDLA support is added to the TensorRT backend.
- ONNX Runtime version updated to 1.8.0.
- Windows build documentation simplified and improved.
- Improved detailed and summary reports in Model Analyzer.
- Added an offline mode to Model Analyzer.
- Refer to the 21.06 column of the Frameworks Support Matrix for container image versions that the 21.05 inference server container is based on.
- Ubuntu 20.04 with May 2021 updates.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.06	2.11.0	20.04	NVIDIA CUDA 11.3.1	TensorRT 7.2.3.4
21.05	2.10.0		NVIDIA CUDA 11.3.0	
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Triton's TensorRT support depends on the input-consumed feature of TensorRT. In some rare cases using TensorRT 8.0 and earlier versions, the input-consumed event fires earlier than expected, causing Triton to overwrite input tensors while they are still in use and leading to corrupt input data being used for inference. This situation occurs when the inputs feed directly into a TensorRT layer that is optimized into a ForeignNode in the builder log. If you encounter accuracy issues with your TensorRT model, you can work around the issue by enabling the `output_copy_stream` option in your model's configuration (https://github.com/triton-inference-server/common/blob/main/protobuf/model_config.proto#L816).
- The 21.06 release of Triton was built against the wrong commit of the FIL backend code, causing an incompatible version of RAPIDS to be used instead of the intended RAPIDS 21.06 stable release. This issue is fixed in the new 21.06.1 container released on NGC. Although the Triton server itself and other integrated backends will work, the FIL backend will not work in the 21.06 Triton container
- Running a PyTorch TorchScript model using the PyTorch backend, where multiple instances of a model are configured can lead to a slowdown in model execution due to the following PyTorch issue: https://github.com/pytorch/pytorch/issues/27902.
- Compared with the 21.02 and earlier releases, there are backwards incompatible changes in the example Python client shared-memory support library when that library is used for tensors of type BYTES. The `utils.serialize_byte_tensor()` and `utils.deserialize_byte_tensor()` functions now return np.object_ numpy arrays where previously they returned `np.bytes_ numpy` arrays. Code depending on `np.bytes_` must be updated. This change was necessary because the np.bytes_ type removes all trailing zeros from each array element and so binary sequences ending in zero(s) could not be represented with the old behavior. Correct usage of the Python client shared-memory support library is shown inhttps://github.com/triton-inference-server/server/blob/r21.03/src/clients/python/examples/simple_http_shm_string_client.py.

Chapter 52. Triton Inference Server Release 21.05

The Triton Inference Server container image, release 21.05, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.3.0
- cuBLAS 11.5.1.101
- NVIDIA cuDNN 8.2.0.51
- NVIDIA NCCL 2.9.8 (optimized for NVLink™)
- rdma-core 32.1
- OpenMPI 4.1.1rc1
- OpenUCX 1.10.0
- GDRCopy 2.2
- NVIDIA HPC-X 2.8.2rc3
- TensorRT 7.2.3.4

Driver Requirements

Release 21.05 is based on NVIDIA CUDA 11.3.0, which requires NVIDIA Driver release 465.19.01 or later. However, if you are running on Data Center GPUs (formerly Tesla), for example, T4, you may use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51 (or later R450), or 460.27 (or later R460). The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more

information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 21.05 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Triton on Jetson now supports ONNX via the ONNX Runtime backend.
- The Triton server and HTTP clients (Python and C++) now support compression.
- Ragged batching is now supported for ONNX models.
- The Triton clients have moved to a separate repo: https://github.com/triton-inference-server/client
- Trace now correctly reports all timestamps for all backends.
- NVTX annotations are fixed.
- The legacy custom backend support is removed. All custom backends must be implemented using the TRITONBACKEND API described here: https://github.com/triton-inference-server/backend.
- Added CLI subcommands in Model Analyzer for `profile`, `analyze`, and `report`. See CLI documentation for usage instructions.
  - This is a breaking change and requires updating Model Analyzer config files and CLI flags. See Configuring Model Analyzer and Quick Start for more information.
- Model Analyzer can create a detailed report of any specific model configuration with the `report` subcommand.
- CPU only mode is supported in Model Analyzer.
- Refer to the 21.05 column of the Frameworks Support Matrix for container image versions that the 21.05 inference server container is based on.
- Ubuntu 20.04 with April 2021 updates.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.05	2.10.0	20.04	NVIDIA CUDA 11.3.0	TensorRT 7.2.3.4
21.04	2.9.0			
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Triton's TensorRT support depends on the input-consumed feature of TensorRT. In some rare cases using TensorRT 8.0 and earlier versions, the input-consumed event fires earlier than expected, causing Triton to overwrite input tensors while they are still in use and leading to corrupt input data being used for inference.

This situation occurs when the inputs feed directly into a TensorRT layer that is optimized into a ForeignNode in the builder log. If you encounter accuracy issues with your TensorRT model, you can work around the issue by enabling the `output_copy_stream` option in your model's configuration (https://github.com/triton-inference-server/common/blob/main/protobuf/model_config.proto#L816).

- Running a PyTorch TorchScript model using the PyTorch backend, where multiple instances of a model are configured can lead to a slowdown in model execution due to the following PyTorch issue: https://github.com/pytorch/pytorch/issues/27902.
- Compared with the 21.02 and earlier releases, there are backwards incompatible changes in the example Python client shared-memory support library when that library is used for tensors of type BYTES. The `utils.serialize_byte_tensor()` and `utils.deserialize_byte_tensor()` functions now return np.object_ numpy arrays where previously they returned `np.bytes_ numpy` arrays. Code depending on `np.bytes_` must be updated. This change was necessary because the np.bytes_ type removes all trailing zeros from each array element and so binary sequences ending in zero(s) could not be represented with the old behavior. Correct usage of the Python client shared-memory support library is shown inhttps://github.com/triton-inference-server/server/blob/r21.03/src/clients/python/examples/simple_http_shm_string_client.py.

Chapter 53. Triton Inference Server Release 21.04

The Triton Inference Server container image, release 21.04, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.3.0
- cuBLAS 11.5.1.101
- NVIDIA cuDNN 8.2.0.41
- NVIDIA NCCL 2.9.6 (optimized for NVLink™)
- rdma-core 32.1
- OpenMPI 4.1.1rc1
- OpenUCX 1.10.0
- GDRCopy 2.2
- NVIDIA HPC-X 2.8.2rc3
- TensorRT 7.2.3.4

Driver Requirements

Release 21.04 is based on NVIDIA CUDA 11.3.0, which requires NVIDIA Driver release 465.19.01 or later. However, if you are running on Data Center GPUs (formerly Tesla), for example, T4, you may use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51 (or later R450), or 460.27 (or later R460). The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more

information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 21.04 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Python backend performance has been increased significantly.
- ONNX Runtime update to version 1.7.1.
- Triton Server is now available as a GKE Marketplace Application, see https://github.com/triton-inference-server/server/tree/master/deploy/gke-marketplace-app.
- The GRPC client libraries now allow compression to be enabled.
- Ragged batching is now supported for TensorFlow models.
- For TensorFlow models represented with SavedModel format, it is now possible to choose which graph and signature_def to load. See https://github.com/triton-inference-server/tensorflow_backend/tree/r21.04#parameters.
- A Helm Chart example is added for AWS. See https://github.com/triton-inference-server/server/tree/master/deploy/aws.
- The Model Control API is enhanced to provide an option when unloading an ensemble model. The option allows all contained models to be unloaded as part of unloading the ensemble. See https://github.com/triton-inference-server/server/blob/master/docs/protocol/extension_model_repository.md#model-repository-extension.
- Model reloading using the Model Control API previously resulted in the model being unavailable for a short period of time. This is now fixed so that the model remains available during reloading.
- Latency statistics and metrics for TensorRT models are fixed. Previously the sum of the "compute input", "compute infer" and "compute output" times accurately indicated the entire compute time but the total time could be incorrectly attributed across the three components. This incorrect attribution is now fixed and all values are now accurate.
- Error reporting is improved for the Azure, S3 and GCS cloud file system support.
- Fixed trace support for ensembles. The models contained within an ensemble are now traced correctly.
- Model Analyzer improvements:

- Summary report now includes GPU Power usage.
- Model Analyzer will find the Top N model configuration across multiple models.

- Refer to the 21.04 column of the Frameworks Support Matrix for container image versions that the 21.04 inference server container is based on.
- Ubuntu 20.04 with March 2021 updates.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.04	2.9.0	20.04	NVIDIA CUDA 11.3.0	TensorRT 7.2.3.4
21.03	2.8.0		NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Triton's TensorRT support depends on the input-consumed feature of TensorRT. In some rare cases using TensorRT 8.0 and earlier versions, the input-consumed event fires earlier than expected, causing Triton to overwrite input tensors while they are still in use and leading to corrupt input data being used for inference. This situation occurs when the inputs feed directly into a TensorRT layer that is optimized into a ForeignNode in the builder log. If you encounter accuracy issues with your TensorRT model, you can work around the issue by enabling the `output_copy_stream` option in your model's configuration (https://github.com/triton-inference-server/common/blob/main/protobuf/model_config.proto#L816).
- Compared with the 21.02 and earlier releases, there are backwards incompatible changes in the example Python client shared-memory support library when that library is used for tensors of type BYTES. The `utils.serialize_byte_tensor()` and `utils.deserialize_byte_tensor()` functions now return np.object_ numpy arrays where previously they returned `np.bytes_ numpy` arrays. Code depending on `np.bytes_` must be updated. This change was necessary because the np.bytes_ type removes all trailing zeros from each array element and so binary sequences ending in zero(s) could not be represented with the old behavior. Correct usage of the Python client shared-memory support library is shown inhttps://github.com/triton-inference-server/server/blob/r21.03/src/clients/python/examples/simple_http_shm_string_client.py.

Chapter 54. Triton Inference Server Release 21.03

The Triton Inference Server container image, release 21.03, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.2.1 including cuBLAS 11.4.1.1026
- NVIDIA cuDNN 8.1.1
- NVIDIA NCCL 2.8.4 (optimized for NVLink™)
- MLNX_OFED 5.1
- OpenMPI 4.0.5
- TensorRT 7.2.2.3

Driver Requirements

Release 21.03 is based on NVIDIA CUDA 11.2.1, which requires NVIDIA Driver release 460.32.03 or later. However, if you are running on Data Center GPUs (formerly Tesla), for example, T4, you may use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51(or later R450). The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 21.03 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families.

Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Repository agent is a new extensibility C API added to Triton that allows implementation of custom authentication, decryption, conversion, or similar operations when a model is loaded. See https://github.com/triton-inference-server/server/blob/master/docs/repository_agents.md.
- An OpenVINO backend is added to Triton to enable the execution of OpenVINO models on CPUs. See https://github.com/triton-inference-server/openvino_backend.
- The PyTorch backend is now maintained in its own repository: https://github.com/triton-inference-server/pytorch_backend
- The ONNX Runtime backend is now maintained in its own repository: https://github.com/triton-inference-server/onnxruntime_backend
- The Jetson release of Triton now supports the shared-memory protocol between clients and the Triton server.
- SSL/TLS Mutual Authentication support is added to the GRPC client library.
- A new Model Configuration option, "gather_kernel_buffer_threshold", can be specified to instruct Triton to use a CUDA kernel to gather inputs buffers onto the GPU. Using this option can improve inference performance for some models.
- The Python client libraries have been improved to more efficiently create numpy arrays for input and output tensors.
- The client libraries examples have been improved to more clearly describe how string and byte-blob tensors are supported by the Python Client API. See https://github.com/triton-inference-server/server/blob/master/docs/client_examples.md.
- Ubuntu 20.04 with February 2021 updates.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.03	2.8.0	20.04	NVIDIA CUDA 11.2.1	TensorRT 7.2.2.3

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.02	2.7.0		NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Triton's TensorRT support depends on the input-consumed feature of TensorRT. In some rare cases using TensorRT 8.0 and earlier versions, the input-consumed event fires earlier than expected, causing Triton to overwrite input tensors while they are still in use and leading to corrupt input data being used for inference. This situation occurs when the inputs feed directly into a TensorRT layer that is optimized into a ForeignNode in the builder log. If you encounter accuracy issues with your TensorRT model, you can work around the issue by enabling the `output_copy_stream` option in your model's configuration (https://github.com/triton-inference-server/common/blob/main/protobuf/model_config.proto#L816).

- There are backwards incompatible changes in the example Python client shared-memory support library when that library is used for tensors of type BYTES. The `utils.serialize_byte_tensor()` and `utils.deserialize_byte_tensor()` functions now return np.object_ numpy arrays where previously they returned `np.bytes_ numpy` arrays. Code depending on `np.bytes_` must be updated. This change was necessary because the np.bytes_ type removes all trailing zeros from each array element and so binary sequences ending in zero(s) could not be represented with the old behavior. Correct usage of the Python client shared-memory support library is shown inhttps://github.com/triton-inference-server/server/blob/r21.03/src/clients/python/examples/simple_http_shm_string_client.py.
- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.

Chapter 55. Triton Inference Server Release 21.02

The Triton Inference Server container image, release 21.02, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.2.0 including cuBLAS 11.3.1
- NVIDIA cuDNN 8.0.5
- NVIDIA NCCL 2.8.4 (optimized for NVLink™)
- MLNX_OFED 5.1
- OpenMPI 4.0.5
- TensorRT 7.2.2

Driver Requirements

Release 21.02 is based on NVIDIA CUDA 11.2.0, which requires NVIDIA Driver release 460.27.04 or later. However, if you are running on Data Center GPUs (formerly Tesla), for example, T4, you may use NVIDIA driver release 418.40 (or later R418), 440.33 (or later R440), 450.51(or later R450). The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades and NVIDIA CUDA and Drivers Support.

GPU Requirements

Release 21.02 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families.

Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Refer to the 21.02 column of the Frameworks Support Matrix for container image versions that the 21.02 inference server container is based on.
- Fixed a bug in TensorRT backend that could, in rare cases, lead to corruption of output tensors.
- Fixed a performance issue in the HTTP/REST client that occurred when the client does not explicitly request specific outputs. In this case all outputs are now returned as binary data where previously they were returned as JSON.
- Added an example Java and Scala client based on GRPC-generated API.
- Extended perf_analyzer to be able to work with TFServing and TorchServe.
- The legacy custom backend API is deprecated and will be removed in a future release. The Triton Backend API should be used as the API for custom backends. The Triton Backend API remains fully supported and that support will continue indefinitely.
- Model Analyzer parameters and test model configurations can be specified with JSON configuration file.
- Model Analyzer will report performance metrics for end-to-end latency and CPU memory usage.
- Ubuntu 20.04 with January 2021 updates.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
21.02	2.7.0	20.04	NVIDIA CUDA 11.2.0	TensorRT 7.2.2.3+cuda11.1.0.024
20.12	2.6.0		NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.
- Observed memory leak in gRPC client library. Suggested work around process: Restart service to free memory or run within Kubernetes with failover mechanism. For more details on the issue in gRPC, please reference: https://github.com/triton-inference-server/server/issues/2517. The memory leak is fixed on master branch by https://github.com/triton-inference-server/server/pull/2533 and the fix will be included in the 21.03 release. If required, the change can be applied to the 21.02 branch and the client library can be rebuilt: https://github.com/triton-inference-server/server/blob/master/docs/client_libraries.md.

Chapter 56. Triton Inference Server Release 21.01

The NVIDIA container image release for Triton Inference Server 21.01 has been canceled. The next release will be the 21.02 release which is expected to be released at the end of February.

Chapter 57. Triton Inference Server Release 20.12

The Triton Inference Server container image, release 20.12, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 20.04 including Python 3.8
- NVIDIA CUDA 11.1.1 including cuBLAS 11.3.0
- NVIDIA cuDNN 8.0.5
- NVIDIA NCCL 2.8.3 (optimized for NVLink™)
- MLNX_OFED 5.1
- OpenMPI 4.0.5
- TensorRT 7.2.2

Driver Requirements

Release 20.12 is based on NVIDIA CUDA 11.1.1, which requires NVIDIA Driver release 455 or later. However, if you are running on Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 418.xx, 440.30, or 450.xx. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 20.12 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Refer to the 20.12 column of the Frameworks Support Matrix for container image versions that the 20.12 inference server container is based on.
- Due to interactions with Ubuntu 20.04, the ONNX Runtime's OpenVINO execution provider is disabled in this release. OpenVINO support will be re-enabled in a subsequent release.
- The Triton *-py3-clientsdk container has been renamed to *-py3-sdk and now contains the Model Analyzer as well as the client libraries and examples.
- The PyTorch backend has been moved to a separate repository: https://github.com/triton-inference-server/pytorch_backend. As a result, it is now easy to add or remove it from Triton without requiring a rebuild: see https://github.com/triton-inference-server/server/blob/master/docs/compose.md.
- Initial release of the Model Analyzer tool in the Triton SDK container and PIP package in the NVIDIA Py Index.
- Ubuntu 20.04 with November 2020 updates.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.12	2.6.0	20.04	NVIDIA CUDA 11.1.1	TensorRT 7.2.2
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.

Chapter 58. Triton Inference Server Release 20.11

The Triton Inference Server container image, release 20.11, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 18.04 including Python 3.6
- NVIDIA CUDA 11.1.0 including cuBLAS 11.2.1
- NVIDIA cuDNN 8.0.4
- NVIDIA NCCL 2.8.2 (optimized for NVLink™)
- MLNX_OFED 5.1
- OpenMPI 4.0.5
- TensorRT 7.2.1

Driver Requirements

Release 20.11 is based on NVIDIA CUDA 11.1.0, which requires NVIDIA Driver release 455 or later. However, if you are running on Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 418.xx, 440.30, or 450.xx. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 20.11 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- ONNX Runtime backend updated to use ONNX Runtime 1.5.3.
- The PyTorch backend is moved to a dedicated repo: triton-inference-server/pytorch_backend.
- The Caffe2 backend is removed. Caffe2 models are no longer supported.
- Fixed handling of failed model reloads. If a model reload fails, the currently loaded version of the model will remain loaded and its availability will be uninterrupted.
- Releasing Triton ModelAnalyzer in the Triton SDK container and as a PIP package available in NVIDIA PyIndex.
- Refer to the 20.11 column of the Frameworks Support Matrix for container image versions that the 20.11 inference server container is based on.
- Ubuntu 18.04 with October 2020 updates.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.11	2.5.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.10	2.4.0			
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.

Chapter 59. Triton Inference Server Release 20.10

The Triton Inference Server container image, release 20.10, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 18.04 including Python 3.6
- NVIDIA CUDA 11.1.0 including cuBLAS 11.2.1
- NVIDIA cuDNN 8.0.4
- NVIDIA NCCL 2.7.8 (optimized for NVLink™)
- MLNX_OFED
- OpenMPI 3.1.6
- TensorRT 7.2.1

Driver Requirements

Release 20.10 is based on NVIDIA CUDA 11.1.0, which requires NVIDIA Driver release 455 or later. However, if you are running on Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 418.xx, 440.30, or 450.xx. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 20.10 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- A new Python backend allows Python code to run as a model within Triton. See https://github.com/triton-inference-server/python_backend.
- A new DALI backend allows running pre-processing and augmentation pipelines within Triton. See https://github.com/triton-inference-server/dali_backend.
- The perf_client application is renamed to perf_analyzer; functionality remains the same.
- A new Model Analyzer project is started with the goal of providing analysis and guidance on how to best optimize single or multiple models within Triton. The initial release analyzes GPU memory usage. See https://github.com/triton-inference-server/model_analyzer.
- Triton documentation now resides on GitHub and is reachable from https://github.com/triton-inference-server/server/blob/master/README.md.
- Build process for Triton has changed, see https://github.com/triton-inference-server/server/blob/master/docs/build.md.
- Triton backends are moving to separate repositories. In this release the TensorFlow, ONNX Runtime, Python and DALI backends are moved; see https://github.com/triton-inference-server/backend#where-can-i-find-all-the-backends-that-are-available-for-triton.
- Refer to the 20.10 column of the Frameworks Support Matrix for container image versions that the 20.10 inference server container is based on.
- Ubuntu 18.04 with September 2020 updates.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.10	2.4.0	18.04	NVIDIA CUDA 11.1.0	TensorRT 7.2.1
20.09	2.3.0		NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- TensorRT reformat-free I/O is not supported.
- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.

Chapter 60. Triton Inference Server Release 20.09

The Triton Inference Server container image, release 20.09, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 18.04 including Python 3.6
- NVIDIA CUDA 11.0.3 including cuBLAS 11.2.0
- NVIDIA cuDNN 8.0.4
- NVIDIA NCCL 2.7.8 (optimized for NVLink™)
- MLNX_OFED
- OpenMPI 3.1.6
- TensorRT 7.1.3

Driver Requirements

Release 20.09 is based on NVIDIA CUDA 11.0.3, which requires NVIDIA Driver release 450 or later. However, if you are running on Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 418.xx or 440.30. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 20.09 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Python Client library is now a pip package available from the NVIDIA pypi index. See https://github.com/triton-inference-server/server/blob/master/src/clients/python/library/README.md for more information.
- Fixed a performance issue with the HTTP/REST protocol and the Python client library that caused reduced performance when outputs were not requested explicitly in an inference request.
- Fixed some issues in reporting of statistics for ensemble models.
- GRPC updated to version 1.25.0.
- Refer to the 20.09 column of the Frameworks Support Matrix for container image versions that the 20.09 inference server container is based on.
- Ubuntu 18.04 with August 2020 updates.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.09	2.3.0	18.04	NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.08	2.2.0			
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- The KFServing HTTP/REST and GRPC protocols and corresponding Python and C ++ clients are beta quality and are likely to change.
- The new C API specified in `tritonserver.h` is beta quality and is likely to change.
- TensorRT reformat-free I/O is not supported.
- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.

Chapter 61. Triton Inference Server Release 20.08

The Triton Inference Server container image, release 20.08, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 18.04 including Python 3.6
- NVIDIA CUDA 11.0.3 including cuBLAS 11.2.0
- NVIDIA cuDNN 8.0.2
- NVIDIA NCCL 2.7.8 (optimized for NVLink™)
- MLNX_OFED
- OpenMPI 3.1.6
- TensorRT 7.1.3

Driver Requirements

Release 20.08 is based on NVIDIA CUDA 11.0.3, which requires NVIDIA Driver release 450 or later. However, if you are running on Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 418.xx or 440.30. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 20.08 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- TensorFlow 2.x is now supported in addition to TensorFlow 1.x. See the Frameworks Support Matrix for the supported TensorFlow versions. The version of TensorFlow used can be selected when launching Triton with the `--backend-config=tensorflow,version=<version>` flag. Set <version> to 1 or 2 to select TensorFlow1 or TensorFlow2 respectively. By default, TensorFlow1 is used.
- Added inference request timeout option to Python and C++ client libraries.
- Updated GRPC inference protocolti to fix performance regression.
- Explicit major/minor versioning added to TRITONSERVER and TRITONBACKED APIs.
- New CMake option TRITON_CLIENT_SKIP_EXAMPLES to disable building the client examples.
- Refer to the 20.08 column of the Frameworks Support Matrix for container image versions that the 20.08 inference server container is based on.
- Ubuntu 18.04 with July 2020 updates.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.08	2.2.0	18.04	NVIDIA CUDA 11.0.3	TensorRT 7.1.3
20.07	1.15.0 2.1.0		NVIDIA CUDA 11.0.194	
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- The KFServing HTTP/REST and GRPC protocols and corresponding Python and C ++ clients are beta quality and are likely to change.
- The new C API specified in `tritonserver.h` is beta quality and is likely to change.
- TensorRT reformat-free I/O is not supported.
- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.

Chapter 62. Triton Inference Server Release 20.07

The Triton Inference Server container image, release 20.07, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 18.04 including Python 3.6
- NVIDIA CUDA 11.0.194 including cuBLAS 11.1.0
- NVIDIA cuDNN 8.0.1
- NVIDIA NCCL 2.7.6 (optimized for NVLink™)
- MLNX_OFED
- OpenMPI 3.1.6
- TensorRT 7.1.3

Driver Requirements

Release 20.07 is based on NVIDIA CUDA 11.0.194, which requires NVIDIA Driver release 450 or later. However, if you are running on Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 418.xx or 440.30. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 20.07 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- For Triton V2, add TensorFlow optimization option that enables automatic FP16 optimization of the model.
- For Triton V2, the PyTorch backend now includes support for TorchVision operations.
- This release includes support for both the new KFServing based protocols as well as the legacy V1 protocols.
- Support for the new KFServing HTTP/REST, GRPC and corresponding client libraries is released on GitHub branch r20.07 and as NGC container 20.07-py3.
- Support for the legacy V1 HTTP/REST, GRPC and corresponding client libraries is released on GitHub branch r20.07-v1 and as NGC container 20.07-v1-py3.
- Migration from Triton V1 to Triton V2 requires significant changes; see the “Backwards Compatibility” and “Roadmap” sections of the GitHub README for more information.
- Refer to the 20.07 column of the Frameworks Support Matrix for container image versions that the 20.07 inference server container is based on.
- Ubuntu 18.04 with June 2020 updates.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.07	1.15.0 2.1.0	18.04	NVIDIA CUDA 11.0.194	TensorRT 7.1.3
20.06	1.14.0 2.0.0		NVIDIA CUDA 11.0.167	TensorRT 7.1.2
20.03.1	1.13.0		NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.03	1.12.0			
20.02	1.11.0			
20.01	1.10.0			
19.12	1.9.0			TensorRT 6.0.1

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- When using the TensorRT NGC container to generate TensorRT models for Triton, the 20.07.1 version of the TensorRT container must be used to ensure compatibility with Triton 20.07.
- The KFServing HTTP/REST and GRPC protocols and corresponding Python and C ++ clients are beta quality and are likely to change.
- The new C API specified in `tritonserver.h` is beta quality and is likely to change.
- TensorRT reformat-free I/O is not supported.
- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.

Chapter 63. Triton Inference Server Release 20.06

The Triton Inference Server container image, release 20.06, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 18.04 including Python 3.6
- NVIDIA CUDA 11.0.167 including cuBLAS 11.1.0
- NVIDIA cuDNN 8.0.1
- NVIDIA NCCL 2.7.5 (optimized for NVLink™)
- MLNX_OFED
- OpenMPI 3.1.6
- TensorRT 7.1.2

Driver Requirements

Release 20.06 is based on NVIDIA CUDA 11.0.167, which requires NVIDIA Driver release 450 or later. However, if you are running on Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 418.xx or 440.30. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 20.06 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, Turing, and Ampere Architecture GPU families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Updates for KFserving HTTP/REST and GRPC protocols and corresponding Python and C++ client libraries. This release includes support for both the new KFServing based protocols as well as the legacy V1 protocols.
- Support for the new KFServing HTTP/REST, GRPC and corresponding client libraries is released on GitHub branch r20.06 and as NGC container 20.06-py3.
- Support for the legacy V1 HTTP/REST, GRPC and corresponding client libraries is released on GitHub branch r20.06-v1 and as NGC container 20.06-v1-py3.
- Migration from Triton V1 to Triton V2 requires significant changes; see the “Backwards Compatibility” and “Roadmap” sections of the GitHub README for more information.
- Refer to the 20.06 column of the Frameworks Support Matrix for container image versions that the 20.06 inference server container is based on.
- The latest version of NVIDIA CUDA 11.0.167 including cuBLAS 11.1.0
- The latest version of NVIDIA cuDNN 8.0.1
- The latest version of NVIDIA NCCL 2.7.5
- The latest version of OpenMPI 3.1.6
- The latest version of TensorRT 7.1.2
- Ubuntu 18.04 with May 2020 updates.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

<table>
<tr><th>Container Version</th><th>Triton Inference Server</th><th>Ubuntu</th><th>CUDA Toolkit</th><th>TensorRT</th></tr>
<tr><td>20.06</td><td>1.14.0
2.0.0</td><td rowspan="6">18.04</td><td>NVIDIA CUDA 11.0.167</td><td>TensorRT 7.1.2</td></tr>
<tr><td>20.03.1</td><td>1.13.0</td><td rowspan="4">NVIDIA CUDA 10.2.89</td><td rowspan="3">TensorRT 7.0.0</td></tr>
<tr><td>20.03</td><td>1.12.0</td></tr>
<tr><td>20.02</td><td>1.11.0</td></tr>
<tr><td>20.01</td><td>1.10.0</td></tr>
<tr><td>19.12</td><td>1.9.0</td><td>TensorRT 6.0.1</td></tr>
</table>

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- The KFServing HTTP/REST and GRPC protocols and corresponding V2 experimental Python and C++ clients are beta quality and are likely to change.
- The new C API specified in `tritonserver.h` is beta quality and is likely to change.
- TensorRT reformat-free I/O is not supported.
- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.

Chapter 64. Triton Inference Server Release 20.03.1

The Triton Inference Server container image, release 20.03.1, is available on NGC and is open source on GitHub.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 18.04 including Python 3.6
- NVIDIA CUDA 10.2.89 including cuBLAS 10.2.2.89
- NVIDIA cuDNN 7.6.5
- NVIDIA NCCL 2.6.3 (optimized for NVLink™)
- MLNX_OFED
- OpenMPI 3.1.4
- TensorRT 7.0.0

Driver Requirements

Release 20.03.1 is based on NVIDIA CUDA 10.2.89, which requires NVIDIA Driver release 440.33.01. However, if you are running on Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 396, 384.111+, 410, 418.xx or 440.30. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 20.03.1 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the NVIDIA Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Updates for KFserving HTTP/REST and GRPC protocols and corresponding Python and C++ client libraries. See the Roadmap section of the README for more information.
- Updated GRPC version to 1.24.0.
- Several issues with S3 storage were resolved.
- Fixed `last_inferrence_timestamp` value to correctly show the time when inference last occurred for each model.
- The Caffe2 backend is deprecated. Support for Caffe2 models will be removed in a future release.
- Refer to the 20.03 column of the Frameworks Support Matrix for container image versions that the 20.03.1 inference server container is based on.
- The inference server container image version 20.03.1 is additionally based on ONNX Runtime 1.2.0.
- Ubuntu 18.04 with April 2020 updates.

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

<table>
<tr><th>Container Version</th><th>Triton Inference Server</th><th>Ubuntu</th><th>CUDA Toolkit</th><th>TensorRT</th></tr>
<tr><td>20.03.1</td><td>1.13.0</td><td rowspan="9">18.04</td><td rowspan="6">NVIDIA CUDA 10.2.89</td><td rowspan="3">TensorRT 7.0.0</td></tr>
<tr><td>20.03</td><td>1.12.0</td></tr>
<tr><td>20.02</td><td>1.11.0</td></tr>
<tr><td>20.01</td><td>1.10.0</td></tr>
<tr><td>19.12</td><td>1.9.0</td><td rowspan="3">TensorRT 6.0.1</td></tr>
<tr><td>19.11</td><td>1.8.0</td></tr>
<tr><td>19.10</td><td>1.7.0</td><td rowspan="3">NVIDIA CUDA 10.1.243</td></tr>
<tr><td>19.09</td><td>1.6.0</td></tr>
<tr><td>19.08</td><td>1.5.0</td><td>TensorRT 5.1.5</td></tr>
</table>

Known Issues

- The KFServing HTTP/REST and GRPC protocols and corresponding V2 experimental Python and C++ clients are beta quality and are likely to change. Specifically:
  - The data returned by the statistics API will be changing to include additional information.
  - The data returned by the repository index API will be changing to include additional information.
- The new C API specified in `tritonserver.h` is beta quality and is likely to change.
- When using the experimental V2 HTTP/REST C++ client, classification results are not supported for output tensors.
- When using the experimental V2 perf_client_v2, for high concurrency values perf_client_v2 may not be able to achieve throughput as high as V1 perf_client.
- TensorRT reformat-free I/O is not supported.
- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.

Chapter 65. Triton Inference Server Release 20.03

The Triton Inference Server container image, release 20.03, is available on NGC and is open source on GitHub.

Starting in release 20.03, TensorRT Inference Server is now called Triton Inference Server.

Contents of the Triton Inference Server container

The Triton Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tritonserver`.

The container also includes the following:

- Ubuntu 18.04 including Python 3.6
- NVIDIA CUDA 10.2.89 including cuBLAS 10.2.2.89
- NVIDIA cuDNN 7.6.5
- NVIDIA NCCL 2.6.3 (optimized for NVLink™)
- MLNX_OFED
- OpenMPI 3.1.4
- TensorRT 7.0.0

Driver Requirements

Release 20.03 is based on NVIDIA CUDA 10.2.89, which requires NVIDIA Driver release 440.33.01. However, if you are running on Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 396, 384.111+, 410, 418.xx or 440.30. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 20.03 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Added queuing policies for dynamic batching scheduler. These policies are specified in the model configuration and allow each model to set maximum queue size, time outs, and priority levels for inference requests.
- Support for large ONNX models where weights are stored in separate files.
- Allow ONNX Runtime optimization level to be configured via the model configuration optimization setting.
- Experimental Python client and server support for community standard GRPC inferencing API.
- Added `--min-supported-compute-capability` flag to allow Triton Server to use older, unsupported GPUs.
- Fixed `perf_client` shared memory support. In some cases the shared-memory option did not work correctly due to the input and output tensor names. This issue is now resolved.
- Refer to the Frameworks Support Matrix for container image versions that the 20.03 inference server container is based on.
- The inference server container image version 20.03 is additionally based on ONNX Runtime 1.1.1.
- Ubuntu 18.04 with February 2020 updates

NVIDIA Triton Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, Triton Inference Server, and TensorRT are supported in each of the NVIDIA containers for Triton Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
20.03	1.12.0	18.04 16.04	NVIDIA CUDA 10.2.89	TensorRT 7.0.0
20.02	1.11.0			
20.01	1.10.0			

Container Version	Triton Inference Server	Ubuntu	CUDA Toolkit	TensorRT
19.12	1.9.0			TensorRT 6.0.1
19.11	1.8.0			
19.10	1.7.0		NVIDIA CUDA 10.1.243	
19.09	1.6.0			
19.08	1.5.0			TensorRT 5.1.5

Known Issues

- TensorRT reformat-free I/O is not supported.
- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.

Chapter 66. Triton Inference Server Release 20.02

The TensorRT Inference Server container image, release 20.02, is available on NGC and is open source on GitHub.

Contents of the Triton inference server container

The TensorRT Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 18.04 including Python 3.6
- NVIDIA CUDA 10.2.89 including cuBLAS 10.2.2.89
- NVIDIA cuDNN 7.6.5
- NVIDIA NCCL 2.5.6 (optimized for NVLink™)
- MLNX_OFED
- OpenMPI 3.1.4
- TensorRT 7.0.0

Driver Requirements

Release 20.02 is based on NVIDIA CUDA 10.2.89, which requires NVIDIA Driver release 440.33.01. However, if you are running on Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 396, 384.111+, 410, 418.xx or 440.30. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 20.02 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Refer to the Frameworks Support Matrix for container image versions that the 20.02 inference server container is based on.
- The inference server container image version 20.02 is additionally based on ONNX Runtime 1.1.1.
- The TensorRT backend is improved to have significantly better performance. Improvements include reducing thread contention, using pinned memory for faster CPU<->GPU transfers, and increasing compute and memory copy overlap on GPUs.
- Reduce memory usage of TensorRT models in many cases by sharing weights across multiple model instances.
- Boolean data-type and shape tensors are now supported for TensorRT models.
- A new model configuration option allows the dynamic batcher to create “ragged” batches for custom backend models. A ragged batch is a batch where one or more of the input/output tensors have different shapes in different batch entries.
- Local S3 storage endpoints are now supported for model repositories. A local S3 endpoint is specified as `s3://host:port/path/to/repository`.
- The Helm chart showing an example Kubernetes deployment is updated to include Prometheus and Grafana support so that inference server metrics can be collected and visualized.
- The inference server container no longer sets LD_LIBRARY_PATH, instead the server uses RUNPATH to locate its shared libraries.
- Python 2 is end-of-life so all support has been removed. Python 3 is still supported.
- Ubuntu 18.04 with January 2020 updates

NVIDIA TensorRT Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, TensorRT Inference Server, and TensorRT are supported in each of the NVIDIA containers for TensorRT Inference Server. For older container versions, refer to the Frameworks Support Matrix.

Container Version	Ubuntu	CUDA Toolkit	TensorRT Inference Server	TensorRT
20.02	18.04	NVIDIA CUDA 10.2.89	1.12.0	TensorRT 7.0.0

Container Version	Ubuntu	CUDA Toolkit	TensorRT Inference Server	TensorRT
20.01			1.11.0	
			1.10.0	
19.12			1.9.0	TensorRT 6.0.1
19.11			1.8.0	
19.10		NVIDIA CUDA 10.1.243	1.7.0	
19.09			1.6.0	
19.08			1.5.0	TensorRT 5.1.5

Known Issues

- TensorRT reformat-free I/O is not supported.
- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.

Chapter 67. Triton Inference Server Release 20.01

The TensorRT Inference Server container image, release 20.01, is available on NGC and is open source on GitHub.

Contents of the Triton inference server container

The TensorRT Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 18.04 including Python 3.6
- NVIDIA CUDA 10.2.89 including cuBLAS 10.2.2.89
- NVIDIA cuDNN 7.6.5
- NVIDIA NCCL 2.5.6 (optimized for NVLink™)
- MLNX_OFED
- OpenMPI 3.1.4
- TensorRT 7.0.0

Driver Requirements

Release 20.01 is based on NVIDIA CUDA 10.2.89, which requires NVIDIA Driver release 440.33.01. However, if you are running on Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 396, 384.111+, 410, 418.xx or 440.30. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 20.01 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Refer to the Frameworks Support Matrix for container image versions that the 20.01 inference server container is based on.
- The inference server container image version 20.01 is additionally based on ONNX Runtime 1.1.1.
- Server status can be requested in JSON format using the HTTP/REST API. Use endpoint `/api/status?format=json`.
- The dynamic batcher now has an option to preserve the ordering of batched requests when there are multiple model instances. See model_config.proto for more information.
- Latest version of TensorRT 7.0.0
- Ubuntu 18.04 with December 2019 updates

NVIDIA TensorRT Inference Server Container Versions

The following table shows what versions of Ubuntu, CUDA, TensorRT Inference Server, and TensorRT are supported in each of the NVIDIA containers for TensorRT Inference Server. For older container versions, refer to the Frameworks Support Matrix.

<table>
<tr><th>Container Version</th><th>Ubuntu</th><th>CUDA Toolkit</th><th>TensorRT Inference Server</th><th>TensorRT</th></tr>
<tr><td>20.01</td><td rowspan="6">18.04
16.04</td><td rowspan="3">NVIDIA CUDA 10.2.89</td><td>1.10.0</td><td>TensorRT 7.0.0</td></tr>
<tr><td>19.12</td><td>1.9.0</td><td rowspan="4">TensorRT 6.0.1</td></tr>
<tr><td>19.11</td><td>1.8.0</td></tr>
<tr><td>19.10</td><td rowspan="2">NVIDIA CUDA 10.1.243</td><td>1.7.0</td></tr>
<tr><td>19.09</td><td>1.6.0</td></tr>
<tr><td>19.08</td><td>1.5.0</td><td>TensorRT 5.1.5</td></tr>
</table>

Known Issues

- TensorRT reformat-free I/O is not supported.
- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.

Chapter 68. Triton Inference Server Release 19.12

The TensorRT Inference Server container image, release 19.12, is available on NGC and is open source on GitHub.

Contents of the Triton inference server container

The TensorRT Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 18.04 including Python 3.6
- NVIDIA CUDA 10.2.89 including cuBLAS 10.2.2.89
- NVIDIA cuDNN 7.6.5
- NVIDIA NCCL 2.5.6 (optimized for NVLink™)
- MLNX_OFED
- OpenMPI 3.1.4
- TensorRT 6.0.1

Driver Requirements

Release 19.12 is based on NVIDIA CUDA 10.2.89, which requires NVIDIA Driver release 440.30. However, if you are running on Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 396, 384.111+, 410, 418.xx or 440.30.. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.12 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Refer to the Frameworks Support Matrix for container image versions that the 19.12 inference server container is based on.
- The inference server container image version 19.12 is additionally based on ONNX Runtime 1.1.1.
- The model configuration now includes a *model warmup* option. This option provides the ability to tune and optimize the model before inference requests are received, avoiding initial inference delays. This option is especially useful for frameworks like TensorFlow that perform network optimization in response to the initial inference requests. Models can be warmed-up with one or more synthetic or realistic workloads before they become ready in the server
- An enhanced sequence batcher now has multiple scheduling strategies. A new *Oldest* strategy integrates with the dynamic batcher to enable improved inference performance for models that don't require all inference requests in a sequence to be routed to the same batch slot.
- The `perf_client` now has an option to generate requests using a realistic poisson distribution or a user provided distribution.
- A new repository API (available in the shared library API, HTTP, and gRPC) returns an index of all models available in the model repositories) visible to the server. This index can be used to see what models are available for loading onto the server.
- The server status returned by the server status API now includes the timestamp of the last inference request received for each model.
- Inference server tracing capabilities are now documented in the Optimization section of the *User Guide*. Tracing support is enhanced to provide trace for ensembles and the contained models.
- A community contributed Dockerfile is now available to build the TensorRT Inference Server clients on CentOS.
- Ubuntu 18.04 with November2019 updates

Known Issues

- The beta of the custom backend API version 2 has non-backwards compatible changes to enable complete support for input and output tensors in both CPU and GPU memory:
  - The signature of the `CustomGetNextInputV2Fn_t` function adds the `memory_type_id` argument.
  - The signature of the `CustomGetOutputV2Fn_t` function adds the `memory_type_id` argument.

- The beta of the inference server library API has non-backwards compatible changes to enable complete support for input and output tensors in both CPU and GPU memory:
  - The signature and operation of the `TRTSERVER_ResponseAllocatorAllocFn_t` function has changed. See `src/core/trtserver.h` for a description of the new behavior.
  - The signature of the `TRTSERVER_InferenceRequestProviderSetInputData` function adds the `memory_type_id` argument.
  - The signature of the `TRTSERVER_InferenceResponseOutputData` function add the `memory_type_id` argument.
- TensorRT reformat-free I/O is not supported.
- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.

Chapter 69. Triton Inference Server Release 19.11

The TensorRT Inference Server container image, release 19.11, is available on NGC and is open source on GitHub.

Contents of the Triton inference server container

The TensorRT Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 18.04 including Python 3.6
- NVIDIA CUDA 10.2.89 including cuBLAS 10.2.2.89
- NVIDIA cuDNN 7.6.5
- NVIDIA NCCL 2.5.6 (optimized for NVLink™)
- MLNX_OFED
- OpenMPI 3.1.4
- TensorRT 6.0.1

Driver Requirements

Release 19.11 is based on NVIDIA CUDA 10.2.89, which requires NVIDIA Driver release 440.30. However, if you are running on Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 396, 384.111+, 410 or 418.xx. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.11 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- Refer to the Frameworks Support Matrix for container image versions that the 19.11 inference server container is based on.
- The inference server container image version 19.11 is additionally based on ONNX Runtime 1.1.1.
- Shared-memory support is expanded to include CUDA shared memory.
- Improve efficiency of pinned-memory used for ensemble models.
- The `perf_client` application has been improved with easier-to-use command-line arguments (while maintaining compatibility with existing arguments).
- Support for string tensors added to `perf_client`.
- Documentation contains a new *Optimization* section discussing some common optimization strategies and how to use `perf_client` to explore these strategies.
- Latest version of NVIDIA CUDA 10.2.89 including cuBLAS 10.2.2.89
- Latest version of NVIDIA cuDNN 7.6.5
- Latest version of NVIDIA NCCL 2.5.6
- Ubuntu 18.04 with October 2019 updates

Deprecated Features

- The asynchronous inference API has been modified in the C++ and Python client libraries.
  - In the C++ library:
    - The non-callback version of the `AsyncRun` function is removed.
    - The `GetReadyAsyncRequest` function is removed.
    - The signature of the `GetAsyncRunResults` function was changed to remove the `is_ready` and `wait` arguments.
  - In the Python library:
    - The non-callback version of the `async_run` function was removed.
    - The `get_ready_async_request` function was removed.
    - The signature of the `get_async_run_results` function was changed to remove the `wait` argument.

Known Issues

- The beta of the custom backend API version 2 has non-backwards compatible changes to enable complete support for input and output tensors in both CPU and GPU memory:

  - The signature of the `CustomGetNextInputV2Fn_t` function adds the `memory_type_id` argument.
  - The signature of the `CustomGetOutputV2Fn_t` function adds the `memory_type_id` argument.
- The beta of the inference server library API has non-backwards compatible changes to enable complete support for input and output tensors in both CPU and GPU memory:
  - The signature and operation of the `TRTSERVER_ResponseAllocatorAllocFn_t` function has changed. See `src/core/trtserver.h` for a description of the new behavior.
  - The signature of the `TRTSERVER_InferenceRequestProviderSetInputData` function adds the `memory_type_id` argument.
  - The signature of the `TRTSERVER_InferenceResponseOutputData` function add the `memory_type_id` argument.
- TensorRT reformat-free I/O is not supported.
- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.

Chapter 70. Triton Inference Server Release 19.10

The TensorRT Inference Server container image, release 19.10, is available on NGC and is open source on GitHub.

Contents of the Triton inference server container

The TensorRT Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 18.04
- NVIDIA CUDA 10.1.243 including cuBLAS 10.2.1.243
- NVIDIA cuDNN 7.6.4
- NVIDIA NCCL 2.4.8 (optimized for NVLink™)
- MLNX_OFED
- OpenMPI 3.1.4
- TensorRT 6.0.1

Driver Requirements

Release 19.10 is based on NVIDIA CUDA 10.1.243, which requires NVIDIA Driver release 418.xx. However, if you are running on Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 396, 384.111+ or 410. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.10 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The inference server container image version 19.10 is based on NVIDIA TensorRT Inference Server 1.7.0, TensorFlow 1.14.0, ONNX Runtime 0.4.0, and PyTorch 1.3.0.
- A Client SDK container is now provided on NGC in addition to the inference server container. The client SDK container includes the client libraries and examples.
- Latest version of NVIDIA cuDNN 7.6.4
- TensorRT optimization may now be enabled for any TensorFlow model by enabling the feature in the optimization section of the model configuration.
- The ONNXRuntime backend now includes the TensorRT and Open Vino execution providers. These providers are enabled in the optimization section of the model configuration.
- Automatic configuration generation (`--strict-model-config=false`) now works correctly for TensorRT models with variable-sized inputs and/or outputs.
- Multiple model repositories may now be specified on the command line. Optional command-line options can be used to explicitly load specific models from each repository.
- Ensemble models are now pruned dynamically so that only models needed to calculate the requested outputs are executed.
- The example clients now include a simple Go example that uses the GRPC API.
- Ubuntu 18.04 with September 2019 updates

Known Issues

- In TensorRT 6.0.1, reformat-free I/O is not supported.
- Some versions of Google Kubernetes Engine (GKE) contain a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version or a GKE 1.14.6 or later version to avoid this issue.

Chapter 71. Triton Inference Server Release 19.09

The TensorRT Inference Server container image, release 19.09, is available on NGC and is open source on GitHub.

Contents of the Triton inference server container

The TensorRT Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 18.04
- NVIDIA CUDA 10.1.243 including cuBLAS 10.2.1.243
- NVIDIA cuDNN 7.6.3
- NVIDIA NCCL 2.4.8 (optimized for NVLink™)
- MLNX_OFED
- OpenMPI 3.1.4
- TensorRT 6.0.1

Driver Requirements

Release 19.09 is based on NVIDIA CUDA 10.1.243, which requires NVIDIA Driver release 418.xx. However, if you are running on Tesla (for example, T4 or any other Tesla board), you may use NVIDIA driver release 396, 384.111+ or 410. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.09 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The inference server container image version 19.09 is based on NVIDIA TensorRT Inference Server 1.6.0, TensorFlow 1.14.0, ONNX Runtime 0.4.0, and PyTorch 1.2.0.
- Latest version of NVIDIA cuDNN 7.6.3
- Latest version of TensorRT 6.0.1
- Added TensorRT 6 support, which includes support for TensorRT dynamic shapes
- Shared memory support is added as an alpha feature in this release. This support allows input and output tensors to be communicated via shared memory instead of over the network. Currently only system (CPU) shared memory is supported.
- Amazon S3 is now supported as a remote file system for model repositories. Use the `s3://` prefix on model repository paths to reference S3 locations.
- The inference server library API is available as a beta in this release. The library API allows you to link against `libtrtserver.so` so that you can include all the inference server functionality directly in your application.
- GRPC endpoint performance improvement. The inference server's GRPC endpoint now uses significantly less memory while delivering higher performance.
- The ensemble scheduler is now more flexible in allowing batching and non-batching models to be composed together in an ensemble.
- The ensemble scheduler will now keep tensors in GPU memory between models when possible. Doing so significantly increases performance of some ensembles by avoiding copies to and from system memory.
- The performance client, `perf_client`, now supports models with variable-sized input tensors.
- Ubuntu 18.04 with August 2019 updates

Known Issues

- The ONNX Runtime backend could not be updated to the 0.5.0 release due to multiple performance and correctness issues with that release.
- TensorRT 6:
  - Reformat-free I/O is not supported.
  - Only models that have a single optimization profile are currently supported.
- Google Kubernetes Engine (GKE) version 1.14 contains a regression in the handling of `LD_LIBRARY_PATH` that prevents the inference server container from running correctly (see issue 141255952). Use a GKE 1.13 or earlier version to avoid this issue.

Chapter 72. Triton Inference Server Release 19.08

The TensorRT Inference Server container image, release 19.08, is available on NGC and is open source on GitHub.

Contents of the Triton inference server container

The TensorRT Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 18.04
- NVIDIA CUDA 10.1.243 including cuBLAS 10.2.1.243
- NVIDIA cuDNN 7.6.2
- NVIDIA NCCL 2.4.8 (optimized for NVLink™)
- MLNX_OFED +4.0
- OpenMPI 3.1.4
- TensorRT 5.1.5

Driver Requirements

Release 19.08 is based on NVIDIA CUDA 10.1.243, which requires NVIDIA Driver release 418.87. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384.111+ or 410. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.08 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The inference server container image version 19.08 is based on NVIDIA TensorRT Inference Server 1.5.0, TensorFlow 1.14.0, ONNX Runtime 0.4.0, and PyTorch 1.2.0a0.
- Added a new execution mode allows the inference server to start without loading any models from the model repository. Model loading and unloading is then controlled by a new GRPC/HTTP model control API.
- Added a new instance-group mode allows TensorFlow models that explicitly distribute inferencing across multiple GPUs to run in that manner in the inference server.
- Improved input/output tensor reshape to allow variable-sized dimensions in tensors being reshaped.
- Added a C++ wrapper around the custom backend C API to simplify the creation of custom backends. This wrapper is included in the custom backend SDK.
- Improved the accuracy of the *compute* statistic reported for inference requests. Previously the compute statistic included some additional time beyond the actual compute time.
- The performance client, `perf_client`, now reports more information for ensemble models, including statistics for all contained models and the entire ensemble.
- Latest version of NVIDIA CUDA 10.1.243 including cuBLAS 10.2.1.243
- Latest version of NVIDIA cuDNN 7.6.2
- Latest version of NVIDIA NCCL 2.4.8
- Latest version of MLNX_OFED +4.0
- Latest version of OpenMPI 3.1.4
- Ubuntu 18.04 with July 2019 updates

Known Issues

There are no known issues in this release.

Chapter 73. Triton Inference Server Release 19.07

The TensorRT Inference Server container image, release 19.07, is available on NGC and is open source on GitHub.

Contents of the Triton inference server container

The TensorRT Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 18.04
- NVIDIA CUDA 10.1.168 including cuBLAS 10.2.0.168
- NVIDIA cuDNN 7.6.1
- NVIDIA NCCL 2.4.7 (optimized for NVLink™)
- MLNX_OFED +3.4
- OpenMPI 3.1.3
- TensorRT 5.1.5

Driver Requirements

Release 19.07 is based on NVIDIA CUDA 10.1.168, which requires NVIDIA Driver release 418.67. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384.111+ or 410. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.07 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The inference server container image version 19.07 is based on NVIDIA TensorRT Inference Server 1.4.0, TensorFlow 1.14.0, ONNX Runtime 0.4.0, and Caffe2 0.8.2.
- Added `libtorch` as a new backend. PyTorch models manually decorated or automatically traced to produce TorchScript can now be run directly by the inference server.
- Build system converted from bazel to CMake. The new CMake-based build system is more transparent, portable and modular.
- To simplify the creation of custom backends, a Custom Backend SDK and improved documentation is now available.
- Improved AsyncRun API in C++ and Python client libraries.
- `perf_client` can now use user-supplied input data (previously used random or zero input data).
- `perf_client` now reports latency at multiple confidence percentiles (p50, p90, p95, p99) as well as a user-supplied percentile that is also used to stabilize latency results.
- Improvements to automatic model configuration creation (`--strict-model-config=false`).
- C++ and Python client libraries now allow additional HTTP headers to be specified when using the HTTP protocol.
- Latest version of NVIDIA cuDNN 7.6.1
- Latest version of MLNX_OFED +3.4
- Latest version of Ubuntu 18.04

Known Issues

There are no known issues in this release.

Chapter 74. Triton Inference Server Release 19.06

The TensorRT Inference Server container image, release 19.06, is available on NGC and is open source on GitHub.

Contents of the Triton inference server container

The TensorRT Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 10.1.168 including cuBLAS 10.2.0.168
- NVIDIA cuDNN 7.6.0
- NVIDIA NCCL 2.4.7 (optimized for NVLink™)
- OpenMPI 3.1.3
- TensorRT 5.1.5

Driver Requirements

Release 19.06 is based on NVIDIA CUDA 10.1.168, which requires NVIDIA Driver release 418.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384.111+ or 410. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.06 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The inference server container image version 19.06 is based on NVIDIA TensorRT Inference Server 1.3.0, TensorFlow 1.13.1, ONNX Runtime 0.4.0, and Caffe2 0.8.2.
- Latest version of NVIDIA CUDA 10.1.168 including cuBLAS 10.2.0.168
- Latest version of NVIDIA NCCL 2.4.7
- Added ONNX Runtime as a new backend. The ONNX Runtime backend allows the inference server to directly run ONNX models without requiring conversion to Caffe2 or TensorRT.
- HTTP health port may be specified independently of inference and status HTTP port with `--http-health-port` flag.
- Fixed bug in `perf_client` that caused high CPU usage by client that could lower the measured inference/sec in some cases.
- Ubuntu 16.04 with May 2019 updates (see Announcements)

Announcements

In the next release, we will no longer support Ubuntu 16.04. Release 19.07 will instead support Ubuntu 18.04.

Known Issues

- Google Cloud Storage (GCS) support is not available in this release. Support for GCS will be re-enabled in the 19.07 release.

Chapter 75. Triton Inference Server Release 19.05

The TensorRT Inference Server container image, release 19.05, is available on NGC and is open source on GitHub.

Contents of the Triton inference server container

The TensorRT Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 10.1 Update 1 including cuBLAS 10.1 Update 1
- NVIDIA cuDNN 7.6.0
- NVIDIA NCCL 2.4.6 (optimized for NVLink™)
- OpenMPI 3.1.3
- TensorRT 5.1.5

Driver Requirements

Release 19.05 is based on CUDA 10.1 Update 1, which requires NVIDIA Driver release 418.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384.111+ or 410. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.05 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The inference server container image version 19.05 is based on NVIDIA TensorRT Inference Server 1.2.0, TensorFlow 1.13.1, and Caffe2 0.8.2.
- Latest version of NVIDIA CUDA 10.1 Update 1 including cuBLAS 10.1 Update 1
- Latest version of NVIDIA cuDNN 7.6.0
- Latest version of TensorRT 5.1.5
- Ensembling is now available. An ensemble represents a pipeline of one or more models and the connection of input and output tensors between those models. A single inference request to an ensemble will trigger the execution of the entire pipeline.
- Added a Helm chart that deploys a single TensorRT Inference Server into a Kubernetes cluster.
- The client `Makefile` now supports building for both Ubuntu 18.04 and Ubuntu 16.04. The Python wheel produced from the build is now compatible with both Python2 and Python3.
- The `perf_client` application now has a `--percentile` flag that can be used to report latencies instead of reporting average latency (which remains the default). For example, using `--percentile=99` causes `perf_client` to report the 99th percentile latency.
- The `perf_client` application now has a `-z` option to use zero-valued input tensors instead of random values.
- Improved error reporting of incorrect input/output tensor names for TensorRT models.
- Added `--allow-gpu-metrics` option to enable/disable reporting of GPU metrics.

Known Issues

There are no known issues in this release.

Chapter 76. Triton Inference Server Release 19.04

The TensorRT Inference Server container image, release 19.04, is available on NGC and is open source on GitHub.

Contents of the Triton inference server container

The TensorRT Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 10.1.105 including cuBLAS 10.1.0.105
- NVIDIA cuDNN 7.5.0
- NVIDIA NCCL 2.4.6 (optimized for NVLink™)
- OpenMPI 3.1.3
- TensorRT 5.1.2 RC

Driver Requirements

Release 19.04 is based on CUDA 10.1, which requires NVIDIA Driver release 418.xx.x+. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384.111+ or 410. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.04 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The inference server container image version 19.04 is based on NVIDIA TensorRT Inference Server 1.1.0, TensorFlow 1.13.1, and Caffe2 0.8.2.
- Latest version of NVIDIA NCCL 2.4.6
- Latest version of cuBLAS 10.1.0.105
- Client libraries and examples now build with a separate Makefile (a Dockerfile is also included for convenience).
- Input or output tensors with variable-size dimensions (indicated by -1 in the model configuration) can now represent tensors where the variable dimension has value 0 (zero).
- Zero-sized input and output tensors are now supported for batching models. This enables the inference server to support models that require inputs and outputs that have shape `[ batch-size ]`.
- TensorFlow custom operations (C++) can now be built into the inference server. An example and documentation are included in this release.
- Ubuntu 16.04 with March 2019 updates

Known Issues

There are no known issues in this release.

Chapter 77. Triton Inference Server Release 19.03

The TensorRT Inference Server container image, release 19.03, is available on NGC and is open source on GitHub.

Contents of the Triton inference server container

The TensorRT Inference Server Docker image contains the inference server executable and related shared libraries in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 10.1.105 including cuBLAS 10.1.105
- NVIDIA cuDNN 7.5.0
- NVIDIA NCCL 2.4.3 (optimized for NVLink™)
- OpenMPI 3.1.3
- TensorRT 5.1.2 RC

Driver Requirements

Release 19.03 is based on CUDA 10.1, which requires NVIDIA Driver release 418.xx+. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384.111+ or 410. The CUDA driver's compatibility package only supports particular drivers. For a complete list of supported drivers, see the CUDA Application Compatibility topic. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.03 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The inference server container image version 19.03 is based on NVIDIA TensorRT Inference Server 1.0.0, TensorFlow 1.13.1, and Caffe2 0.8.2.
- 19.03 is the first GA release of TensorRT Inference Server. See the README at the GitHub project for information on backwards-compatibility guarantees for this and future releases.
- Added support for “stateful” models and backends that require multiple inference requests be routed to the same model instance/batch slot. The new *sequence batcher* provides scheduling and batching capabilities for this class of models.
- Added GRPC streaming protocol support for inference requests.
- HTTP front-end is now asynchronous to enable lower-latency and higher-throughput handling of inference requests.
- Enhanced `perf_client` to support “stateful”/sequence models and backends.
- Latest version of NVIDIA CUDA 10.1.105 including cuBLAS 10.1.105
- Latest version of NVIDIA cuDNN 7.5.0
- Latest version of NVIDIA NCCL 2.4.3
- Latest version of TensorRT 5.1.2 RC
- Ubuntu 16.04 with February 2019 updates

Known Issues

- If using or upgrading to a 3-part-version driver, for example, a driver that takes the format of `xxx.yy.zz`, you will receive a `Failed to detect NVIDIA driver version.` message. This is due to a known bug in the entry point script's parsing of 3-part driver versions. This message is non-fatal and can be ignored. This will be fixed in the 19.04 release.

Chapter 78. Triton Inference Server Release 19.02 Beta

The TensorRT Inference Server container image, release 19.02, is available as a beta release and is open source on GitHub.

Contents of the Triton inference server

This container image contains the TensorRT Inference Server executable and related shared libraries in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.4.2
- NVIDIA Collective Communications Library (NCCL) 2.3.7 (optimized for NVLink™)
- OpenMPI 3.1.3
- TensorRT 5.0.2

Driver Requirements

Release 19.02 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.02 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The inference server container image version 19.02 is based on NVIDIA TensorRT Inference Server 0.11.0 beta, TensorFlow 1.13.0-rc0, and Caffe2 0.8.2.
- Variable-size input and output tensors are now supported.
- `STRING` datatype is now supported for input and output tensors for TensorFlow models and custom backends.
- The inference server can now be run on systems without GPUs or that do not have CUDA installed.
- Ubuntu 16.04 with January 2019 updates

Known Issues

- This is a beta release of the Inference Server. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.
- If using or upgrading to a 3-part-version driver, for example, a driver that takes the format of `xxx.yy.zz`, you will receive a `Failed to detect NVIDIA driver version.` message. This is due to a known bug in the entry point script's parsing of 3-part driver versions. This message is non-fatal and can be ignored. This will be fixed in the 19.04 release.

Chapter 79. Triton Inference Server Release 19.01 Beta

The TensorRT Inference Server container image, release 19.01, is available as a beta release and is open source on GitHub.

Contents of the Triton inference server

This container image contains the TensorRT Inference Server executable and related shared libraries in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 16.04
- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.4.2
- NCCL 2.3.7 (optimized for NVLink™)
- OpenMPI 3.1.3
- TensorRT 5.0.2

Driver Requirements

Release 19.01 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 19.01 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The inference server container image version 19.01 is based on NVIDIA TensorRT Inference Server 0.10.0 beta, TensorFlow 1.12.0, and Caffe2 0.8.2.
- Latest version of NVIDIA cuDNN 7.4.2
- Latest version of OpenMPI 3.1.3
- Custom backend support. The inference server allows individual models to be implemented with custom backends instead of by a deep learning framework. With a custom backend, a model can implement any logic desired, while still benefiting from the GPU support, concurrent execution, dynamic batching and other features provided by the inference server.
- Ubuntu 16.04 with December 2018 updates

Known Issues

- This is a beta release of the Inference Server. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.
- If using or upgrading to a 3-part-version driver, for example, a driver that takes the format of `xxx.yy.zz`, you will receive a `Failed to detect NVIDIA driver version.` message. This is due to a known bug in the entry point script's parsing of 3-part driver versions. This message is non-fatal and can be ignored. This will be fixed in the 19.04 release.

Chapter 80. Triton Inference Server Release 18.12 Beta

The TensorRT Inference Server container image, previously referred to as Inference Server, release 18.12, is available as a beta release.

Contents of the Triton inference server

This container image contains the TensorRT Inference Server (TRTIS) executable and related shared libraries in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 16.04 including Python 3.5
- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.4.1
- NCCL 2.3.7 (optimized for NVLink™)
- OpenMPI 3.1.2
- TensorRT 5.0.2

Driver Requirements

Release 18.12 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

GPU Requirements

Release 18.12 supports CUDA compute capability 6.0 and higher. This corresponds to GPUs in the Pascal, Volta, and Turing families. Specifically, for a list of GPUs that this compute capability corresponds to, see CUDA GPUs. For additional support details, see Deep Learning Frameworks Support Matrix.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The Inference Server container image version 18.12 is based on NVIDIA Inference Server 0.9.0 beta, TensorFlow 1.12.0, and Caffe2 0.8.2.
- TensorRT inference server is now open source. For more information, see GitHub.
- TRTIS now monitors the model repository for any change and dynamically reloads the model when necessary, without requiring a server restart. It is now possible to add and remove model versions, add/remove entire models, modify the model configuration, and modify the model labels while the server is running.
- Added a model priority parameter to the model configuration. Currently the model priority controls the CPU thread priority when executing the model and for TensorRT models also controls the CUDA stream priority.
- Fixed a bug in GRPC API: changed the model version parameter from string to int. **This is a non-backwards compatible change.**
- Added `--strict-model-config=false` option to allow some model configuration properties to be derived automatically. For some model types, this removes the need to specify the `config.pbtxt` file.
- Improved performance from an asynchronous GRPC frontend.
- Ubuntu 16.04 with November 2018 updates

Known Issues

This is a beta release of the Inference Server. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.

Chapter 81. Triton Inference Server Release 18.11 Beta

The inference server container image, previously referred to as Inference Server, release 18.11, is available as a beta release.

Contents of the Triton inference server

This container image contains the Triton inference server executable in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 16.04 including Python 3.5
- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.4.1
- NCCL 2.3.7 (optimized for NVLink™)
- OpenMPI 3.1.2
- TensorRT 5.0.2

Driver Requirements

Release 18.11 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The Inference Server container image version 18.11 is based on NVIDIA Inference Server 0.8.0 beta, TensorFlow 1.12.0-rc2, and Caffe2 0.8.2.
- Models may now be added to and removed from the model repository without requiring an inference server restart.

- Fixed an issue with models that don't support batching. For models that don't support batching, set the model configuration to `max_batch_size = 0`.
- Added a metric to indicate GPU energy consumption.
- Latest version of NCCL 2.3.7.
- Latest version of NVIDIA cuDNN 7.4.1.
- Latest version of TensorRT 5.0.2
- Ubuntu 16.04 with October 2018 updates

Known Issues

This is a beta release of the Inference Server. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.

Chapter 82. Triton Inference Server Release 18.10 Beta

The Inference Server container image, previously referred to as Inference Server, release 18.10, is available as a beta release.

Contents of the Triton inference server

This container image contains the Triton inference server executable in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 16.04 including Python 3.5
- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.3.0
- NCCL 2.3.6 (optimized for NVLink™)
- OpenMPI 3.1.2
- TensorRT 5.0.0 RC

Driver Requirements

Release 18.10 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The Inference Server container image version 18.10 is based on NVIDIA Inference Server 0.7.0 beta, TensorFlow 1.10.0, and Caffe2 0.8.2.
- Latest version of NCCL 2.3.6.
- Latest version of OpenMPI 3.1.2.

- Dynamic batching support is added for all model types. Dynamic batching can be enabled and configured on a per-model bases.
- An improved inference request scheduler provides better handling of inference requests.
- Added new metrics to indicate GPU power limit, GPU utilization, and model executions (which is useful for determining the impact of dynamic batching).
- Prometheus metrics are now tagged with GPU UUID, model name, and model version as appropriate, so that metric values can be correlated to specific GPUs and models.
- Request latencies reported by status API and metrics are more clear in what they report, for example total request time, queuing time, and inference compute time are now reported.
- Ubuntu 16.04 with September 2018 updates

Known Issues

This is a beta release of the Inference Server. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.

Chapter 83. Triton Inference Server Release 18.09 Beta

The Inference Server container image, previously referred to as Inference Server, release 18.09, is available as a beta release.

Contents of the Triton inference server

This container image contains the Triton inference server executable in `/opt/tensorrtserver`.

The container also includes the following:

- Ubuntu 16.04 including Python 3.5
- NVIDIA CUDA 10.0.130 including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 10.0.130
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.3.0
- NCCL 2.3.4 (optimized for NVLink™)
- OpenMPI 2.0
- TensorRT 5.0.0 RC

Driver Requirements

Release 18.09 is based on CUDA 10, which requires NVIDIA Driver release 410.xx. However, if you are running on Tesla (Tesla V100, Tesla P4, Tesla P40, or Tesla P100), you may use NVIDIA driver release 384. For more information, see CUDA Compatibility and Upgrades.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The Inference Server container image version 18.09 is based on NVIDIA Inference Server 0.6.0 beta, TensorFlow 1.10.0, and Caffe2 0.8.1.
- Latest version of cuDNN 7.3.0.

- Latest version of CUDA 10.0.130 which includes support for DGX-2, Turing, and Jetson Xavier.
- Latest version of cuBLAS 10.0.130.
- Latest version of NCCL 2.3.4.
- Latest version of TensorRT 5.0.0 RC.
- Google Cloud Storage paths are now allowed when specifying the location of the model store. For example, `--model-store=gs://<bucket>/<mode store path>`.
- Additional Prometheus metrics are exposed on the metrics endpoint: GPU power usage; GPU power limit; per-model request, queue and compute time.
- The C++ and Python client API now supports asynchronous requests.
- Ubuntu 16.04 with August 2018 updates

Known Issues

- This is a beta release of the Inference Server. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.
- Starting with the 18.09 release, the directory holding the Triton inference server components has changed from `/opt/inference_server` to `/opt/tensorrtserver` and the Triton inference server executable name has changed from `inference_server` to `trtserver`.

Chapter 84. Inference Server Release 18.08 Beta

The NVIDIA container image of the Inference Server, release 18.08, is available as a beta release.

Contents of the Inference Server

This container image contains the Inference Server executable in `/opt/inference_server`.

The container also includes the following:

- Ubuntu 16.04

> Container image `18.08-py2` contains Python 2.7; `18.08-py3` contains Python 3.5.

- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.425
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.2.1
- NCCL 2.2.13 (optimized for NVLink™)
- TensorRT 4.0.1

Driver Requirements

Release 18.08 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The Inference Server container image version 18.08 is based on NVIDIA Inference Server 0.5.0 beta, TensorFlow 1.9.0, and Caffe2 0.8.1.
- Latest version of cuDNN 7.2.1.
- Added support for Kubernetes compatible ready and live endpoints.

- Added support for Prometheus metrics. Load metric is reported that can be used for Kubernetes-style auto-scaling.
- Enhance example `perf_client` application to generate latency vs. inferences/second results.
- Improve performance of TensorRT models by allowing multiple TensorRT model instances to execute simultaneously.
- Improve HTTP client performance by reusing connections for multiple inference requests.
- Ubuntu 16.04 with July 2018 updates

Known Issues

- This is a beta release of the Inference Server. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.
- There is a known performance regression in the inference benchmarks for ResNet-50. We haven't seen this regression in the inference benchmarks for VGG or training benchmarks for any network. The cause of the regression is still under investigation.

Chapter 85. Inference Server Release 18.07 Beta

The NVIDIA container image of the Inference Server, release 18.07, is available as a beta release.

Contents of the Inference Server

This container image contains the Inference Server executable in `/opt/inference_server`.

The container also includes the following:

- Ubuntu 16.04

> Container image `18.07-py2` contains Python 2.7; `18.07-py3` contains Python 3.5.

- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.425
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.4
- NCCL 2.2.13 (optimized for NVLink™)
- TensorRT 4.0.1

Driver Requirements

Release 18.07 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The Inference Server container image version 18.07 is based on NVIDIA Inference Server 0.4.0 beta, TensorFlow 1.8.0, and Caffe2 0.8.1.
- Latest version of CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.425.
- Support added for TensorFlow SavedModel format.

- Support added for gRPC in addition to existing HTTP REST.
- Ubuntu 16.04 with June 2018 updates

Known Issues

This is a beta release of the Inference Server. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.

Chapter 86. Inference Server Release 18.06 Beta

The NVIDIA container image of the Inference Server, release 18.06, is available as a beta release.

Contents of the Inference Server

This container image contains the Inference Server executable in `/opt/inference_server`.

The container also includes the following:

- Ubuntu 16.04

> Container image `18.06-py2` contains Python 2.7; `18.06-py3` contains Python 3.5.

- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.333 (see section 2.3.1)
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.4
- NCCL 2.2.13 (optimized for NVLink™)
- TensorRT 4.0.1

Driver Requirements

Release 18.06 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The Inference Server container image version 18.06 is based on NVIDIA Inference Server 0.3.0 beta, TensorFlow 1.8.0, and Caffe2 0.8.1.
- Support added for Caffe2 NetDef models.

- Support added for CPU-only servers in addition to servers that have one or more GPUs. The Inference Server can simultaneously use both CPUs and GPUs for inferencing.
- Logging format and control is unified across all inferencing backends: TensorFlow, TensorRT, and Caffe2.
- Gracefully exits upon receiving SIGTERM or SIGINT. Any in-flight inferences are allowed to complete before exiting, subject to a timeout.
- Server status is enhanced to report the readiness and availability of the server and of each model (and model version).
- Ubuntu 16.04 with May 2018 updates

Known Issues

This is a beta release of the Inference Server. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.

Chapter 87. Inference Server Release 18.05 Beta

The NVIDIA container image of the Inference Server, release 18.05, is available as a beta release.

Contents of the Inference Server

This container image contains the Inference Server executable in `/opt/inference_server`.

The container also includes the following:

- Ubuntu 16.04

> Container image `18.05-py2` contains Python 2.7; `18.05-py3` contains Python 3.5.

- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.333 (see section 2.3.1)
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.2
- NCCL 2.1.15 (optimized for NVLink™)
- TensorRT 3.0.4

Driver Requirements

Release 18.05 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- The Inference Server container image version 18.05 is based on NVIDIA Inference Server 0.2.0 beta and TensorFlow 1.7.0.
- Multiple model support. The Inference Server can manage any number and mix of TensorFlow to TensorRT models (limited by system disk and memory resources).

- TensorFlow to TensorRT integrated model support. The Inference Server can manage TensorFlow models that have been optimized with TensorRT.
- Multi-GPU support. The Inference Server can distribute inferencing across all system GPUs. Systems with heterogeneous GPUs are supported.
- Multi-tenancy support. Multiple models (or multiple instances of the same model) can run simultaneously on the same GPU.
- Batching support
- Ubuntu 16.04 with April 2018 updates

Known Issues

This is a beta release of the Inference Server. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.

Chapter 88. Inference Server Release 18.04 Beta

The NVIDIA container image of the Inference Server, release 18.04, is available as a beta release.

Contents of the Inference Server

This container image contains the Inference Server executable in `/opt/inference_server`.

The container also includes the following:

- Ubuntu 16.04 including Python 2.7 environment
- NVIDIA CUDA 9.0.176 (see Errata section and 2.1) including CUDA® Basic Linear Algebra Subroutines library™ (cuBLAS) 9.0.333 (see section 2.3.1)
- NVIDIA CUDA® Deep Neural Network library™ (cuDNN) 7.1.1
- NCCL 2.1.15 (optimized for NVLink™)

Driver Requirements

Release 18.04 is based on CUDA 9, which requires NVIDIA Driver release 384.xx.

Key Features and Enhancements

This Inference Server release includes the following key features and enhancements.

- This is the beta release of the Inference Server container.
- The Inference Server container image version 18.04 is based on NVIDIA Inference Server 0.1.0 beta.
- Multiple model support. The Inference Server can manage any number and mix of models (limited by system disk and memory resources). Supports TensorRT and TensorFlow GraphDef model formats.
- Multi-GPU support. The server can distribute inferencing across all system GPUs.
- Multi-tenancy support. Multiple models (or multiple instances of the same model) can run simultaneously on the same GPU.

- Batching support.
- Latest version of NCCL 2.1.15
- Ubuntu 16.04 with March 2018 updates

Known Issues

This is a beta release of the Inference Server. All features are expected to be available, however, some aspects of functionality and performance will likely be limited compared to a non-beta release.

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

Trademarks

NVIDIA, the NVIDIA logo, and cuBLAS, CUDA, CUDA Toolkit, cuDNN, DALI, DIGITS, DGX, DGX-1, DGX-2, DGX Station, DLProf, GPU, Jetson, Kepler, Maxwell, NCCL, Nsight Compute, Nsight Systems, NVCaffe, NVIDIA Ampere GPU architecture, NVIDIA Deep Learning SDK, NVIDIA Developer Program, NVIDIA GPU Cloud, NVLink, NVSHMEM, PerfWorks, Pascal, SDK Manager, Tegra, TensorRT, Tesla, TF-TRT, Triton Inference Server, Turing, and Volta are trademarks and/or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright