# NVIDIA DGX BasePOD: Run:ai

## Deployment Guide

Featuring NVIDIA DGX A100 Systems

DG-12300-001 V3

2023-12-13

# Contents

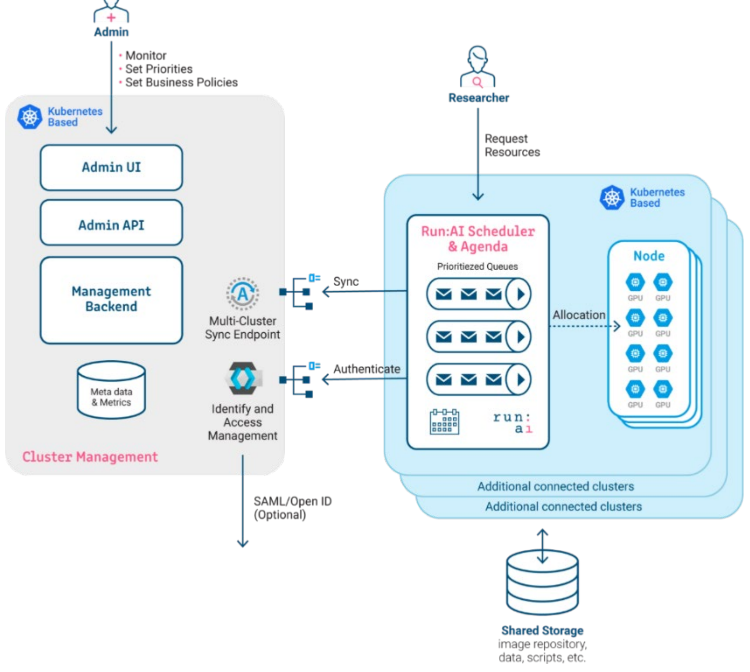
# Chapter 1 Introducing Run:ai

This document describes how to deploy the Run:ai Atlas Platform on NVIDIA DGX BasePOD™ configurations through a simplified deployment using NVIDIA Base Command™ Manager (BCM) software.

The Run:ai Atlas platform enables IT organizations to architect their AI environment with cloud-like resource accessibility and management on any infrastructure. It also enables researchers to use any machine learning (ML) and data science tools they choose. The platform builds off powerful distributed computing and scheduling concepts from High Performance Computing (HPC) and is implemented as a Kubernetes (K8s) plug-in. It speeds up data science workflows and creates visibility for IT teams who can now manage valuable resources more efficiently and ultimately reduce idle GPU time.

The platform consists of the Run:ai cluster (Figure 1, right) and the Run:ai control plane or backend (Figure 1, left). For installation on BCM, the platform hosts the control plane component, and the customer deploys the cluster components onto the K8s cluster or clusters created by BCM.

**Figure 1. Run:ai architecture**



The Run:ai scheduler extends the capabilities of the default K8s scheduler without replacing it. The Run:ai scheduler uses business rules based on project quotas to schedule workloads sent by researchers and data scientists. Fractional GPU, a Run:ai technology, enables researchers to allocate subsets of a GPU rather than the whole GPU, enhancing the underlying resource utilization. Additionally, the Run:ai agent and other monitoring tools are responsible for sending monitoring data to the Run:ai control plane.

Through Run:ai, researchers and data scientists have various methods for submitting and interacting with their workloads. They can submit ML workloads using the Run:ai CLI, directly by sending YAML files to K8s, through the K8s or Run:ai API, or through the Run:ai researcher user interface (UI). The researcher UI enables the simple submission of workloads based on templates that prefill necessary fields for workload submission. This enables administrators to provide a streamlined process for onboarding new users to the platform. Templates delivered through the Run:ai web UI enable a streamlined approach to deploying Jupyter notebooks while the UI also enables researchers and data scientists to connect directly to their notebook after it is running.

The Run:ai Atlas platform gives IT the control and visibility they need into their AI and ML environment while also abstracting away all the complex underlying infrastructure, enabling researchers and data scientists to leverage the tools of their choosing to drive innovation within the enterprise.

# Chapter 2 Deploying Run:ai

This section details the preferred method for installing Run:ai on DGX BasePOD configurations.

## 2.1 Prerequisites

The deployment procedure in this guide relies on having BCM with K8s already installed on a DGX BasePOD configuration. If needed, see the NVIDIA DGX BasePOD Deployment Guide for information on how to perform the installation.

Ensure that the tenant name and application secret key have been provided by the Run:ai support team before starting the installation. The tenant name is the dedicated control plane URL for accessing the Run:ai Atlas platform and the application secret key is an API key required to securely communicate with the platform.

In addition, the Cluster URL, certificate (in `.crt` format), and private key (in a `key` file) must be generated by the IT department of customer before installation begins.

**Note:** Run:ai will not function properly with self-signed certificates.

The Cluster URL corresponds to a DNS A record that is created and maintained by the enterprise DNS server of the customer. The hostname for the DNS A record should be unique and resolve to one of the nodes within the BCM K8s cluster.

The Cluster URL must be appended with the port that the NGINX ingress service is listening on. Run the following command in the CLI to determine the `ingress-nginx` port mapping:

```
root@basepod-head1:~# kubectl get svc -n ingress-nginx
NAME                            TYPE          CLUSTER-IP     EXTERNAL-IP    PORT(S) AGE
ingress-nginx-controller        NodePort      10.150.30.30   10.130.122.9   80:30080/TCP,
443:3044 3/TCP 12d
ingress-nginx-controller-admission   ClusterIP     10.150.166.37  <none>  443/TCP 12d
```

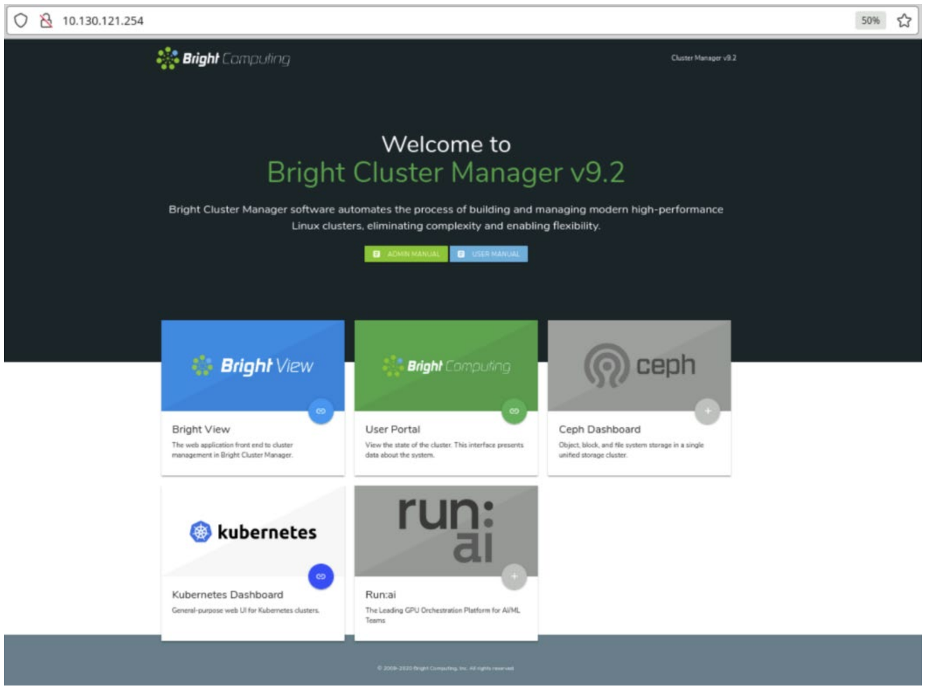
In this example, the port mapping for `443` is `30453`.

Additionally, the certificate CN must be matched to the DNS record of the Cluster URL to secure all the inbound traffic to the cluster.

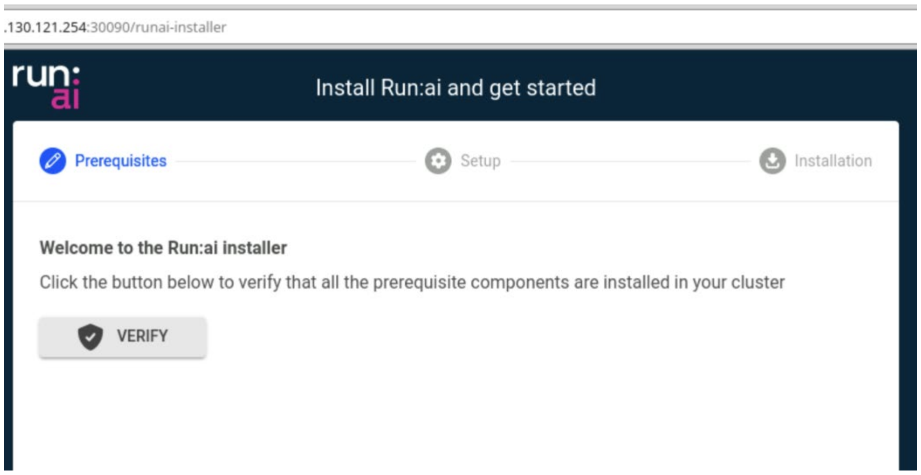
Other prerequisites needed for the Run:ai installation are at:
https://docs.run.ai/latest/admin/runai-setup/cluster-setup/cluster-prerequisites/

## 2.2 Installing Run:ai

1. Select the Run:ai tile on the locally installed BCM landing page.


10.130.121.254
Bright Computing
Cluster Manager v9.2
Welcome to
Bright Cluster Manager v9.2
Bright Cluster Manager software automates the process of building and managing modern high-performance Linux clusters, eliminating complexity and enabling flexibility.
ADMIN MANUAL
USER MANUAL
Bright View
The web application front end to cluster management in Bright Cluster Manager.
User Portal
View the state of the cluster. This interface presents data about the system.
Ceph Dashboard
Object, block, and file system storage in a single unified storage cluster.
Kubernetes Dashboard
General-purpose web UI for Kubernetes clusters.
Run:ai
The Leading GPU Orchestration Platform for AI/ML Teams

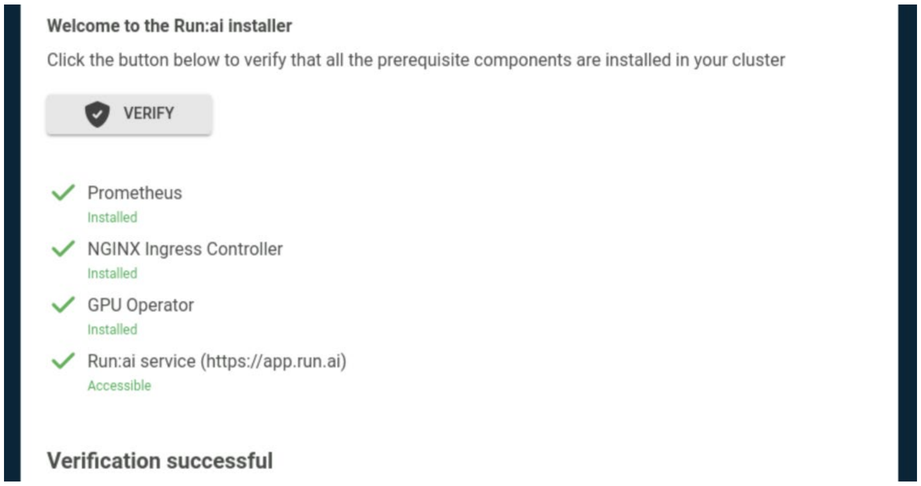
2. Select `Verify` on the resulting the `Prerequisites` window.


.130.121.254:30090/runai-installer
run:ai
Install Run:ai and get started
Prerequisites
Setup
Installation
VERIFY

**Welcome to the Run:ai installer**

Click the button below to verify that all the prerequisite components are installed in your cluster

3. After four green check marks are displayed, select `CONTINUE` to proceed with the Run:ai installation.


Welcome to the Run:ai installer

Click the button below to verify that all the prerequisite components are installed in your cluster

✓ Prometheus
Installed

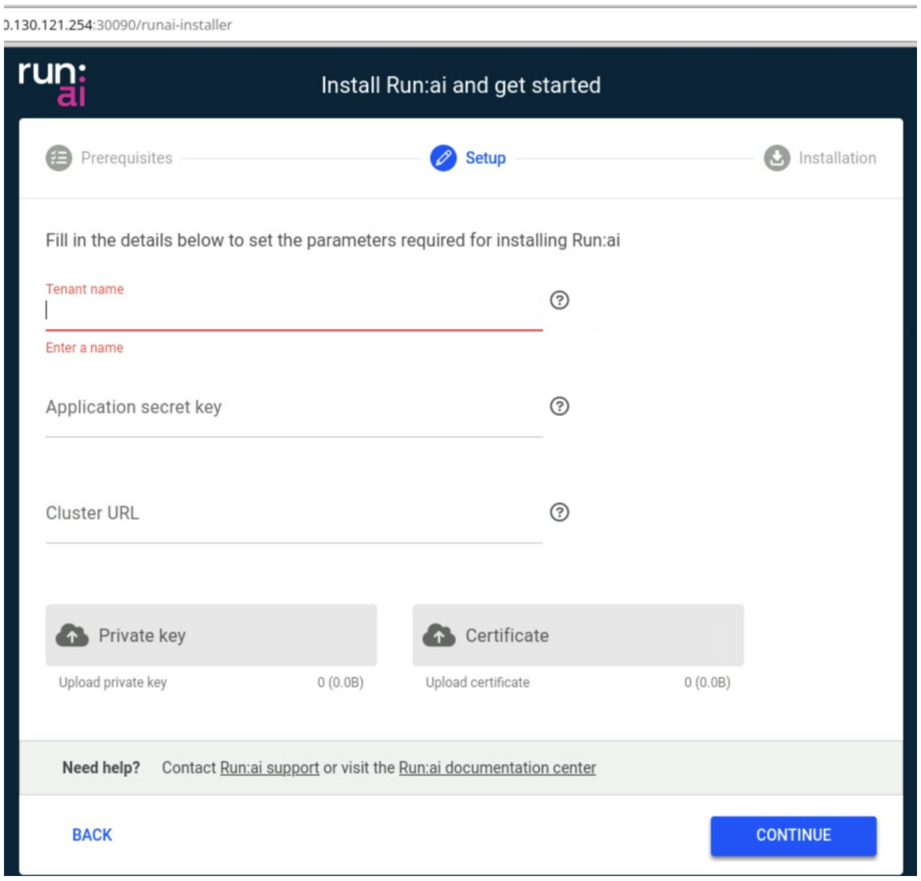
✓ NGINX Ingress Controller
Installed

✓ GPU Operator
Installed

✓ Run:ai service (https://app.run.ai)
Accessible

Verification successful

4. Enter the information collected in Section 2.1 and then select `CONTINUE`.


).130.121.254:30090/runai-installer

Install Run:ai and get started

Fill in the details below to set the parameters required for installing Run:ai

Tenant name

Enter a name

Application secret key

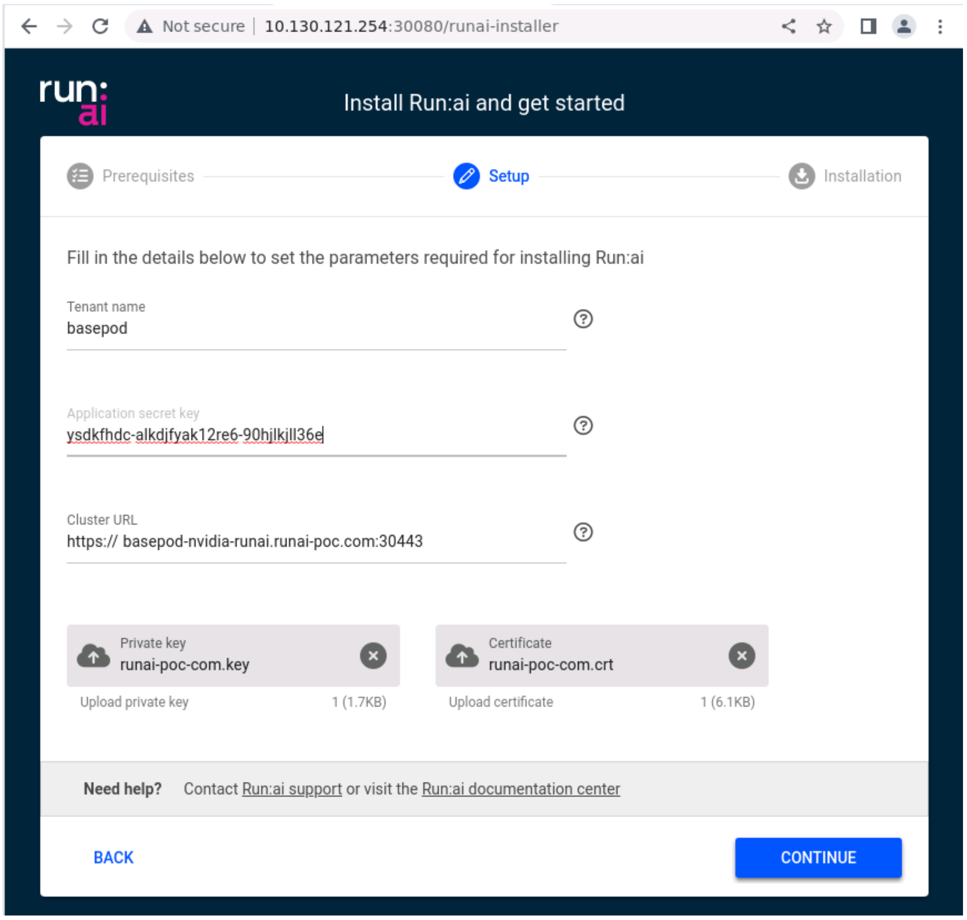
Cluster URL

Upload private key 0 (0.0B)

Upload certificate 0 (0.0B)

Need help? Contact Run:ai support or visit the Run:ai documentation center

The screen will then be replaced with one like the following.


Not secure | 10.130.121.254:30080/runai-installer
Install Run:ai and get started
Prerequisites
Setup
Installation
BACK
CONTINUE

Fill in the details below to set the parameters required for installing Run:ai

Tenant name
basepod

Application secret key
ysdkfhdc-alkdjfyak12re6-90hjlkjll36e

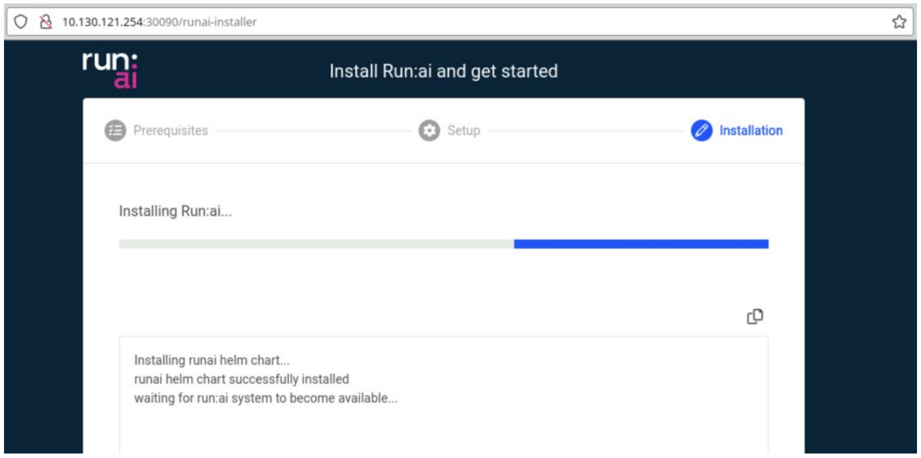
Cluster URL
https:// basepod-nvidia-runai.runai-poc.com:30443

Private key
runai-poc-com.key
Upload private key 1 (1.7KB)

Certificate
runai-poc-com.crt
Upload certificate 1 (6.1KB)

Need help? Contact Run:ai support or visit the Run:ai documentation center

5. View the progress of the installation as components are deployed to the BCM K8s cluster.

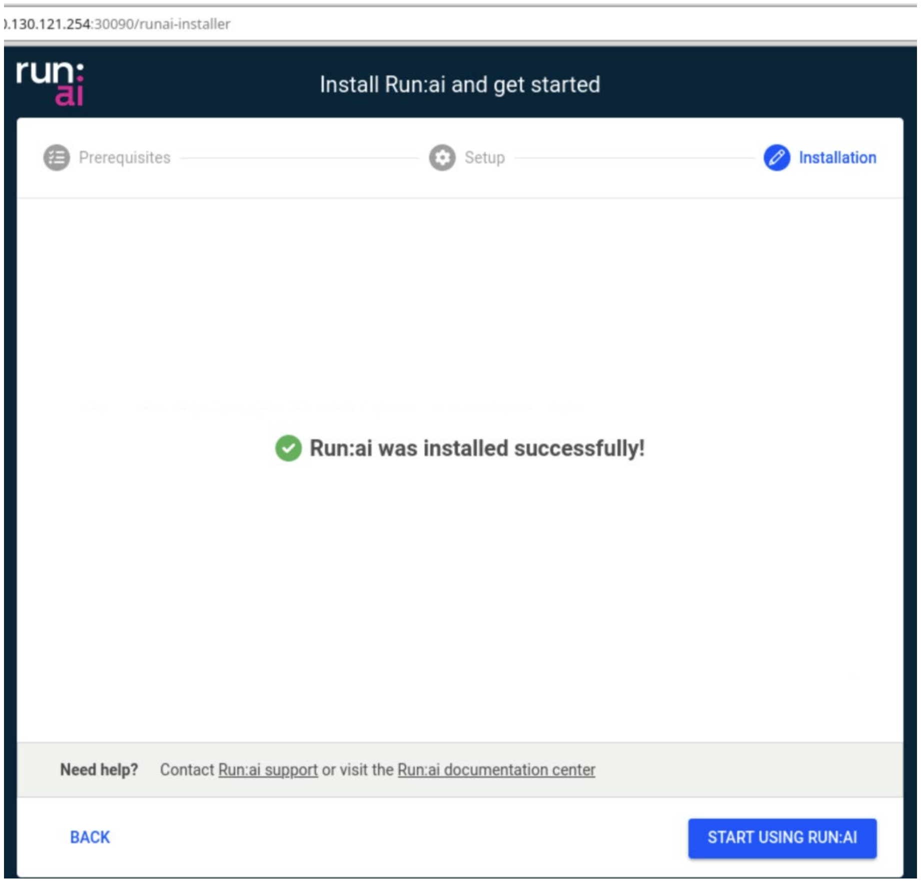

10.130.121.254:30090/runai-installer
Install Run:ai and get started
Prerequisites
Setup
Installation

Installing Run:ai...

```
Installing runai helm chart...
runai helm chart successfully installed
waiting for run:ai system to become available...
```

The Run:ai installation should complete within a few minutes, marking the occasion with a message of `Run:ai was installed successfully!`

6. Select `START USING RUN:AI` to launch the login page of the tenant in a new browser tab.

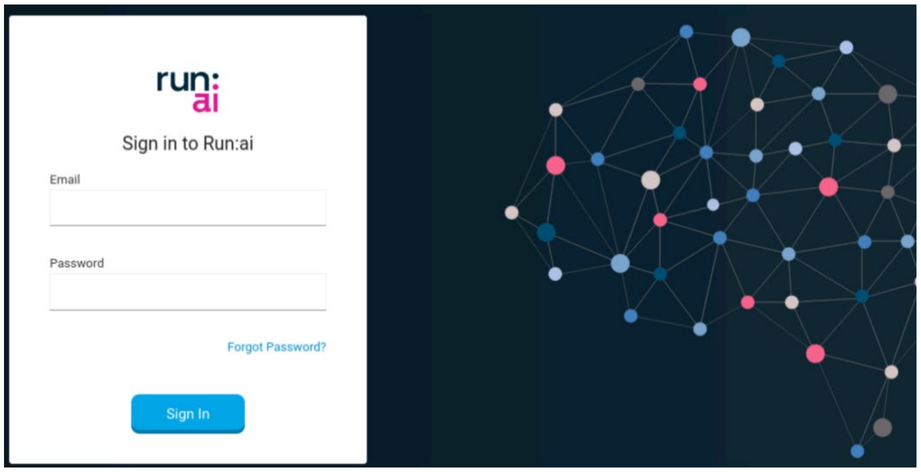

).130.121.254:30090/runai-installer
run:ai
Install Run:ai and get started
Prerequisites
Setup
Installation
BACK
START USING RUN:AI

Run:ai was installed successfully!

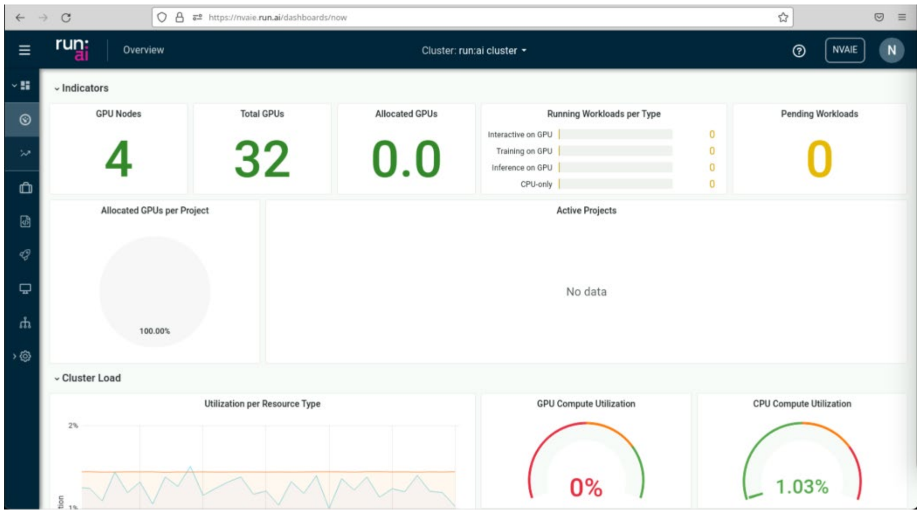
Need help? Contact Run:ai support or visit the Run:ai documentation center

7. Login to the assigned tenant with the email and password provided by the Run:ai support team.

   Upon first login, a password change will be required.



8. After logging in, the `Overview` window is display.



The run.ai installation from BCM landing page is now complete.

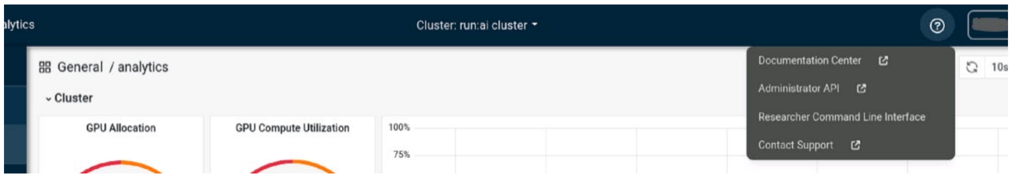
The next section covers using the CLI to finalize the configuration.

## 2.3 Installing the Run:ai CLI

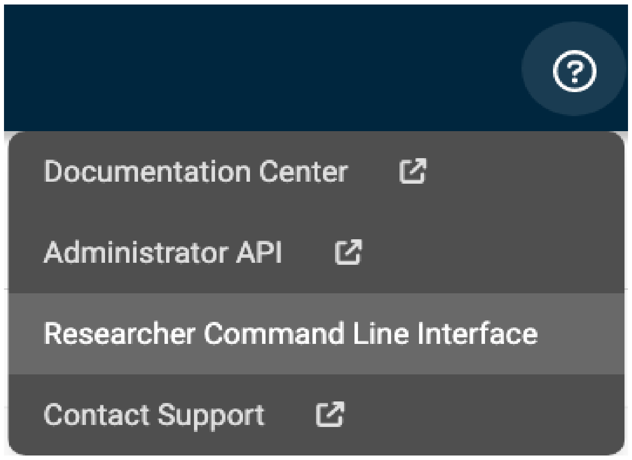
Administrators and researchers can use the Run:ai CLI to perform tasks ranging from configuring and managing the clusters to submitting, listing, and interactive with jobs.

Perform following steps to download Run:ai CLI.

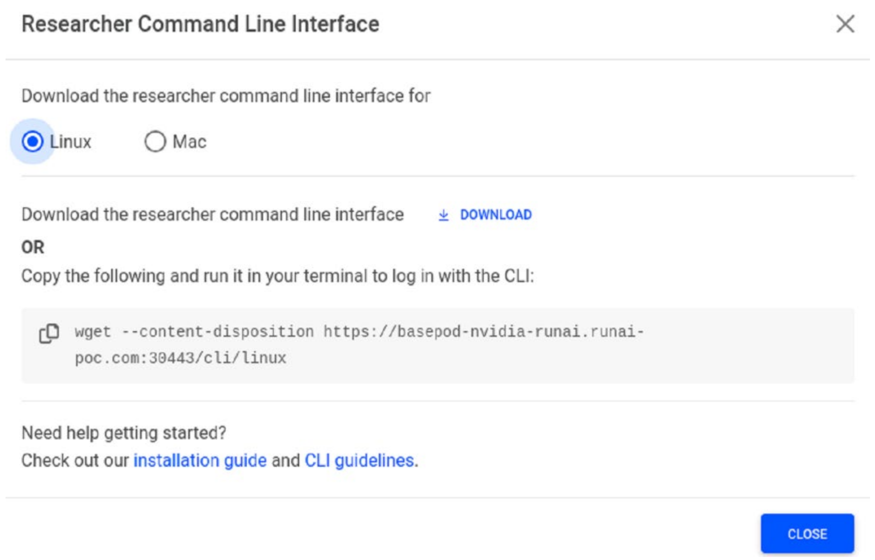
1. Select the `?` on the right side of the navigation bar.



2. Select `Researcher Command Line Interface`.



3. Specify `Linux` on the resulting screen and then copy the highlighted `wget` code.



Researcher Command Line Interface

Download the researcher command line interface for

[✓ Linux] Mac

Download the researcher command line interface DOWNLOAD

OR

Copy the following and run it in your terminal to log in with the CLI:

```
wget --content-disposition https://basepod-nvidia-runai.runai-
poc.com:30443/cli/linux
```

Need help getting started?
Check out our installation guide and CLI guidelines.

4. `ssh` into the head node and paste in the copied `wget` code to download the CLI binary.

```
admin@basepod-head1:~# wget --content-disposition https://basepod-nvidia-runai.runai-
poc.com/cli/linux
--2023-01-20 12:51:44--  https://basepod-nvidia-runai.runai-poc.com/cli/linux
Resolving basepod-nvidia-runai.runai-poc.com (basepod-nvidia-runai.runai-poc.com)...
10.130.122.9
Connecting to basepod-nvidia-runai.runai-poc.com (basepod-nvidia-runai.runai-
poc.com)|10.130.122.9|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/octet-stream]
Saving to: 'runai'

runai                     [           <=>         ]  61.15M  30.2MB/s    in 2.0s

2023-01-20 12:51:46 (30.2 MB/s) - 'runai' saved [64118773]
```

5. Change the permission of the binary to execute.

```
admin@basepod-head1:~# chmod +x runai
```

6. Move the binary to `/usr/local/bin`.

```
admin@basepod-head1:~# mv runai /usr/local/bin/runai
```

The Run:ai CLI is now ready to be executed.

## 2.4 Setting Up Research Access Control

See the Setup Researcher Access Control

section of the Run:ai Documentation Library for information on how setup access control for researchers.

**Note**: Follow the instructions under the Bright tab of the (Mandatory) Kubernetes Configuration heading.

# Chapter 3 Verifying Run:ai

This section runs contains some procedures to verify that Run:ai is functioning correctly.

For information on how to use Run:ai see:

> Run:ai guides: https://www.run.ai/guides

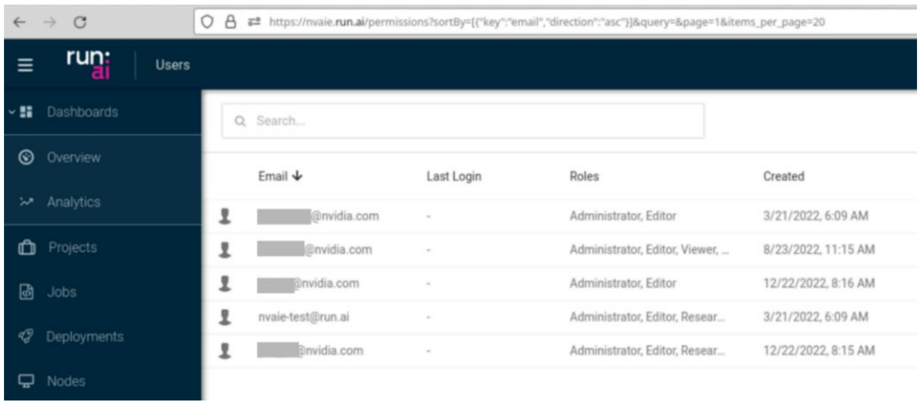
> Run:ai videos and webinars: https://www.run.ai/resources#Videos-Webinars

> NVIDIA Run:ai developer page: https://developer.nvidia.com/run-ai

**Note:** In addition to creating users through the Run:ai UI, existing SSO logins can be linked to Run:ai so that users and groups are mapped to roles within the Run:ai platform. Additional information on the Run:ai SSO integration is available at the following link: https://docs.run.ai/latest/admin/runai-setup/authentication/sso/?h=sso

## 3.1 Create a User and a Project

1. Goto the `Settings and Users` window in the Run:ai UI.


https://nvaie.run.ai/permissions?sortBy=[{"key":"email","direction":"asc"}]&query=&page=1&items_per_page=20
run:ai
Users
Dashboards
Overview
Analytics
Projects
Jobs
Deployments
Nodes
Search...

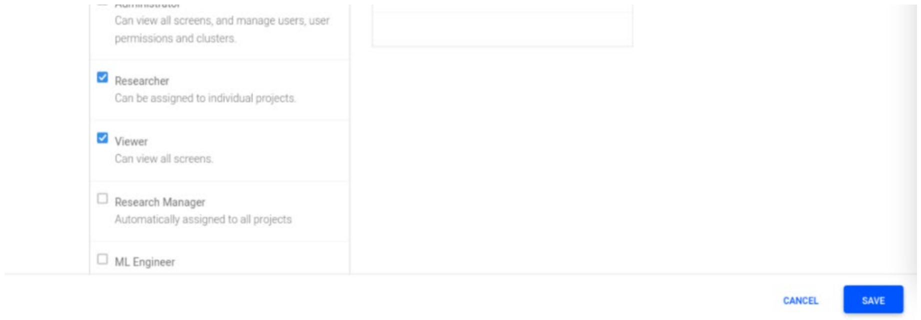
| Email ↓ | Last Login | Roles | Created |
|---|---|---|---|
| [redacted]@nvidia.com | - | Administrator, Editor | 3/21/2022, 6:09 AM |
| [redacted]@nvidia.com | - | Administrator, Editor, Viewer, ... | 8/23/2022, 11:15 AM |
| [redacted]@nvidia.com | - | Administrator, Editor | 12/22/2022, 8:16 AM |
| nvaie-test@run.ai | - | Administrator, Editor, Resear... | 3/21/2022, 6:09 AM |
| [redacted]@nvidia.com | - | Administrator, Editor, Resear... | 12/22/2022, 8:15 AM |

2. On the top right of the page, select `+ New User`.
3. Enter the email address of users and assign them the roles of Researcher and Viewer by selecting the corresponding check boxes.

   For this example, the researcher will use the email of researcher1@nvidia.com.

Information about the various roles within Run:ai is available in the Run:ai Administrator documentation.

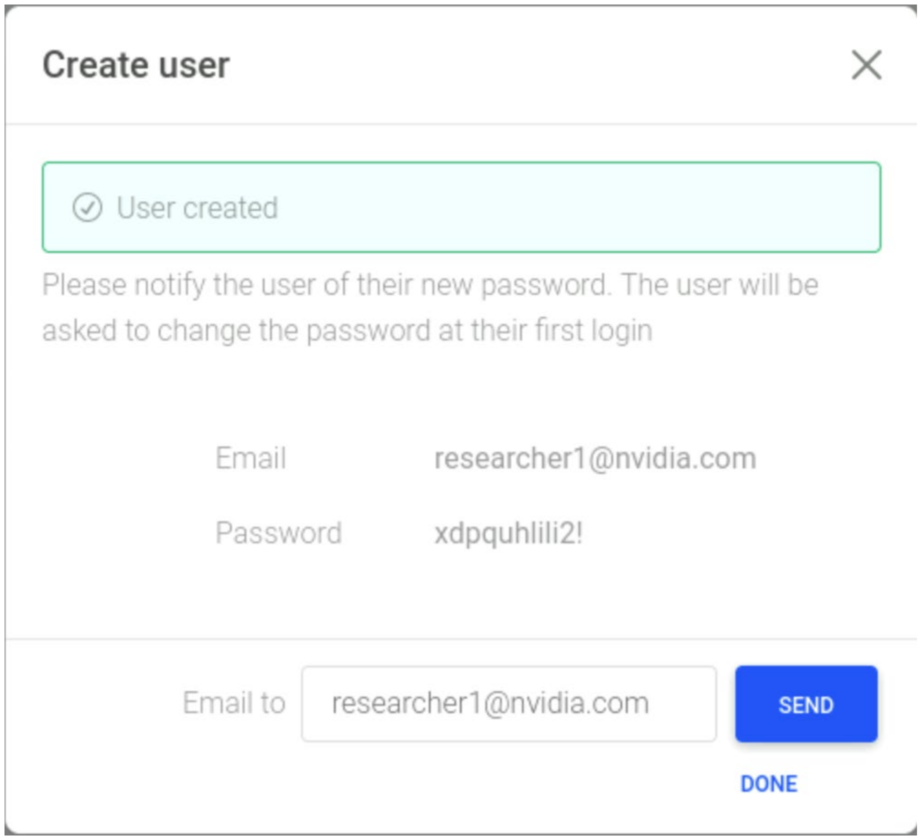
4. Select the `Save` button to complete user creation.



Can view all screens, and manage users, user permissions and clusters.

- [✓] Researcher
  Can be assigned to individual projects.
- [✓] Viewer
  Can view all screens.
- [ ] Research Manager
  Automatically assigned to all projects
- [ ] ML Engineer

5. After `User created` is displayed, make note of temporary password and provide it to the researcher or leverage the `Email to` field if an SMTP server configured.



Create user

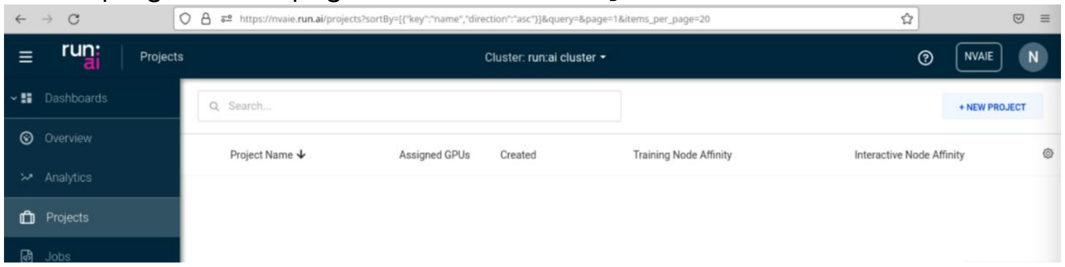
User created

Please notify the user of their new password. The user will be asked to change the password at their first login

| | |
|---|---|
| Email | researcher1@nvidia.com |
| Password | xdpquhlili2! |

Email to: researcher1@nvidia.com

6. Goto the Dashboard and select the drop-down for `Projects`.
7. On the top right of the page, select `+ New Project`.

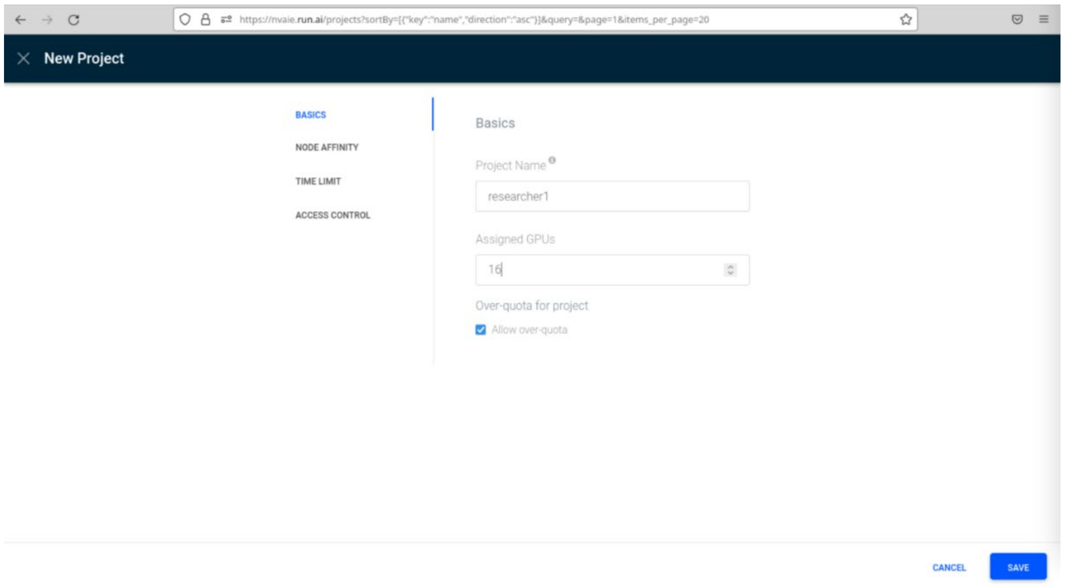


https://nvaie.run.ai/projects?sortBy=[{"key":"name","direction":"asc"}]&query=&page=1&items_per_page=20

Cluster: run:ai cluster

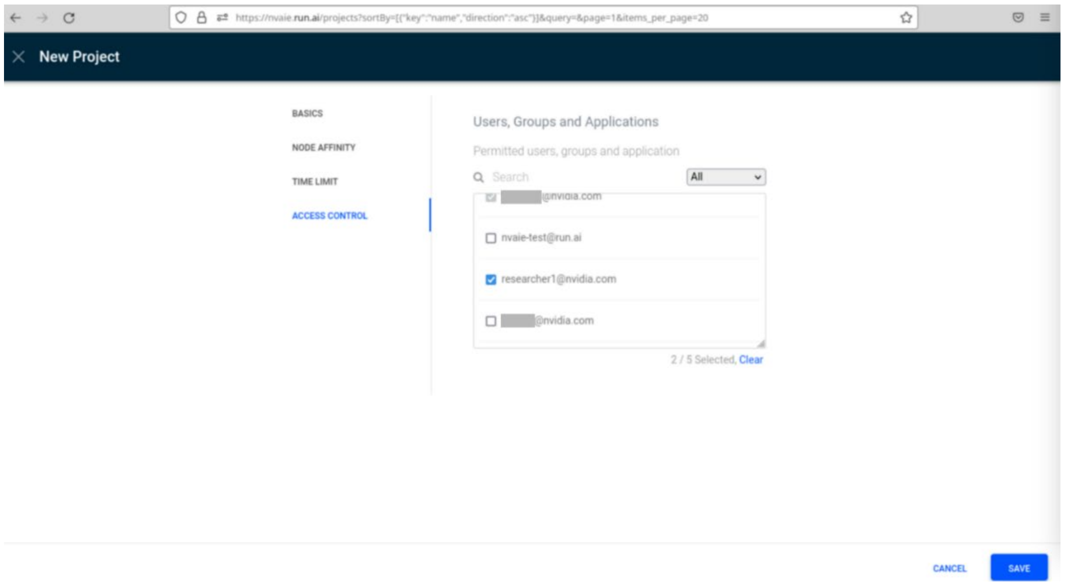
| Project Name | Assigned GPUs | Created | Training Node Affinity | Interactive Node Affinity |
|---|---|---|---|---|

8. Enter the `Project Name` and the number of `Assigned GPUs`.



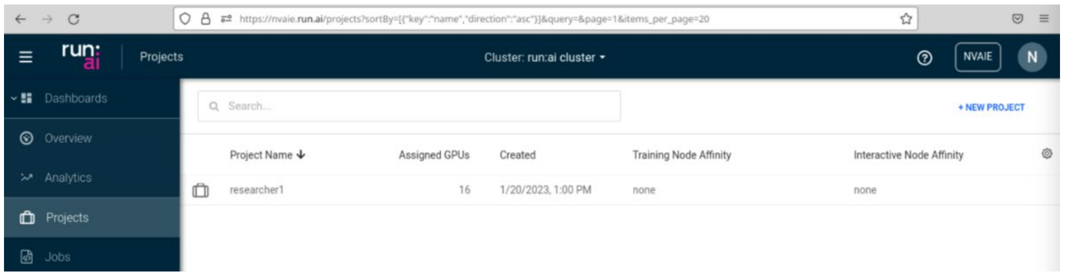
Project Name: researcher1
Assigned GPUs: 16
Over-quota for project
☑ Allow over-quota

9. Choose the `NODE AFFINITY` and `TIME LIMIT` for the Project.
10. Select who can access the project through the `ACCESS CONTROL` screen and then select `Save`.



[redacted]@nvidia.com
☐ nvaie-test@run.ai
☑ researcher1@nvidia.com
☐ [redacted]@nvidia.com
2 / 5 Selected, Clear

11. Ensure that the newly created Project is displayed on the `Projects` screen.

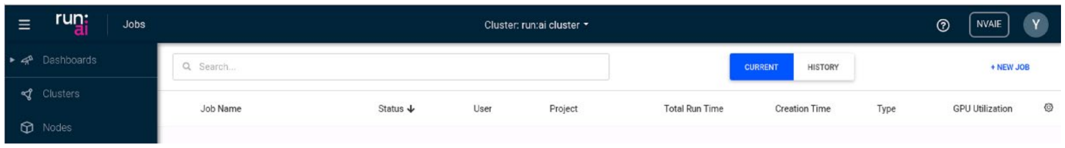


| Project Name | Assigned GPUs | Created | Training Node Affinity | Interactive Node Affinity |
|---|---|---|---|---|
| researcher1 | 16 | 1/20/2023, 1:00 PM | none | none |

## 3.2 Run and Monitor a Job (Run:ai UI)

The NVIDIA NGC TensorFlow Container is optimized for GPU acceleration and contains a validated set of libraries that enable and optimize GPU performance. In this example, the Run:ai UI is used to submit an unattended ResNet-50 training job with NGC TensorFlow container.

1. Goto the Dashboard and select the drop-down for `Jobs`.
2. Select `+ NEW JOB` on the top right of the page.

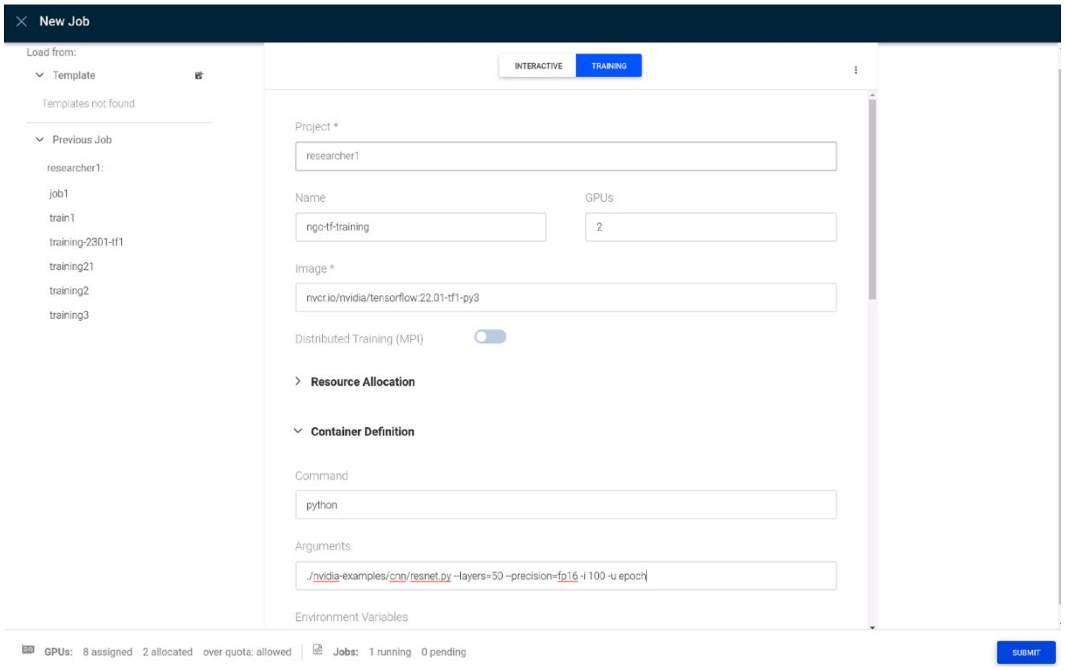


| Job Name | Status | User | Project | Total Run Time | Creation Time | Type | GPU Utilization |
|---|---|---|---|---|---|---|---|

3. In the `New Job` screen, enter the required information, such as the project name, job name, number of GPUs, image name, and commands. Then, select `SUBMIT`.

   The following image and command arguments were used to launch this training job:

   - Image: `nvcr.io/nvidia/tensorflow:22.01-tf1-py3`
   - Arguments: `./nvidia-examples/cnn/resnet.py --layers=50 --precision=fp16 -i 100 -u epoch`



Project *: researcher1

Name: ngc-tf-training

GPUs: 2

Image *: nvcr.io/nvidia/tensorflow:22.01-tf1-py3

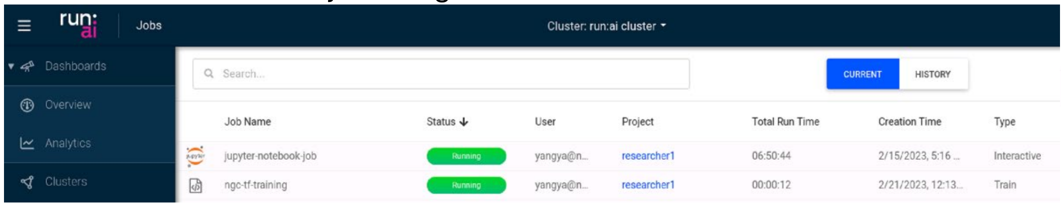
Distributed Training (MPI)

Command: python

Arguments: ./nvidia-examples/cnn/resnet.py --layers=50 --precision=fp16 -i 100 -u epoch

GPUs: 8 assigned 2 allocated over quota: allowed | Jobs: 1 running 0 pending

4. Monitor the status of the job using the `Jobs` screen.



| Job Name | Status | User | Project | Total Run Time | Creation Time | Type |
|---|---|---|---|---|---|---|
| jupyter-notebook-job | Running | yangya@n... | researcher1 | 06:50:44 | 2/15/2023, 5:16 ... | Interactive |
| ngc-tf-training | Running | yangya@n... | researcher1 | 00:00:12 | 2/21/2023, 12:13... | Train |

5. The status should become `Succeeded` when the job completes.

## 3.3 Run and Monitor a Job (Run:ai CLI)

This section shows how to submit and check the status of a job using the Run:ai CLI.

1. Login to `runai` using the `login` option

   Enter the Username and Password.

```
researcher1@basepod-head1:~/runai_dir$ runai login
Username: researcher1@nvidia.com
Password:
INFO[0022] Logged in successfully
```

2. Submit a distributed training job.

```
researcher1@basepod-head1:~/runai_dir$ runai submit-mpi dist-job1 -i \n
gcr.io/run-ai-demo/quickstart-distributed:tf-2.1.0 -g 4 -p researcher1
Job dist-job1 submitted successfully.
```

3. Check the job status using `runai describe`.

```
researcher1@basepod-head1:~/runai_dir$ runai describe job dist-job1 -p
researcher1
Name: dist-job1
Namespace: runai-researcher1
Type: Train
Status: Running
Duration: 5m0
GPUs: 0
Total Requested GPUs: 0
Allocated GPUs: 2
Allocated GPUs memory: 0
Running PODs: 3
Pending PODs: 0
Parallelism: 1
Completions: 1
Succeeded PODs: 0
Failed PODs: 0
Is Distributed Workload: true
Service URLs:
Command Line: runai submit-mpi dist-job1 --processes=2 -g 1 -i gcr.io/run-ai-
demo/quickstart-distributed:tf-2.1.0 -e RUNAI_SLEEP_SECS=60
```

## 3.4 Enable Direct Access to K8s

This section details how to enable direct access to K8s for a researcher.

1. Goto the `.kube` directory for the researcher.

   ```
   researcher1@basepod-head1:~$ cd .kube
   ```

2. Copy the default config file.

   ```
   researcher1@basepod-head1:~/.kube$ cp config-default config-researcher
   ```

3. Edit `config-researcher` file.

   Using the following code as an example, with these exceptions:

   - Keep the `certificate-authority-data` the same as `config-default`.
   - Replace `<REALM-NAME>` with the realm that is on the General Settings page of the Run:ai UI.

   ```
   apiVersion: v1
   clusters:
   - cluster:
       certificate-authority-data:
       server: https://localhost:10443
     name: default
   contexts:
   - context:
       cluster: default
       namespace: runai-researcher1
       user: runai-authenticated-user
     name: default
   current-context: default
   kind: Config
   preferences: {}
   users:
   - name: runai-authenticated-user
     user:
       auth-provider:
         config:
           airgapped: "true"
           auth-flow: cli
           client-id: runai-cli
           idp-issuer-url: https://app.run.ai/auth/realms/<REALM-NAME>
           realm: <REALM-NAME>
         name: oidc
   ```

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051

**http://www.nvidia.com**