



Running NVIDIA Parabricks on AWS

Table of contents

What is NVIDIA Parabricks?

Starting an EC2 Instance

Installing Parabricks

Testing Parabricks

Private Workflows

Closing Remarks

List of Figures

Figure 0. Image Instances

Figure 1. Image Launch Instances

Figure 2. Image Name And Tags

Figure 3. Image App And Os Images

Figure 4. Image Instance Type

Figure 5. Image Key Pair

Figure 6. Image Configure Storage

Figure 7. Image Summary

Figure 8. Image Instances Info

Figure 9. Image Connect To Instance

Figure 10. Image Install Parabricks

Figure 11. Image Parabricks On Ngc

Figure 12. Image Docker Pull Cmd

Figure 13. Image Tree Cmd

Figure 14. Image Banner

Figure 15. Image Ready2run

Figure 16. Image Create Workflow Run

Figure 17. Image Run Details

Figure 18. Image Runs

Figure 19. Image Result

This guide shows how to run Parabricks on [AWS HealthOmics](#) and is divided into two parts:

The [first part](#) shows how to run Parabricks workflows using an EC2 instance. For this method, we will spin up a machine instance on AWS, pull the Parabricks container directly from NVIDIA, and run an example dataset. This option allows for the most flexibility in terms of Parabricks functionality and for easily integrating into larger pipelines and other AWS services.

The [second part](#) shows how to run Parabricks workflows using [Amazon HealthOmics](#). This is Amazon's platform for bioinformatics research and allows you to store data, run analysis pipelines, and look at the results, all in one place. There are two ways to use HealthOmics. [Ready2Run workflows](#) are pre-made analysis pipelines where users can click on a pipeline they want to run, click on the data they want to use, and click run all without ever leaving the console GUI. Power users can also use the AWS CLI to start these jobs. The other way to run HealthOmics is through Private Workflows. These are great for if you want a little more control over the workflows and want to make edits to fit your needs exactly.

What is NVIDIA Parabricks?

Parabricks is an accelerated compute framework that supports applications across the genomics industry, primarily supporting analytical workflows for DNA, RNA, and somatic mutation detection applications. With industry leading compute times, Parabricks rapidly converts a FASTQ file to a VCF using multiple, industry validated variant callers and also includes the ability to QC and annotate those variants. As Parabricks is based upon publicly available tools, results are easy to verify and combine with other publicly available data sets.

More information is available on the [Parabricks Product Page](#).

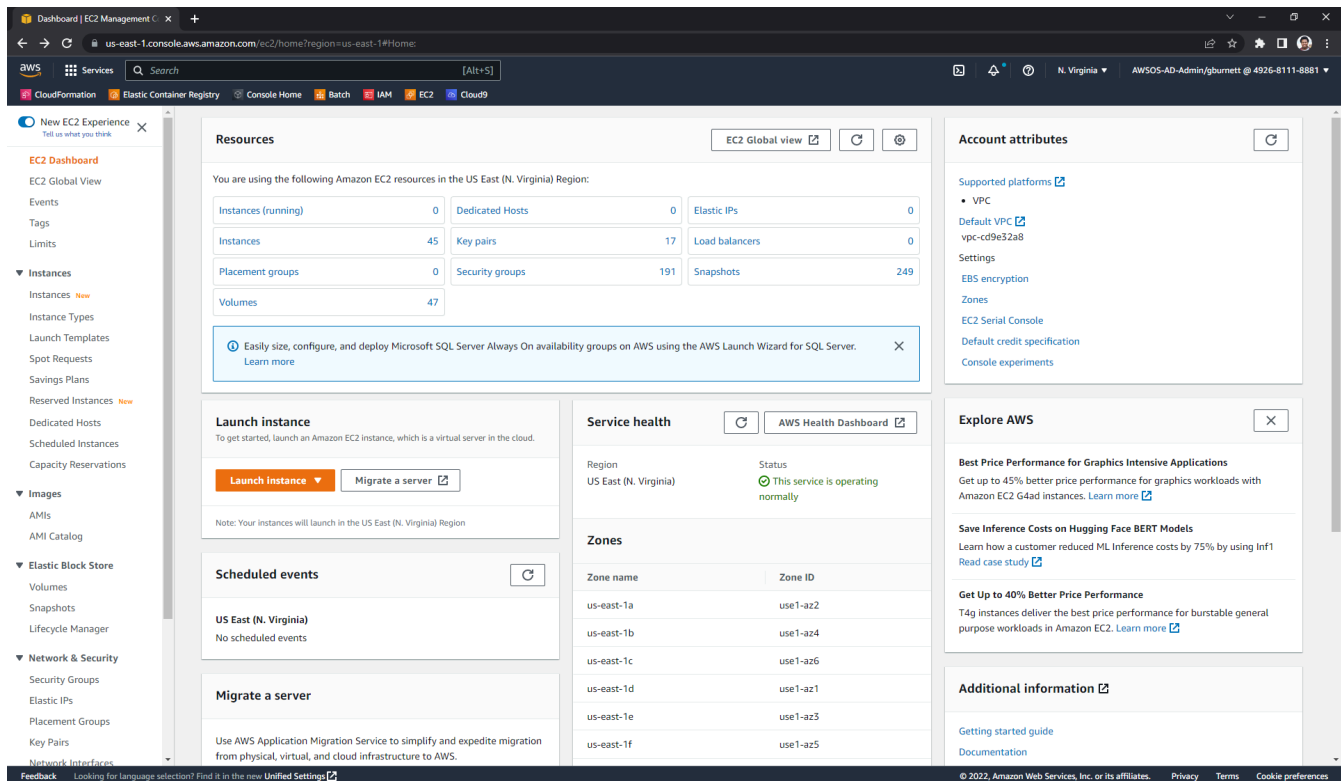
Detailed installation, usage, and tuning information is available in the [Parabricks user guide](#).

Running Workflows on an EC2 Instance

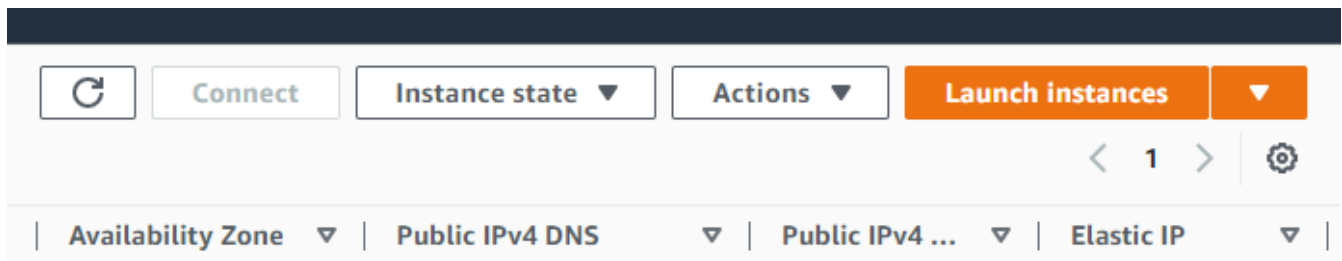
Starting an EC2 Instance

In this section, we will show how to start an EC2 instance on AWS.

Begin by navigating to the EC2 console on AWS. The page should look something like this:



In the left sidebar under “Instances” click “Instances”. Here we can see all the instances we have created. Let’s create a new one where we will install Parabricks, by clicking “Launch instances” in the top right.



In this guide, we will name our instance “Parabricks” but it can be named anything.

Name and tags [Info](#)

Name

[Add additional tags](#)

We will use an Amazon Machine Image (AMI) that has all the software requirements for Parabricks. Under "Application and OS Images" search "Deep Learning AMI" and select any recent version.

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

AMI from catalog

Recents

My AMIs

Quick Start

Amazon Machine Image (AMI)

Deep Learning AMI GPU TensorFlow 2.10.0
(Ubuntu 20.04) 20221104
ami-0a4f85450228bf9ed

Verified provider

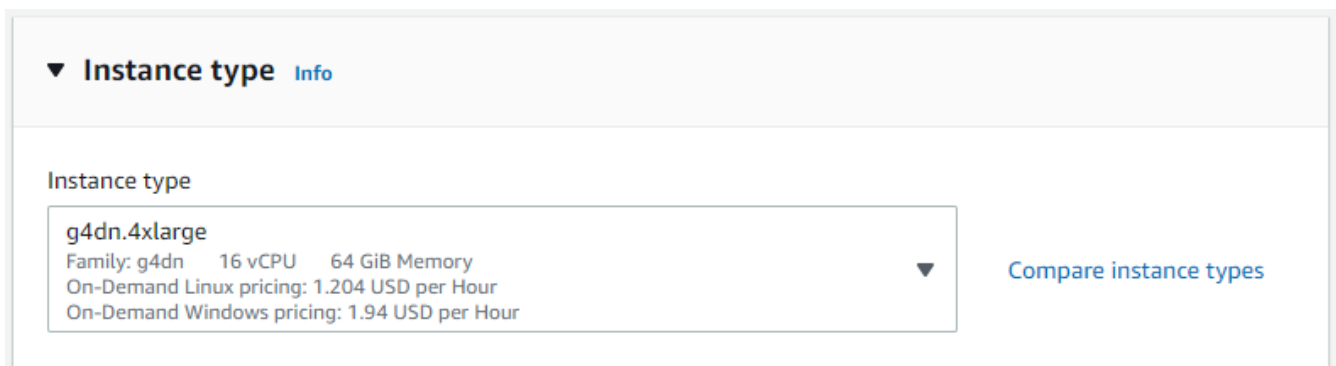


[Browse more AMIs](#)

Including AMIs from
AWS, Marketplace and
the Community

Catalog	Published	Architecture	Virtualization	Root device type	ENA Enabled
Quickstart AMIs	2022-11-07T18:40:05.00Z	x86_64	hvm	ebs	Yes

For installing and testing Parabricks, we will need an instance with at least 1 GPU. Under “Instance type” select “Compare instance types”. In the search bar type “g4dn.4xlarge” and select that instance type from the list of options. This instance has 1 NVIDIA T4 GPU with 16 vCPUs and 64 GB of RAM. Read more about g4dn instances on the [AWS documentation](#).



We need to select a key pair if we want to use SSH to log into the instance. For this tutorial, we will be logging into the instance using “EC2 Instance Connect” which does not require a key pair. In the “Key pair” drop-down, we will select the first options “Proceed without a key pair”.

However, if you do want to generate a key pair, select “Create new key pair”, give the key pair a name, and select “Create key pair”. The key will automatically download. Save this for a later step.

▼ **Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

parabricks_key ▼ [Create new key pair](#)

Lastly, we must increase the storage quota so that when we download and run our test data, we have enough disk space. Under “Configure storage” change the default Root volume size to 500 GB.

▼ **Configure storage** [Info](#) Advanced

1x GiB Root volume (Not encrypted)

[Free tier eligible customers can get up to 30 GB of EBS General Purpose \(SSD\) or Magnetic storage](#) ✕

[Add new volume](#)

Instance store volumes [Show details](#)

Instance Type Volumes are not included in the template unless modified

The selected AMI contains more instance store volumes than the instance allows. Only the first 1 instance store volumes from the AMI will be accessible from the instance

0 x File systems [Edit](#)

Our instance is ready to be launched now. Select “Launch instance”.

▼ **Summary**

Number of instances [Info](#)

1

Software Image (AMI)
NVIDIA GPU-Optimized AMI 22.06...[read more](#)
ami-06e634ccd429848fc

Virtual server type (instance type)
g4dn.xlarge

Firewall (security group)
New security group

Storage (volumes)
2 volume(s) - 157 GiB

i **Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet. **X**

Cancel **Launch instance**

The instance should begin to launch. Navigate back to the “Instances” section of the left side panel and select “Instances to confirm that the instance is running.

Instances (1) Info

Find instance by attribute or tag (case-sensitive)

Key name = parabricks_key X Clear filters

<input type="checkbox"/>	Name	Instance ID	Instance state
<input type="checkbox"/>	Parabricks	i-05f1a151df2c4d3aa	Running

Click on the checkbox next to the instance and a box will appear in the top right saying "Connect". Click that button. If you generated a key pair in the previous steps, you can use it to connect using the SSH client. However, we will be connecting using "EC2 Instance Connect" which does not require a key-pair. Click connect.

Connect to instance [Info](#)

Connect to your instance i-05f1a151df2c4d3aa (Parabricks) using any of these options

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Instance ID
i-05f1a151df2c4d3aa (Parabricks)

Public IP address
44.204.72.177

User name

Connect using a custom user name, or use the default user name root for the AMI used to launch the instance.

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

Cancel **Connect**

We are now greeted with a full terminal with our NVIDIA GPU-Optimized AMI pre-installed. We are now ready to start installing Parabricks.

```
AWS Services Search [Alt+S] N. Virginia AWSOS-AD-Admin/gburnett@4926-8111-8881
CloudFormation Elastic Container Registry Console Home Batch IAM EC2 Cloud9
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.13.0-1031-aws x86_64)
+ Documentation: https://help.ubuntu.com
+ Management: https://landscape.canonical.com
+ Support: https://ubuntu.com/advantage
System information as of Thu Nov 17 00:00:57 UTC 2022
System load: 0.13 Processes: 138
Usage of /: 16.6% of 30.84GB Users logged in: 0
Memory usage: 2% IPv4 address for ens5: 172.31.46.79
Swap usage: 0%
11 updates can be applied immediately.
7 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
*** System restart required ***
The following AWS CLI version has been pre-installed. Begin using the AWS CLI by first configuring your credentials using 'aws configure'
aws-cli/2.7.8 Python/3.9.11 Linux/5.13.0-1031-aws exe/x86_64.ubuntu.20 prompt/off
Welcome to the NVIDIA GPU Cloud image. This image provides an optimized
environment for running the deep learning and HPC containers from the
NVIDIA GPU Cloud Container Registry. Many NGC containers are freely
available. However, some NGC containers require that you log in with
a valid NGC API key in order to access them. This is indicated by a
'pull access denied for xyz ...' or 'Get xyz: unauthorized: ...' error
message from the daemon.
Documentation on using this image and accessing the NVIDIA GPU Cloud
Container Registry can be found at
http://docs.nvidia.com/ngc/index.html
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
root@ip-172-31-46-79:~#
```

Installing Parabricks

We will install Parabricks into our instance that we just created. To do this, we will use the NVIDIA GPU Cloud (NGC) to download the Parabricks Docker image.

Visit the [Parabricks page on NGC](#) to get the Docker pull command for the latest version of Parabricks.

Nvidia Clara Parabricks

Copy Image Path ▾



Description

Nvidia Clara Parabricks is an accelerated compute framework that supports applications across the genomics industry, primarily supporting analytical workflows for DNA, RNA, and somatic mutation detection applications

Publisher

Nvidia

Latest Tag

4.0.0-1

Modified

November 8, 2022

Compressed Size

2.02 GB

Multinode Support

No

Multi-Arch Support

Overview

Tags

Layers

Security Scanning

Related Collections

These instructions and commands are valid for Clara Parabricks v4.0.0-1 only. For earlier versions, please visit [Parabricks user guides](#) for each specific older version.

Note, you will need an installer for versions prior to v4.0.0-1. Instructions for this are also in the [Parabricks user guides](#).

What is Nvidia Clara Parabricks?

Nvidia Clara Parabricks is an accelerated compute framework that supports applications across the genomics industry, primarily supporting analytical workflows for DNA, RNA, and somatic mutation detection applications. With industry leading compute times, Parabricks rapidly converts a FASTQ file to a VCF using multiple, industry validated variant callers and also includes the ability to QC and annotate those variants. As Parabricks is based upon publicly available tools, results are easy to verify and combine with other publicly available datasets.

More information is available on the [Clara Parabricks Product Page](#).

Detailed installation, usage, and tuning information is available in the [Parabricks user guide](#).

The Clara Parabricks docker image can be obtained by running the following command:

```
$ docker pull nvcr.io/nvidia/clara/clara-parabricks:<TAG>
```

An example run of the fq2bam tool using the container will be as follow:

```
#This command assumes all the inputs are in <INPUT_DIR> and all the outputs go to <OUTPUT_DIR>.
$ docker run --rm --gpus all -v <INPUT_DIR>:/workdir \
-v <OUTPUT_DIR>:/outputdir \
-v <TMP_DIR>:/raid/myrun -w /workdir \
nvcr.io/nvidia/clara/clara-parabricks:<TAG> \
pbrun fq2bam \
--ref /workdir/$(REFERENCE_FILE) \
--in-fq /workdir/$(INPUT_FASTQ_1) /workdir/$(INPUT_FASTQ_2) \
--knownSites /workdir/$(KNOWN_SITES_FILE) \
--out-bam /outputdir/$(OUTPUT_BAM) \
--out-recal-file /outputdir/$(OUTPUT_RECAL_FILE)
```

Back in our EC2 instance, let's run the docker pull command:

```
$ docker pull nvcr.io/nvidia/clara/clara-parabricks:4.3.1-1
```

```
gburnett_nvidia_com@parabricks:~$ docker pull nvcr.io/nvidia/clara/clara-parabricks:4.0.0-1
4.0.0-1: Pulling from nvidia/clara/clara-parabricks
d7bfe07ed847: Pull complete
bbbbd451a669: Pull complete
773163705c35: Pull complete
d6949fcf1aef: Pull complete
3eb73064088b: Pull complete
a3ac3ab0ee35: Pull complete
8d88682a5e1d: Pull complete
Digest: sha256:0170beef24131a23bb63bc36ec059e493df1f04a4a78f9d2c5df9bce1d5d9a35
Status: Downloaded newer image for nvcr.io/nvidia/clara/clara-parabricks:4.0.0-1
nvcr.io/nvidia/clara/clara-parabricks:4.0.0-1
```

Now Parabricks is installed! Let's run some sample data to test it.

Testing Parabricks

Parabricks provides a small sample dataset as a test for the installation and hardware which can be downloaded using:

```
wget -O parabricks_sample.tar.gz  
"https://s3.amazonaws.com/parabricks.sample/parabricks_sample.tar.gz"
```

When the download completes we can untar the data using:

```
tar xzvf parabricks_sample.tar.gz
```

The *parabricks_sample* folder should look like this when we're done:

```
gburnett_nvidia_com@parabricks:~$ tree parabricks_sample  
parabricks_sample  
├── Data  
│   ├── sample_1.fq.gz  
│   └── sample_2.fq.gz  
└── Ref  
    ├── Homo_sapiens_assembly38.dict  
    ├── Homo_sapiens_assembly38.fasta  
    ├── Homo_sapiens_assembly38.fasta.amb  
    ├── Homo_sapiens_assembly38.fasta.ann  
    ├── Homo_sapiens_assembly38.fasta.bwt  
    ├── Homo_sapiens_assembly38.fasta.fai  
    ├── Homo_sapiens_assembly38.fasta.pac  
    ├── Homo_sapiens_assembly38.fasta.sa  
    ├── Homo_sapiens_assembly38.known_indels.vcf.gz  
    └── Homo_sapiens_assembly38.known_indels.vcf.gz.tbi  
  
2 directories, 12 files
```

Finally, we can run any of the Parabricks pipelines on it. Let's run the [germline pipeline](#) using the following command:

```
$ docker run \ --rm \ --gpus all \ --volume `pwd`:`pwd` \ --workdir  
`pwd`/parabricks_sample \ nvc.io/nvidia/clara/clara-parabricks:4.3.1-1 \ pbrun
```



```
germline \ --ref Ref/Homo_sapiens_assembly38.fasta \ --in-fq Data/sample_1.fq.gz
Data/sample_2.fq.gz \ --knownSites
Ref/Homo_sapiens_assembly38.known_indels.vcf.gz.tbi \ --out-bam output.bam \ --
out-variants germline.vcf \ --out-recal-file recal.txt
```

We can tell that Parabricks started correctly when we see the Parabricks banner and the ProgressMeter begins to populate with values:

```
Please visit https://docs.nvidia.com/clara/#parabricks for detailed documentation

[Parabricks Options Msg]: Automatically generating ID prefix
[Parabricks Options Msg]: Read group created for /home/gburnett_nvidia_com/parabricks_sample/Data/sample_1.fq.gz and
/home/gburnett_nvidia_com/parabricks_sample/Data/sample_2.fq.gz
[Parabricks Options Msg]: @RG\tID:HK3TJBCX2.1\tLB:lib1\tPL:bar\tSM:sample\tPU:HK3TJBCX2.1

[Parabricks Options Msg]: Checking argument compatibility
[Parabricks Options Msg]: Read group created for /home/gburnett_nvidia_com/parabricks_sample/Data/sample_1.fq.gz and
/home/gburnett_nvidia_com/parabricks_sample/Data/sample_2.fq.gz
[Parabricks Options Msg]: @RG\tID:HK3TJBCX2.1\tLB:lib1\tPL:bar\tSM:sample\tPU:HK3TJBCX2.1
[PB Info 2022-Nov-18 00:54:13] -----
[PB Info 2022-Nov-18 00:54:13] ||                               Parabricks accelerated Genomics Pipeline           ||
[PB Info 2022-Nov-18 00:54:13] ||                               Version 4.0.0-1                               ||
[PB Info 2022-Nov-18 00:54:13] ||                               GPU-BWA mem, Sorting Phase-I           ||
[PB Info 2022-Nov-18 00:54:13] -----
[M::bwa_idx_load_from_disk] read 0 ALT contigs
[PB Info 2022-Nov-18 00:54:18] GPU-BWA mem
[PB Info 2022-Nov-18 00:54:18] ProgressMeter      Reads          Base Pairs Aligned
[PB Info 2022-Nov-18 00:54:51] 5043564          5800000000
[PB Info 2022-Nov-18 00:55:20] 10087128       11600000000
[PB Info 2022-Nov-18 00:55:49] 15130692       17400000000
[PB Info 2022-Nov-18 00:56:19] 20174256       23200000000
[PB Info 2022-Nov-18 00:56:48] 25217820       29000000000
```

This should take ~10 minutes to finish running. When it's done we should see the output files in the sample directory.

Congratulations, we've just run our first Parabricks job!

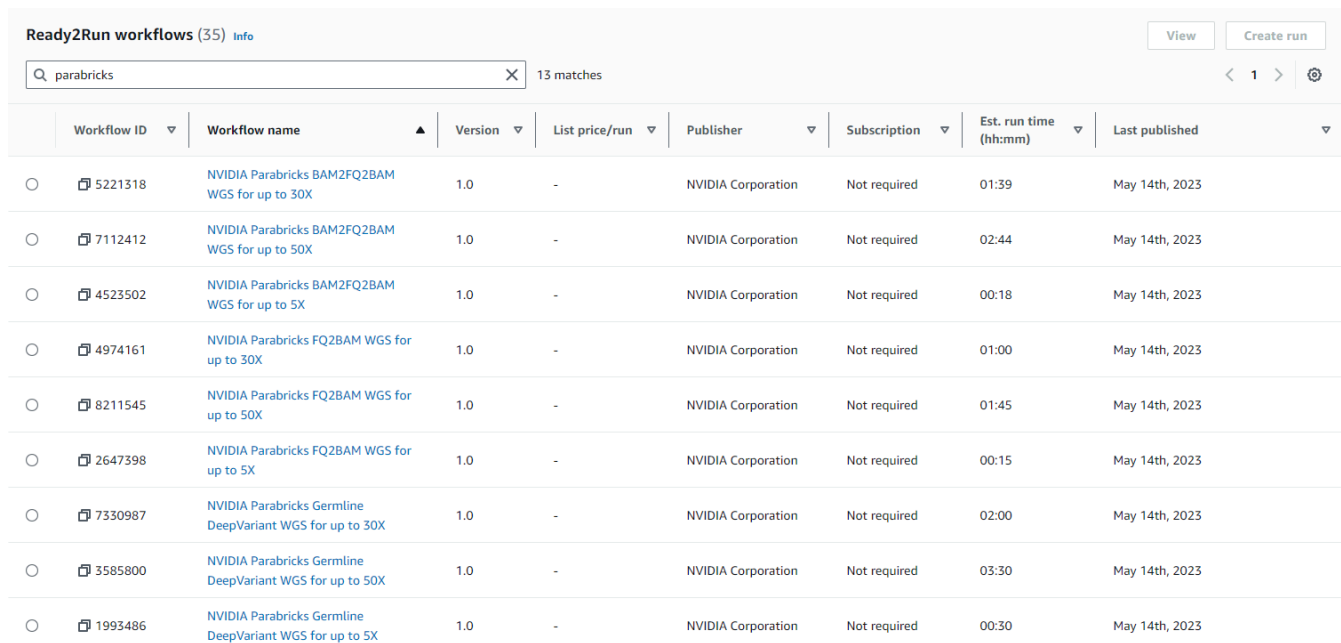
Running Workflows on Amazon HealthOmics

ReadyRun Workflows

Ready2Run Workflows are the pre-made workflows available to anyone on AWS HealthOmics.

Navigate to the AWS HealthOmics homepage and click on “Ready2Run workflows”.

In the search bar type “parabricks” to see all the available Parabricks workflows.



Workflow ID	Workflow name	Version	List price/run	Publisher	Subscription	Est. run time (hh:mm)	Last published
5221318	NVIDIA Parabricks BAM2FQ2BAM WGS for up to 30X	1.0	-	NVIDIA Corporation	Not required	01:39	May 14th, 2023
7112412	NVIDIA Parabricks BAM2FQ2BAM WGS for up to 50X	1.0	-	NVIDIA Corporation	Not required	02:44	May 14th, 2023
4523502	NVIDIA Parabricks BAM2FQ2BAM WGS for up to 5X	1.0	-	NVIDIA Corporation	Not required	00:18	May 14th, 2023
4974161	NVIDIA Parabricks FQ2BAM WGS for up to 30X	1.0	-	NVIDIA Corporation	Not required	01:00	May 14th, 2023
8211545	NVIDIA Parabricks FQ2BAM WGS for up to 50X	1.0	-	NVIDIA Corporation	Not required	01:45	May 14th, 2023
2647398	NVIDIA Parabricks FQ2BAM WGS for up to 5X	1.0	-	NVIDIA Corporation	Not required	00:15	May 14th, 2023
7330987	NVIDIA Parabricks Germline DeepVariant WGS for up to 30X	1.0	-	NVIDIA Corporation	Not required	02:00	May 14th, 2023
3585800	NVIDIA Parabricks Germline DeepVariant WGS for up to 50X	1.0	-	NVIDIA Corporation	Not required	03:30	May 14th, 2023
1993486	NVIDIA Parabricks Germline DeepVariant WGS for up to 5X	1.0	-	NVIDIA Corporation	Not required	00:30	May 14th, 2023

Clicking on any workflow will take us to the workflow homepage where we can see descriptions and diagrams for what the workflow does, parameters that are accepted, and the run history.

NVIDIA Parabricks FQ2BAM WGS for up to 30X [Info](#)

Create run

[Details](#) | [Parameters](#) | [Run history](#)

Details

Workflow ID 4974161	Run storage capacity (TB) 1.2	Publisher NVIDIA Corporation
Name NVIDIA Parabricks FQ2BAM WGS for up to 30X	Workflow accelerators GPU	Subscription No
Workflow description This workflow provides a GPU-accelerated version of GATK4 best practices for alignment with BWA-MEM, and provides functionally equivalent results in significantly less time. Given one or more pairs of FASTQ files (up to 30X WGS), this workflow will utilize the HG38 reference and generate a BAM.	Workflow software Clara Parabricks 4.0.1-1	Estimated run time (hh:mm) 01:00
Workflow version 1.0	Workflow files View test and example parameter files	Amazon Resource Name (ARN) <code>arn:aws:omics:us-west-2:workflow/4974161</code>
Workflow language WDL	Workflow help Support information	Published May 14, 2023, 20:00 (UTC -8)
	Max total input size 71GiB	
	List price/run -	

Workflow overview

The following information describes how the workflow was built.

Click on “Create run” and enter a name, a destination for the output files, and the input parameters.

[AWS HealthOmics](#) > [Ready2Run workflows](#) > [NVIDIA Parabricks FQ2BAM WGS for up to 30X](#) > [Create and run](#)

Step 1
Specify run details

Step 2
Add parameter values

Step 3
Review and start run

Specify run details [Info](#)

Choose the desired workflow for the run, and include details like name, run priority, storage capacity, and output location.

Run details

A run ID will be generated automatically once the workflow is run. Although a run name is not required, it can be useful in helping identify specific runs later.

Workflow ID
Workflow name will display below the workflow ID if one was previously provided.

4974161
NVIDIA Parabricks FQ2BAM WGS for up to 30X

Run name

My run

Maximum of 127 characters.

Run priority
Run priority sets priority within a run group. 100 is the default priority. The higher the number, the greater the priority. 0 is lowest priority.

100

Integers only. Range is 0-1,000

Run storage capacity
The run storage capacity temporarily stores input and output data files from each task. This storage is deleted after the run completes.

1.2TB

Select S3 output destination [Info](#)
Choose the S3 location you would like this workflow run to be saved.

s3://bucket/prefix/object [View](#) [Browse S3](#)

Format: s3://bucket/prefix/object.

Run meta data retention mode [Info](#)
Retain workflow run meta data inside HealthOmics up to your maximum Runs account quota, or have the system remove your oldest runs

Back in the HealthOmics console, you can click on “Runs” in the left sidebar and see the job as it runs and while it’s completed:

AWS HealthOmics × [AWS HealthOmics](#) > [Runs](#)

How it works [Info](#)
View your run history including real-time run status, completed run CPUs and memory used, and the final cost per run. Also re-run past runs, and view past run details including individual run task logs. View run logs provided by Cloudwatch.

Runs (51) [Info](#) 24 matches Delete Clone run Re-run Start run

	Run ID	Name	Status	Workflow ID	Workflow name	Priority	Run time (dd:hh:mm:ss)	Created (UTC-8)
<input type="radio"/>	4486669	test: fq2bam	Completed	5185763	fq2bam	0	00:00:56:45	December 14, 2023, 16:09
<input type="radio"/>	3227051	test: fq2bam	Completed	4692085	fq2bam	0	00:00:40:22	October 20, 2023, 15:59
<input type="radio"/>	8508580	test: fq2bam	Completed	4692085	fq2bam	0	00:00:41:11	October 13, 2023, 14:02
<input type="radio"/>	8478355	test: fq2bam	Completed	7337214	fq2bam	0	00:00:47:19	October 12, 2023, 15:58
<input type="radio"/>	5508804	test: bam2fq2bam	Completed	4444888	bam2fq2bam	0	00:01:19:33	September 25, 2023, 16:48
<input type="radio"/>	4391690	test: germline-deepvariant	Completed	4423358	germline-deepvariant	0	00:01:21:20	September 25, 2023, 16:44
<input type="radio"/>	4846295	test: germline-haplotype	Completed	3862355	germline-haplotype	0	00:01:01:56	September 25, 2023, 16:44

Clicking on the job will show you information such as:

- if the job completed,
- what the inputs were,
- and where the outputs are.

[AWS HealthOmics](#) > [Runs](#) > [test: fq2bam](#)

test: fq2bam [Info](#) Delete Clone run Re-run

[Details](#) [Tasks](#) [Tags](#)

Tasks (3) [Info](#)
Choose View in the Logs column to view a task logstream in Cloudwatch.

1 ⌂

ID	Name	Status	Cloudwatch logs	Run time (dd:hh:mm:ss)	vCPUs	Memory(GiB)	GPU(s)	Start	Stop
3489126	fq2bam	Completed	View Logstream	00:00:17:48	48	192	4	December 14, 2023, 16:45	December 14, 2023, 17:03
9328205	parse_inputs-1	Completed	View Logstream	00:00:00:34	1	1	0	December 14, 2023, 16:33	December 14, 2023, 16:33
6147562	parse_inputs-0	Completed	View Logstream	00:00:00:34	1	1	0	December 14, 2023, 16:31	December 14, 2023, 16:31

And that's it! Running on AWS HealthOmics Ready2Run is designed to be simple and intuitive. You can run any number of the provided workflows using these same steps.

Private Workflows

For users who want more control over how the workflows run, we provide Private Workflows as well. These are edited locally and then run on AWS HealthOmics platform, so users can take advantage of the HealthOmics console while maintaining flexibility in the workflows themselves.

The Parabricks Private Workflows and full instructions are available in our [GitHub repository](#).

Closing Remarks

We encourage you to expand on the demo in this guide by using your own data, trying other pipelines, and generally exploring what Parabricks has to offer. Check out the [documentation](#) for more information about the different pipelines available. You can also find our online developer community on the [Parabricks forum](#), where you can ask questions and search through answers while you are learning how to use Parabricks.

© Copyright 2024, Nvidia.. PDF Generated on 06/05/2024