Technical Report

AI Deployment on the Cloud with NetApp Cloud Volumes Service

Enabling AI Deployments on Amazon Web Services

Nabil Fares and Sundar Ranganathan, NetApp
October 2018 | TR-4718

Abstract

By leveraging artificial intelligence (AI) in their applications, organizations can quickly make data-driven decisions to drive efficiencies in their products and services. The cloud is an indispensable platform that not only allows one to quickly experiment with ideas via proof-of-concepts, but also provides a rich software ecosystem for building AI applications and training deep learning models. In addition to compute engines optimized for massively parallel computations on the cloud, you need a high performing file system to work seamlessly with GPUs and keep them busy at ~100% utilization.

This report describes the steps involved in running AI model training with NVIDIA® Tesla® V100 GPUs on Amazon Web Services (AWS) and NetApp® Cloud Volumes Service. Key training performance and inferencing metrics are included.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

1 Introduction

Artificial intelligence (AI) is increasingly being used in multiple industry verticals to support important business needs such as automating business processes, providing cognitive insights through data analysis, and interacting with customers using natural language processing. Deep learning (DL), a version of machine learning, is effective at learning from large volumes of data to mimic the human brain's pattern recognition (for example, images, speech, and text). Organizations face many challenges in using AI in their cognitive projects, and one of the top challenges is choosing the right set of technologies and infrastructure.

The technologies and infrastructure influence the models' prediction accuracy, which is a function of the dataset size used to train them. And speed is an issue too: depending on the complexity of the neural network and the volume of datasets, model training can run from a few hours to days.

DL success requires an agile infrastructure with an ecosystem of technologies and solutions. The cloud offers an ideal environment that enables small and large organizations to get started with a pay-as-you-go structure, keeping costs low. Public clouds can be used to run proofs of concept (POCs), testing and development, data processing before beginning training, and full-scale production training runs. Computation in model training is parallel, and hence graphic processing units (GPUs) are more effective than general-purpose CPUs. AWS offers GPU instances for compute with an ecosystem of solutions to support model training. Using a storage solution with fast file services is essential to maintaining high GPU utilization, completing the training runs faster, and keeping costs to a minimum.

This technical report includes demonstration and test results for DL model training to process images using Amazon Elastic Compute Cloud (EC2) P3 instances (NVIDIA Tesla V100 GPUs), NetApp Cloud Volumes Service, and the TensorFlow framework.

2 Cloud Volumes Service

NetApp Cloud Volumes Service is a fully managed, cloud-native file storage service based on the proven NetApp ONTAP® data management software and other data services. Cloud Volumes Service combines NetApp's vast file services expertise with the simplicity and flexibility of the biggest clouds (AWS, Azure, Google). It also makes it easier to integrate with the ecosystem of applications in the cloud. The service supports NFSv3 and NFSv4 (coming soon) for Linux/UNIX clients, and SMB for Windows clients operating in the cloud.

2.1 Issues with Existing NFS/Shared File Solutions on the Cloud

The advent of hybrid cloud architectures and introduction of containers has simplified application and workload migration between public clouds and on-premises environments. However, the lack of robust file storage services offered by cloud providers inhibits data mobility.

- Nearly 80% of all file data is created through NFSv3, yet there are few options in the cloud that match the file offerings available in on-premises file storage systems.
- Applications storing file data demand high file storage performance, which can often be an expensive proposition in the cloud.
- Issues around achieving data consistency between on-premises environments and the public cloud can result in costs for data ingress and egress.

2.2 Benefits of Cloud Volumes Service

Cloud Volumes Service delivers fast file-storage performance and business-critical file services to cloud applications. This service harnesses NetApp's file services expertise and addresses key issues with cloud-based file storage services. Here are a few key value propositions of this service:

- Cloud Volumes Service is offered in all the top public clouds that support NFS and SMB.

- It is delivered as a simple native cloud service, giving you an easy-to-use interface and a single payment model through the cloud provider.
- It provides features to migrate, replicate, and synchronize data across on-premises environments and the cloud.
- You can use NetApp Snapshot™ copies for data protection and data restores.
- Built-in, always-on encryption allows protection without a performance impact.
- You can scale from zero to 100TB deployments in a matter of seconds.

3 Cloud Volumes Service on AWS

NetApp Cloud Volumes Service for AWS is designed to increase performance while reducing complexity. The multiprotocol storage service natively running in AWS simplifies data migrations for on-premises environments. It supports NFSv3, NFSv4 (coming soon), and SMB. Being a fully managed service, it handles configuring and managing all storage infrastructure. This service is a great fit for developers, line-of-business (LOB) engineers, database administrators, and application architects who consume storage capacity but do not want to administer it. Soon, Cloud Volumes Service will also be integrated with other AWS services, such as Amazon Elastic MapReduce (EMR), Amazon EC2, and Amazon Elastic Container Service (ECS).

The service supports multiple service levels (IOPS per latency targets) so that users can choose a service level per file share that meets the application need. Three service levels are available:

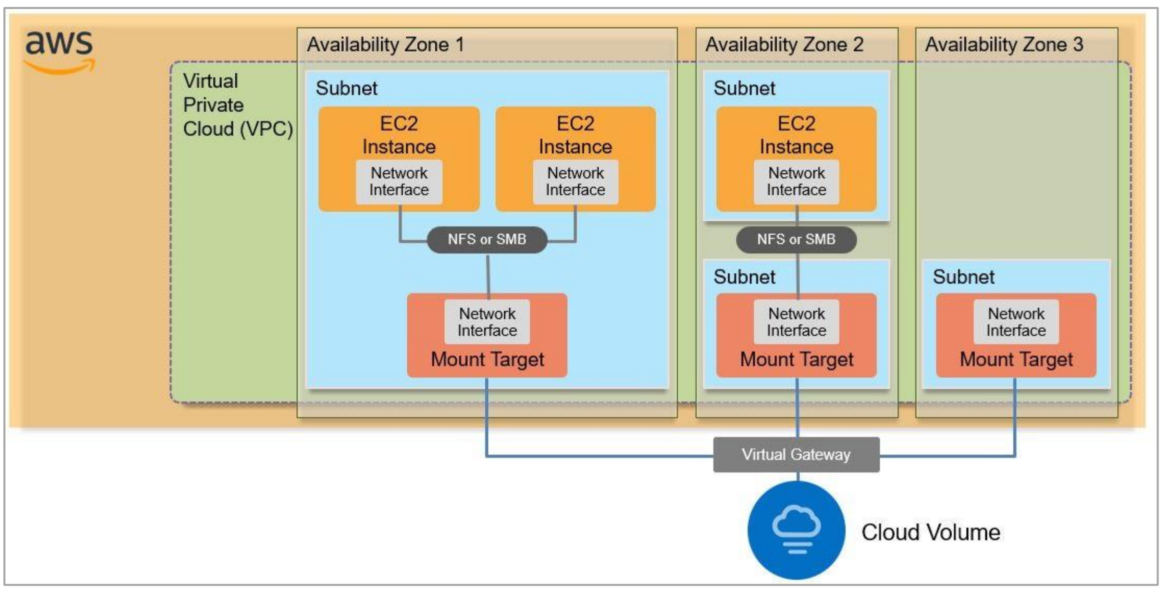
- **Standard,** which provides 1,000 IOPS per TB (16k I/O) and 16MB of throughput per TB
- **Premium,** which provides 4,000 IOPS per TB (16k I/O) and 64MB of throughput per TB
- **Extreme,** which provides 8,000 IOPS per TB (16k I/O) and 128MB of throughput per TB

Note: There are read and write performance limits for single Amazon EC2 instances of ~5Gbps. Much higher performance can be achieved with multiple instances.

Cloud Volumes Service is initially being sold in bundles on the AWS Marketplace, and a pay-as-you-go option will follow. Cloud Volumes is billed by AWS on effective or logical capacity through the AWS billing system. Customers receive a single bill from AWS that includes the charges for this service.

Figure 1 illustrates how Cloud Volumes Service is architected and integrated functionally with AWS.

Figure 1) Cloud Volumes Service architecture.



Cloud Volumes Service is positioned behind AWS virtual private gateways (VPGs) with visibility from any availability zone in a Virtual Private Cloud (VPC) in a particular region. The service allows a single IP address mount from any availability zone for performance consistency and simplicity. Tenants can restrict each availability zone by its Classless Inter-Domain Routing (CIDR) on the Cloud Volumes portal. For example, multiple volumes can be created, and each volume can be restricted to a specific availability zone.

4 TensorFlow

TensorFlow, developed by the Google Brain team, is an open-source software library for numerical computation and large-scale machine learning (ML) applications such as neural networks. This framework facilitates the process of acquiring data, training models, and refining future results. TensorFlow bundles a slew of ML and DL models and algorithms. Its architecture allows deployment across various compute platforms, such as CPUs, GPUs, and tensor processing units (TPUs).

TensorFlow can be used to train and run deep neural networks for applications such as image recognition, recurrent neural networks, and natural language processing–based simulations.

5 Configuration

DL in the cloud is achievable with a simplified approach and rapid verification. NetApp Cloud Volumes Service lets you use common tools and datasets similar to those used in on-premises training. The following sections present the requirements and components typically involved.

This technical report demonstrates the service’s ability to handle high-performance GPUs; we selected Amazon EC2 P3 instances to deliver the highest performance available (NVIDIA Tesla V100 GPUs) at the time of this writing. The following sections describe the steps involved in provisioning Cloud Volumes and details around running model training in the cloud.

5.1 Configuring Cloud Volumes

Configuring a cloud volume is simple. When you provision a volume, the most important aspect is determining the right quality-of-service (QoS) tier suitable for your application to ensure that the GPUs are not restricted. Current tiering policies are as follows:

- **Standard:** 16MBps per 1TB
- **Premium:** 64MBps per 1TB
- **Extreme:** 128MBps per 1TB

The Extreme policy was selected to meet the testing requirements for the type of GPU available, and a volume size of 10TB was provisioned for performance and capacity.

Before you can perform the Cloud Volumes tasks that are described in this report, you must subscribe to Cloud Volumes Service for AWS. The subscription process includes the initial setup and configuration, visit AWS Marketplace for more details.

Note: AWS P3 instances are not available in all regions. Verify that your preferred region supports the GPU required for training.

Selecting the Region

Cloud Volumes Service is available in several AWS regions, but the UI is specific to each region. Before you create a volume, you must specify the region where you want to use for the service. For this validation, US-East was selected because of the GPU availability in this region.

1. Navigate to the [NetApp Cloud Orchestrator](#)site, and then log in with the e-mail address that you provided during your subscription.
2. From the available AWS regions drop-down menu in the top panel, select a region.

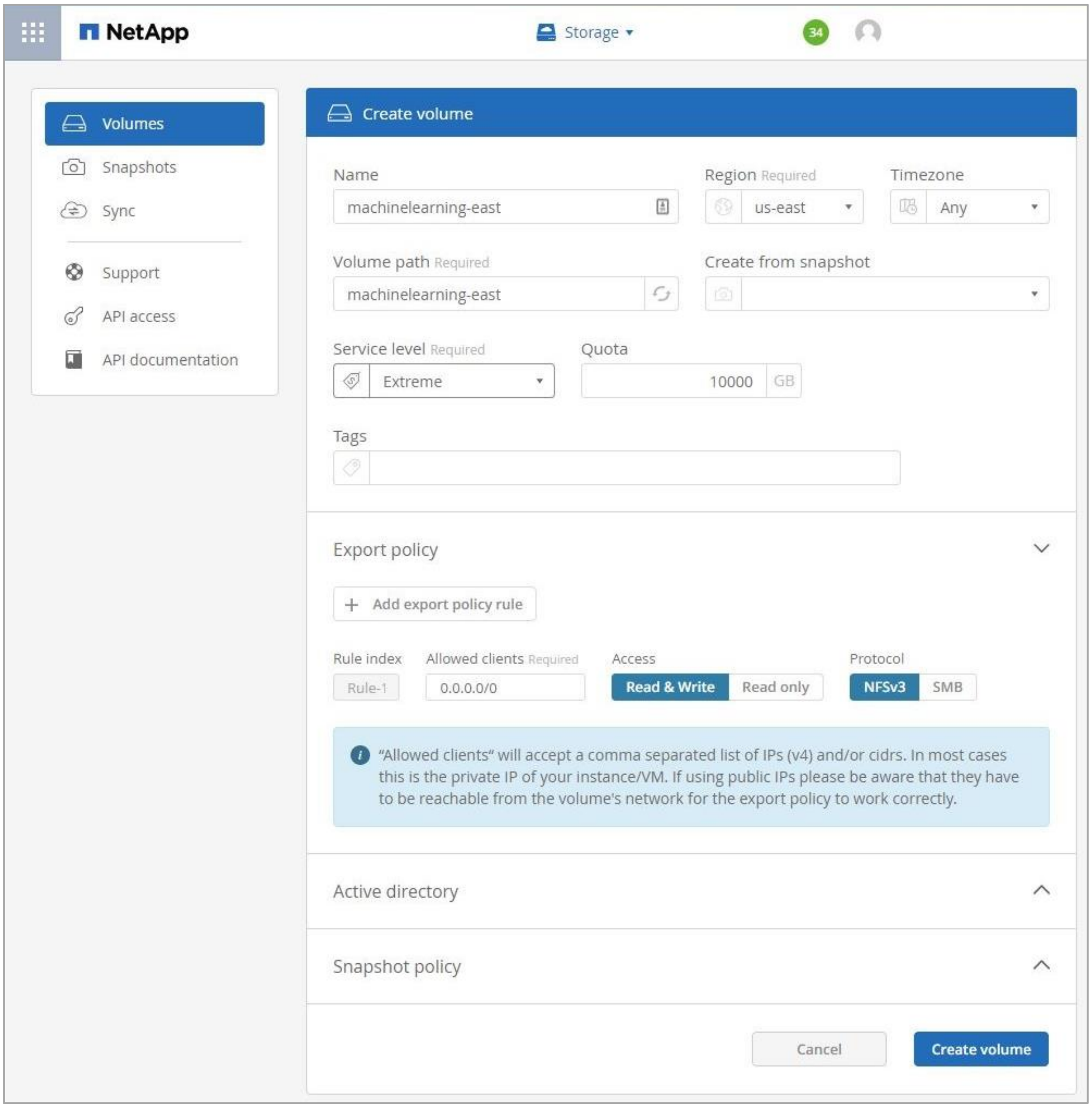
3. Repeat step 2 for each additional region you want to use. For this validation, Us-east-1 was provisioned.

Creating a Cloud Volume

The following steps demonstrate how to create cloud volumes from the NetApp Cloud Orchestrator site.

1. Click the Create New Volume button.

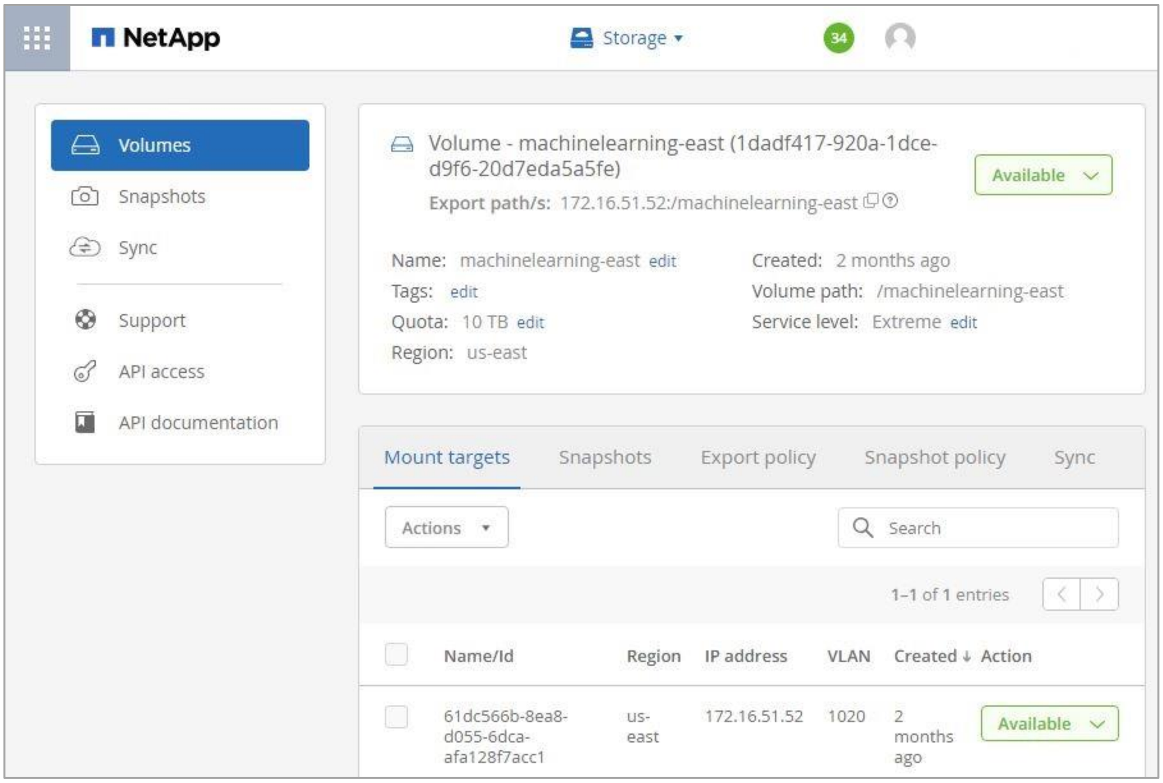
2. Select NFS, SMB, or Dual-Protocol on the Create Volume page, and provide information for the following fields. This technical report focuses on NFS only.
   – **Name:** machinelearning-east
   – **Region:** us-east
   – **Time Zone:** Any
   – **Volume Path:** machinelearning-east
   – **Service Level:** Extreme
   – **Quota:** 10000GB (10TB)
   – **Export Policy:** Left as default for this technical report.



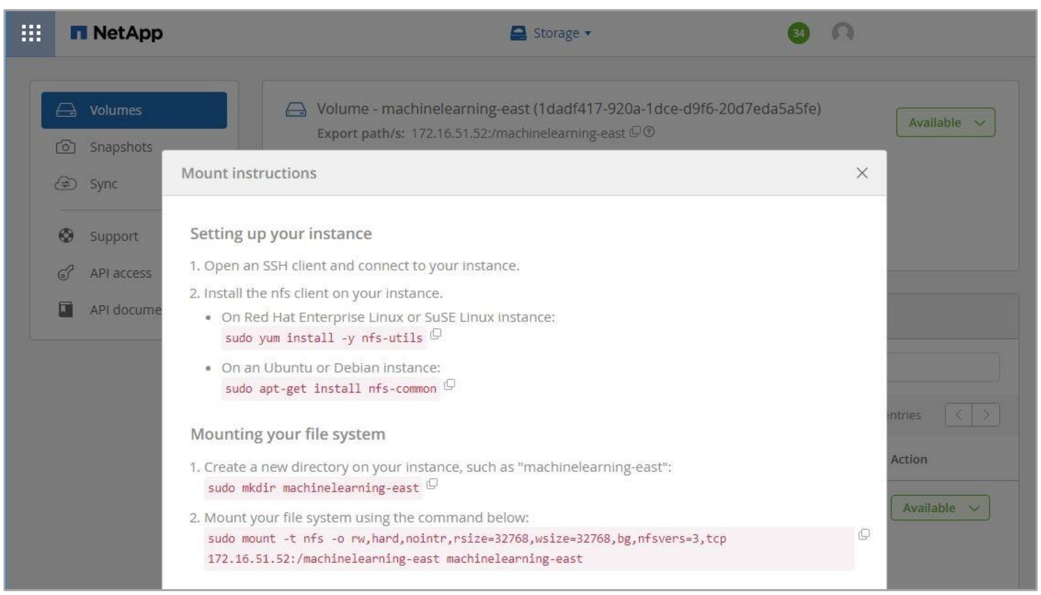
3. Scroll down the Create Volume page, and then click Create Volume.

Mounting a Cloud Volume

1. Click the newly created volume to view additional details. The following screenshot depicts the volume details, the export path, and the option to modify certain details —for example service level and quotas.



2. Click the question mark to display the mount details specific to the selected volume. The following screenshot illustrates the details to mount the NFS volume and install the tools on the Amazon EC2 instance if needed.



Basically, there is only one step to create the volume: It’s as simple as filling out the initial page and clicking Create.

Now the volume is ready to be mounted from any Amazon EC2 instance in the associated VPC. The VPC ID was provided to NetApp during the account creation in the early phases and is not covered in this technical report.

5.2 Configuring AWS Resources

NVIDIA GPU EC2 instances on AWS are available only in particular regions and availability zones. For this validation, the N. Virginia region was selected.

Amazon EC2 P3 instances deliver the highest performance compute for machine learning. Powered by up to eight NVIDIA Tesla V100 GPUs – available in one (p3.2xlarge), four (p3.8xlarge), and eight (p3.16xlarge) V100 GPUs.

In addition to the specific Amazon EC2 instance, the 'NVIDIA Volta Deep Learning AMI' is used. The environment is optimized for running the DL frameworks and HPC containers that are available from the NVIDIA GPU Cloud (NGC) container registry. The Docker containers available on the NGC container registry are tuned, tested, and certified by NVIDIA to take full advantage of Tensor cores, the new driving force behind AI. DL and HPC containers from the NGC container registry require this AMI for the best GPU acceleration on AWS P3 Volta instances.

Note: To download the containers to the Amazon EC2 instance, a free account on NGC is required. After you establish an account, an API key is also required; you can generate it from the NGC portal after you complete your registration.

Environment

- Instance type: P3.16xlarge
- GPU: 8x NVIDIA Tesla V100
- OS: Ubuntu 16.04 LTS
- AMI: NVIDIA Volta Deep Learning AMI
- NVIDIA Driver: NVIDIA Volta Deep Learning AMI 384.xx
- NVIDIA CUDA®: 9.0.176
- CUDA Deep Neural Network (cuDNN) library: 7.1.4
- NVIDIA Collective Communications Library (NCCL): 2.2.13 (optimized for NVLink™)
- TensorFlow version: 1.8.0
- Docker CE: 18.03.1-ce
- Docker Engine Utility for NVIDIA GPUs: 2.03
- Container image: 18.07-py3 (Contains Python3.5)

Launching the Amazon EC2 Instance

These steps give details for instantiating the Amazon EC2 instances from the AWS console. The steps assume that you have an AWS account and are familiar with the AWS console.

We migrated the data to the cloud volume provisioned for this exercise.

Prerequisites

Before you can perform the tasks that are described in this section, you must subscribe to NGC and Amazon AWS.

1. Log in to the AWS console and click on the EC2 Dashboard link.
2. Click Launch Instance; then click AWS Marketplace from the Quick Start menu on the left.
3. In the search box, type `nvidia volta` and press Enter. This results in a single selection. Click the Select button to proceed to the next step.

4. A dialog box detailing the NVIDIA Volta Deep Learning AMI appears. Read the details and click Continue to proceed to the Instance Type page.
5. From the Instance Type page, filter on GPU Compute by clicking the All Instance Types drop-down menu and select the p3.16x instance. Click Next: Configure Instance Details.

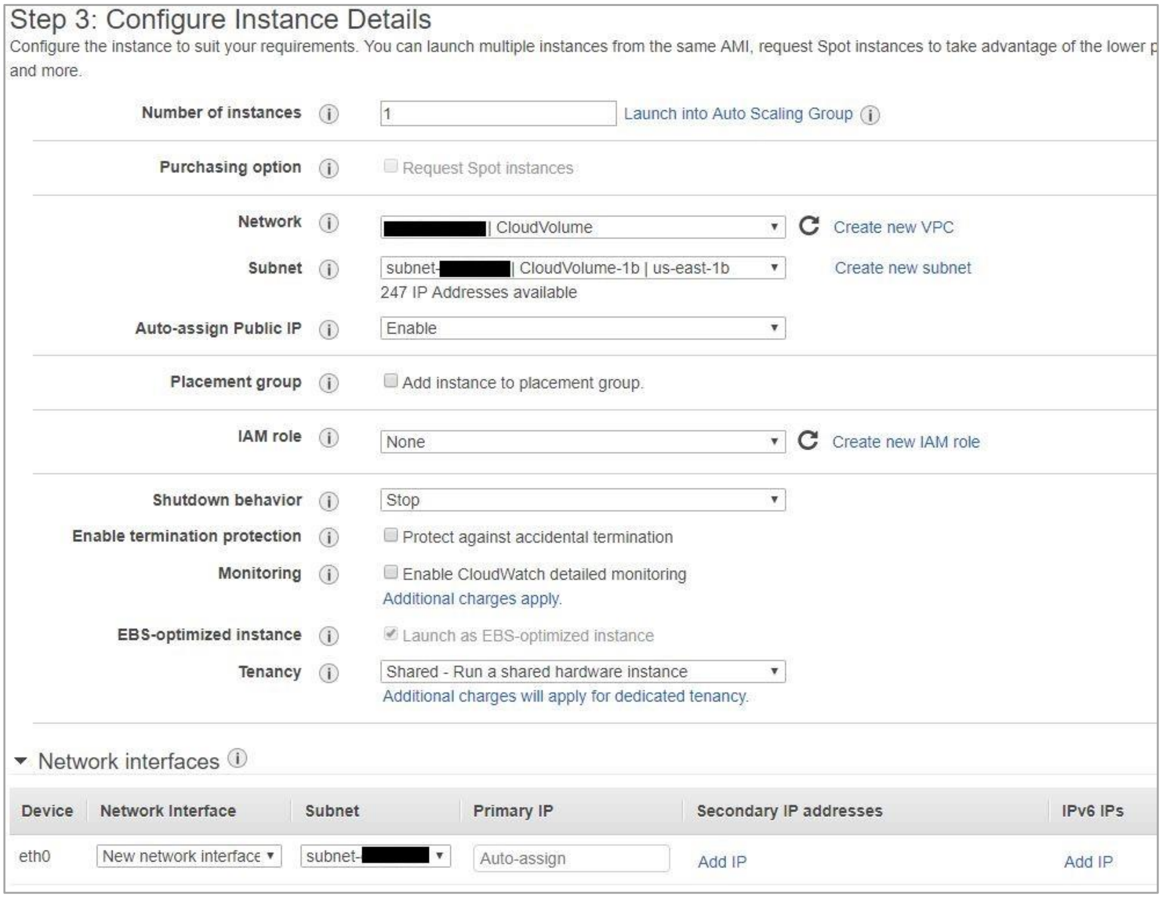
Filter by: GPU compute | Current generation | Show/Hide Columns

Currently selected: p3.16xlarge (188 ECUs, 64 vCPUs, 2.7 GHz, Intel Xeon E5-2686 v4, 488 GiB memory, EBS only)

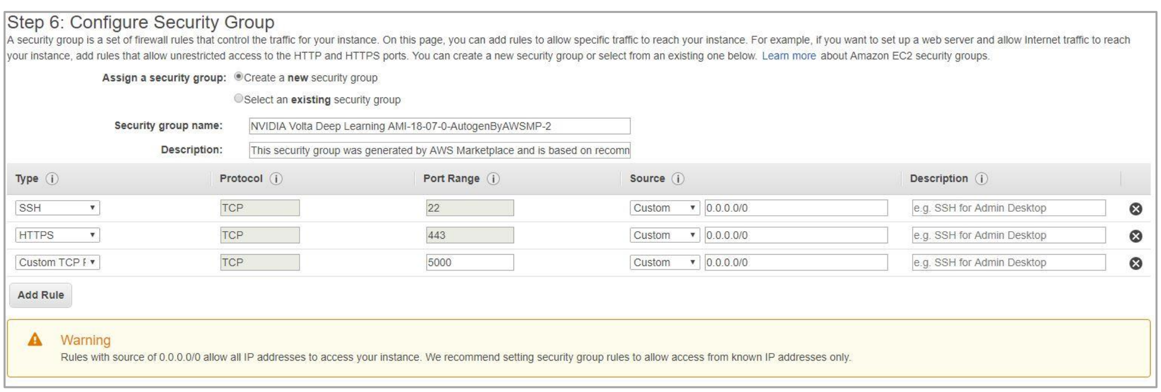
Note: The vendor recommends using a **p3.2xlarge** instance (or larger) for the best experience with this product.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
	GPU compute	p2.xlarge	4	61	EBS only	Yes	High	Yes
	GPU compute	p2.8xlarge	32	488	EBS only	Yes	10 Gigabit	Yes
	GPU compute	p2.16xlarge	64	732	EBS only	Yes	25 Gigabit	Yes
	GPU compute	p3.2xlarge	8	61	EBS only	Yes	Up to 10 Gigabit	Yes
	GPU compute	p3.8xlarge	32	244	EBS only	Yes	10 Gigabit	Yes
	GPU compute	p3.16xlarge	64	488	EBS only	Yes	25 Gigabit	Yes

6. On the Configure Instance Details page, select the VPC that was attached to Cloud Volumes Service during the activation phase; in this example, the VPC name is CloudVolume. Select the subnet corresponding to the availability zones with GPU instances: in the N. Virginia region, zones 1b, 1c, and 1f of NVIDIA Tesla V100 GPUs. Select Enable Auto-Assign Public IP (required if the instance must be accessed from a remote location outside the VPC). Leave the remaining fields unchanged unless your cloud policy requires different settings. Then click Next: Add Storage.



7. Accept the default storage option for this instance and click Next: Add Tags. Add a name tag and click Next: Configure Security Group.



8. The NVIDIA Volta AMI creates a security group for this instance. Accept or modify it to meet your security policy. The default rules for this security group permit access to the instance from anywhere to the ports as shown here. If you're satisfied with the defaults, click Review and Launch.
9. Review the instance details and click Launch. Select an existing key pair or create a new one. Select the checkbox to accept the acknowledgment and click Launch Instances to return to the Amazon EC2 console while the instance is launching.

Note: The AWS instance's public IP address to access the instance for the next section, "Configuring the NVIDIA Volta Instance."

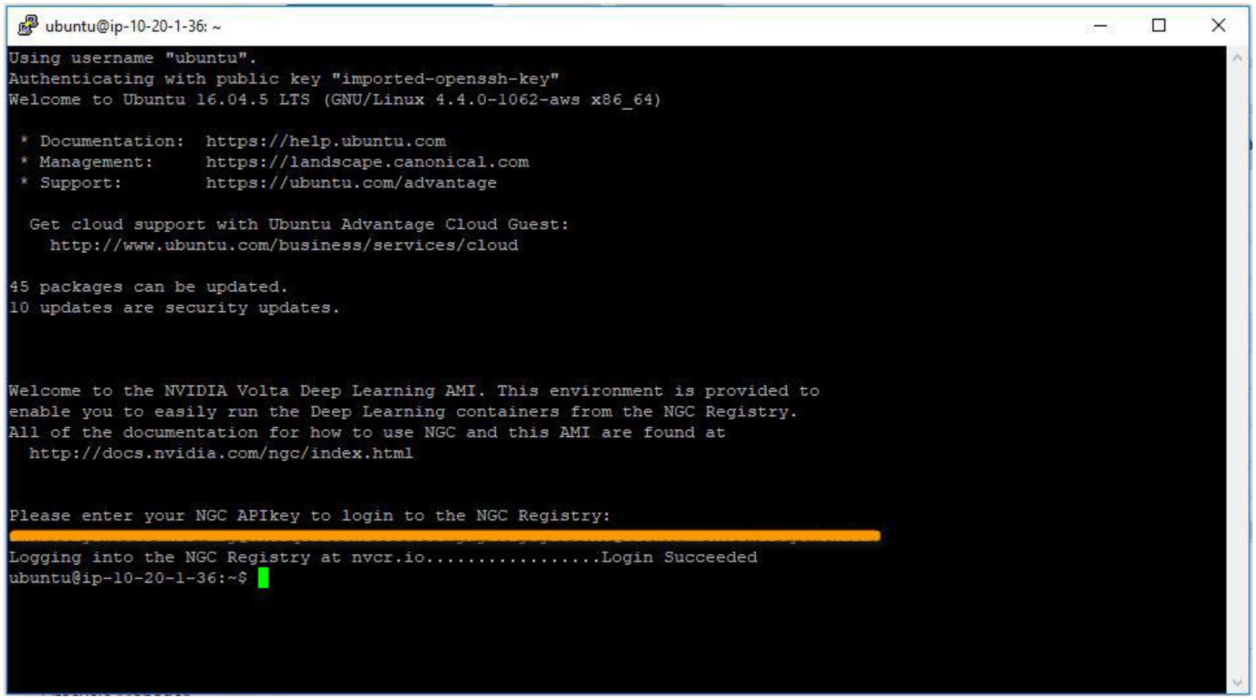
Configuring the NVIDIA Volta Instance

At this point, you should be able to access the Amazon EC2 instance through Secure Shell (SSH) by using the public or private IP address. Use any SSH client to access the instance.

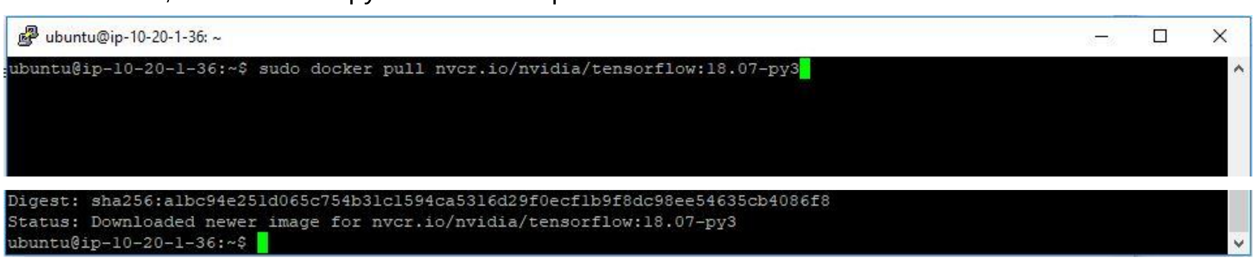
Prerequisites

For this task, the NGC API key is required.

1. Use SSH to access the instance and enter the NGC API key as requested. Once you have logged in, the instance is registered with NGC to pull down an updated container image.

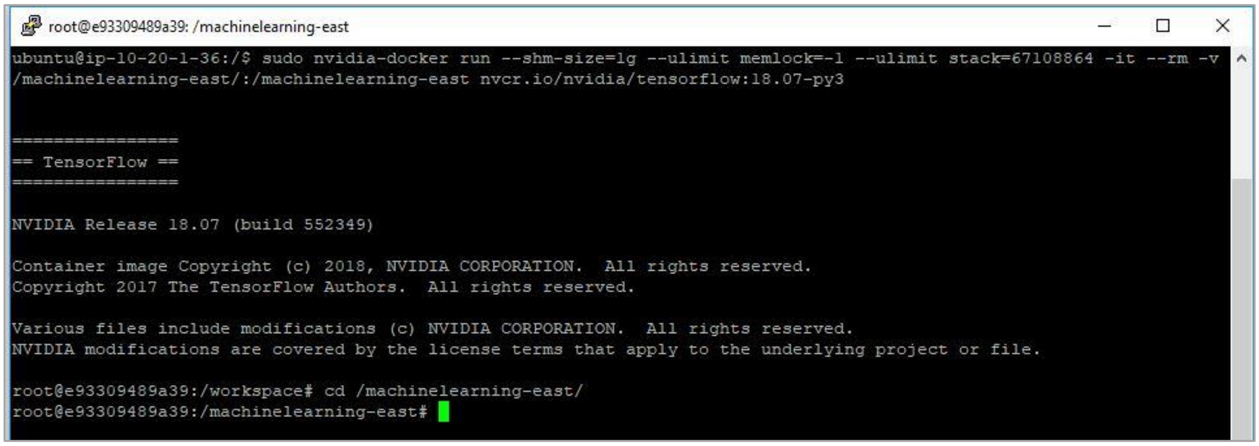


2. Before configuring the container, mount the Cloud Volumes NFS mount so it is accessible from inside the container. The mounting instructions (including the mount point) are available on the Cloud Volumes portal under the volume created earlier in this technical report.
3. Now that the Cloud Volumes NFS volume is mounted, pull the latest Docker container; for this validation, version 18.07-py3 is used. The pull takes about 2 minutes after the command is executed.

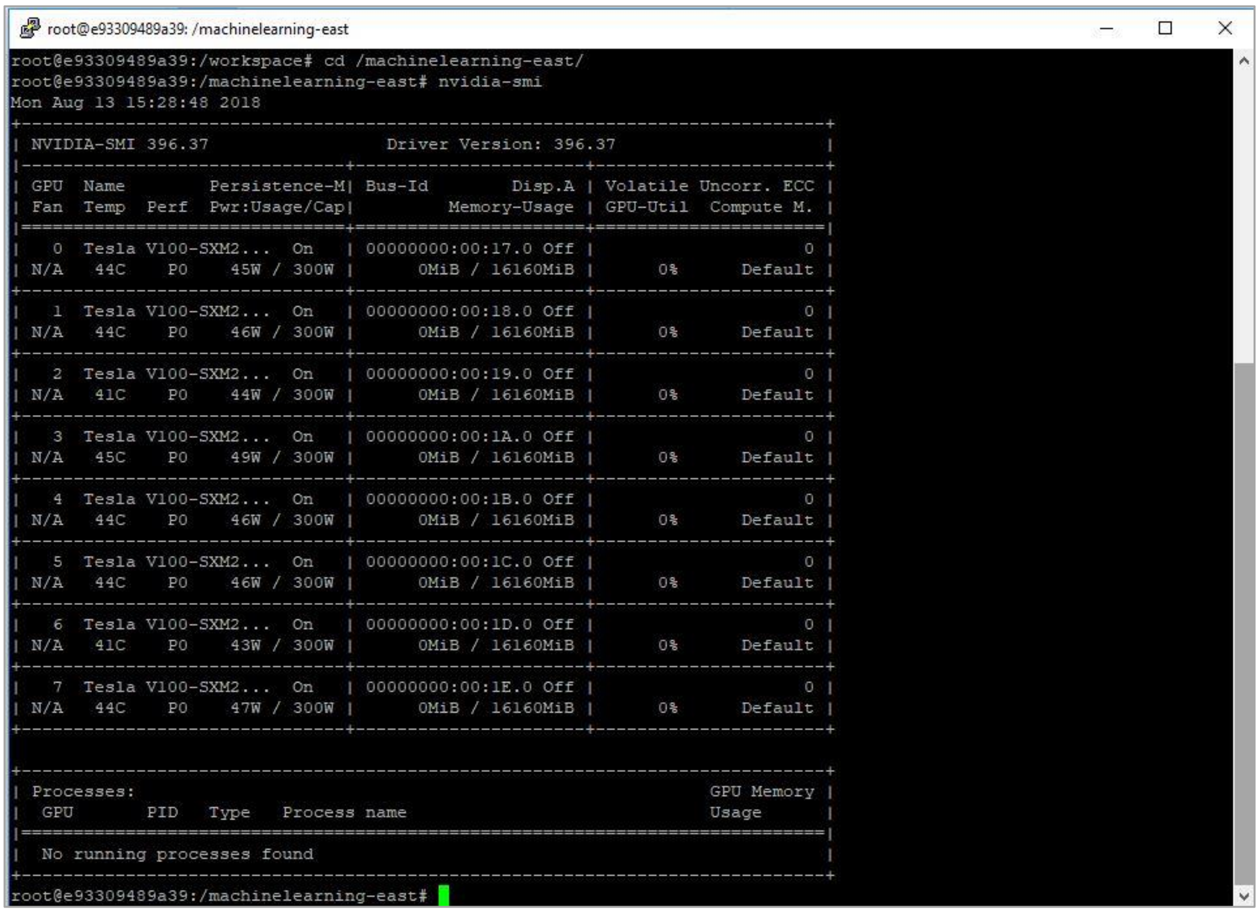


4. Run the Docker image by using the recommendation found on NGC, as shown in the following command. Verify that the Cloud Volumes directory is accessible.

```
sudo nvidia-docker run --shm-size=1g --ulimit memlock=-1 --ulimit stack=67108864 -it --rm -v
/machinelearning-east/:/machinelearning-east nvcr.io/nvidia/tensorflow:18.07-py3
```

5. At this point, the container is ready with all the NVIDIA drivers and tools. Verify that the GPUs are visible from the container by running the `nvidia-smi` command.



Now that the container and the Cloud Volumes mounts are ready, there are couple of options for synchronizing the ImageNet dataset from the on-premises environment or any other location:

- **NetApp Cloud Sync.** This validation uses this option. However, the setup details are beyond the scope of this technical report. Refer to the [NetApp Cloud Documentation]() portal for detailed instructions.
- **SCP.** Since the Amazon EC2 instance supports SSH, it is also possible to use SCP.

The ImageNet dataset is available on the www.image-net.org website, which includes detailed instructions on how to obtain the latest image. The ImageNet dataset is roughly 140GB; it will take a little time to move it to the cloud using NetApp Cloud Sync or SCP.

5.3 Tensor Flow Configuration

The following convolutional neural network (CNN) models with varying degrees of compute and storage complexities were used to demonstrate training rates:

- **ResNet-152** is considered to be the most accurate training model.
- **ResNet-50** delivers better accuracy than AlexNet with faster processing times.
- **VGG16** produces the highest inter-GPU communication.
- **Inception-v3** is another common TensorFlow model.

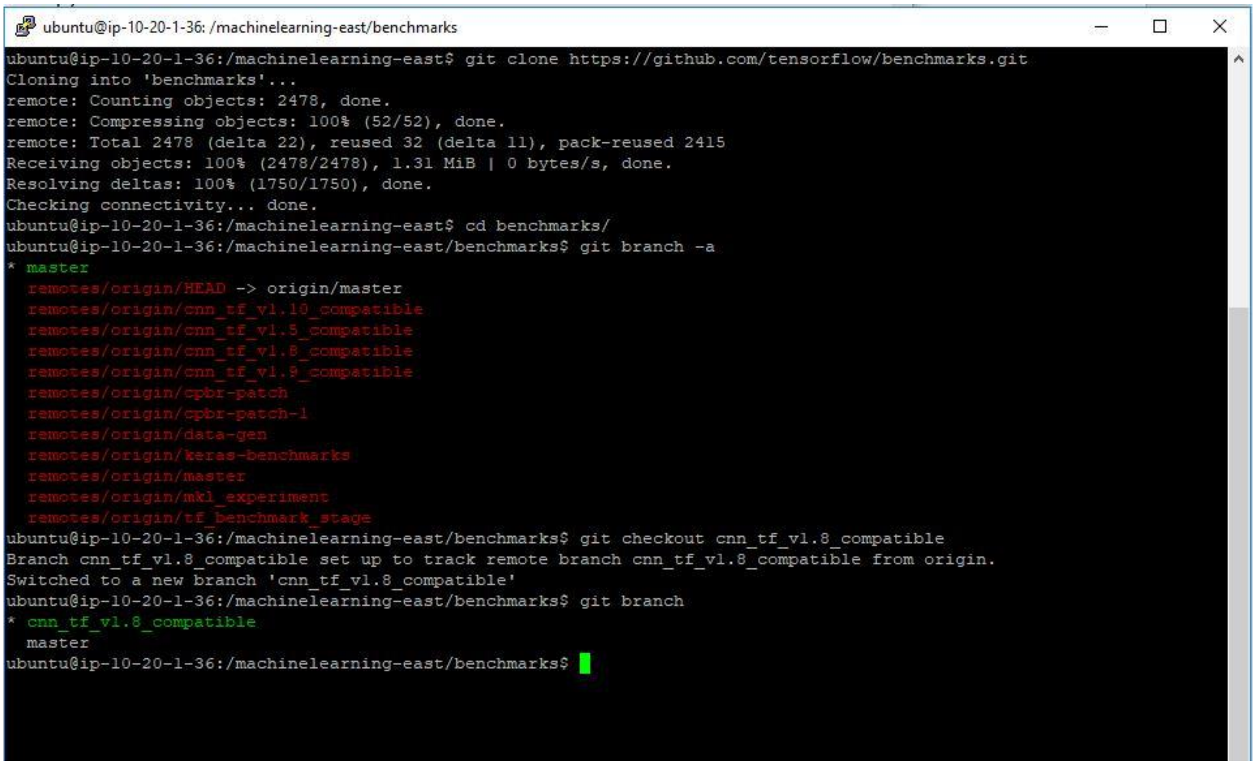
Each of these models was tested with software configurations to study the effects of each option on performance:

- We tested each model with the ImageNet reference dataset.
- We used ImageNet data with distortion disabled to reduce the overhead of CPU processing before copying data into GPU memory.
- We tested each model using Tensor cores.
- We tested each DL model with various batch sizes (results with batch size of 64 are shown in this paper):
  - 64, 128, and 256 for ResNet-50
  - 64 and 128 for all other models
- Each model was tested with eight GPUs for all tests.
- Inference was run using all the models with the batch size of 64 (with 8 GPUs, Tensor cores), and with the ImageNet dataset.

All performance metrics were gathered after at least a single epoch. The focus was to push the performance of the GPUs to the maximum to demonstrate the capabilities of the Cloud Volumes Service.

The following steps demonstrate how to obtain the latest TensorFlow release from GitHub, the scripts are downloaded to the Cloud Volumes mount previously created.

1. Clone the tensorflow repository from GitHub.
2. From the instant CLI clone, branch to the `machinelearning-east` directory. Make sure that you're not in the container and run the `exist` command to drop out of the container.



3. At this point, TensorFlow benchmark scripts and the ImageNet dataset should be on your volume. The next step is to execute the commands to begin the training.

5.4 Tools Used

The tools used to monitor the performance and utilization are readily available on the GPU instance and the AWS portal.

- NVIDIA-SMI
- NFSIOSTAT
- Python
- Amazon EC2 monitoring (Available from the AWS Console).

The tools were executed by a Python script to capture the output during training and inference. The results of each run were captured automatically.

6 Test Methodology

The goal of these tests is demonstrating NetApp Cloud Volumes Service capabilities to meet machine learning's workload throughput, IOPS, and capacity demands. The following sections describe each type of training completed and the output from TensorFlow, GPU utilization, and NFS I/O (throughput) status.

We executed the commands by using a Python script to capture the raw output from each training. The script simply executed the TensorFlow benchmarks tool, NVIDIA-SMI, and `nfsiostat` commands.

Each test was executed from inside the NVIDIA container. The commands to run the docker container and the TensorFlow commands are listed in section 6.1, "Test Sequence." Each test includes two changes: batch size and training model.

6.1 Test Sequence

As described previously, we conducted various tests to assess the general operation and performance of this solution. This section contains highlights of the compute and storage performance data that was collected during those tests.

Model training performance is measured as images per second.

- Storage performance is measured using throughput (MBps) and latency (µs). The storage system CPU was also captured to evaluate the remaining performance capacity on the storage system.
- Each system was tested with multiple batch sizes. Larger batch sizes increase the overall training throughput. Results with a batch size of 64 for each model is included in this paper.
  - ResNet-50, ResNet-152, Inception-v3, and VGG16 tests used a batch size of 64.

Each test was executed in the following sequence:

1. Run the Docker container.

```
nvidia-docker run --shm-size=1g --ulimit memlock=-1 --ulimit stack=67108864 -it --rm -v
/machinelearning-east/:machinelearning-east/ nvcr.io/nvidia/tensorflow:18.07-py3
```

2. Change directory into docker container.
3. Execute the TensorFlow commands with the following options:

```
python tf_cnn_benchmarks.py   --num_gpus=8 \
                              --device=gpu \
                              --data_format=NCHW \
                              --use_fp16=True \
                              --batch_size=64 \
                              --num_batches=500 \
                              --data_name=imagenet \
                              --use_datasets=True \
                              --data_dir=/machine-learning-east/tfrecords \
                              --model=resnet50_v2 \
                              --nodistortions \
                              --variable_update=independent \
                              --datasets_use_prefetch=True \
                              --forward_only=False \
                              --datasets_prefetch_buffer_size=1
```

6.2 Test Results

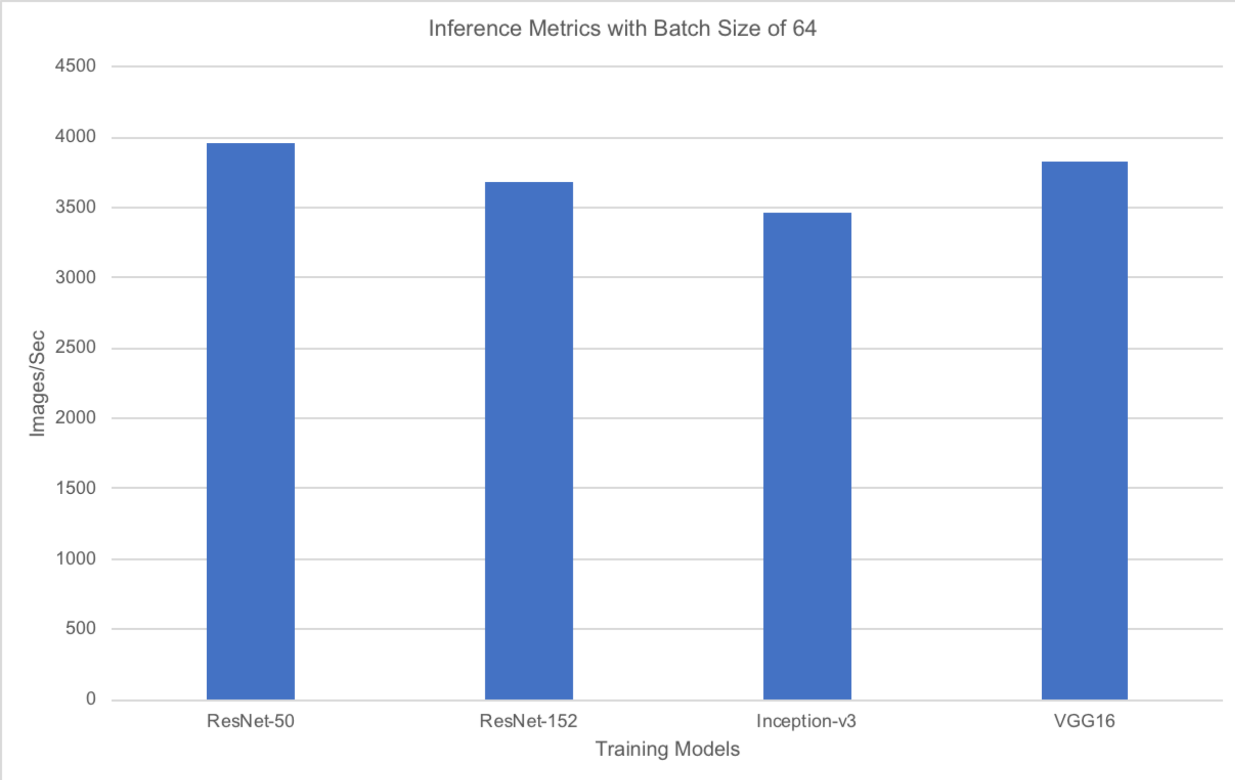
Figure 2 shows the maximum number of training images per second that was achieved with each of the models. Tensor cores were used for maximum performance.

Figure 2) Training metrics for all models.

Inferencing is the process of deploying the DL model to assess a new set of objects and making predictions with a level of accuracy similar to that observed during the training phases. In an application with an image dataset, the goal of inferencing is to classify the input images and to respond to the requesters as quickly as possible. In addition to achieving high throughput, minimizing latency becomes important.

NetApp Cloud Volumes Service was used to demonstrate inferencing and to measure throughput metrics in this phase. Figure 3 shows the number of images that can be processed per second during inferencing. This test compares the throughput that was achieved for each of the models (used eight GPUs, Tensor cores, ImageNet data).

Figure 3) Inference metrics for all models.



The next set of data demonstrates the ability of Cloud Volumes Service to meet the requirements of the eight NVIDIA Tesla V100 GPUs under a full load. Figure 4 shows the GPU utilizations and the storage bandwidth that was generated when we ran training with each model using eight GPUs. As the graph shows, the storage bandwidth starts off very high as the initial data is read from storage into the TensorFlow pipeline cache, then it drops gradually as a larger portion of the dataset becomes available in the GPU local memory over time.

Note: Amazon EC2 instances have TCP limits of 5Gbps per session. An NFS mount from an EC2 has a single session and a maximum theoretical read or write of 5Gbps. When training keep this limit in mind.

Figure 4) GPU utilization and storage bandwidth.

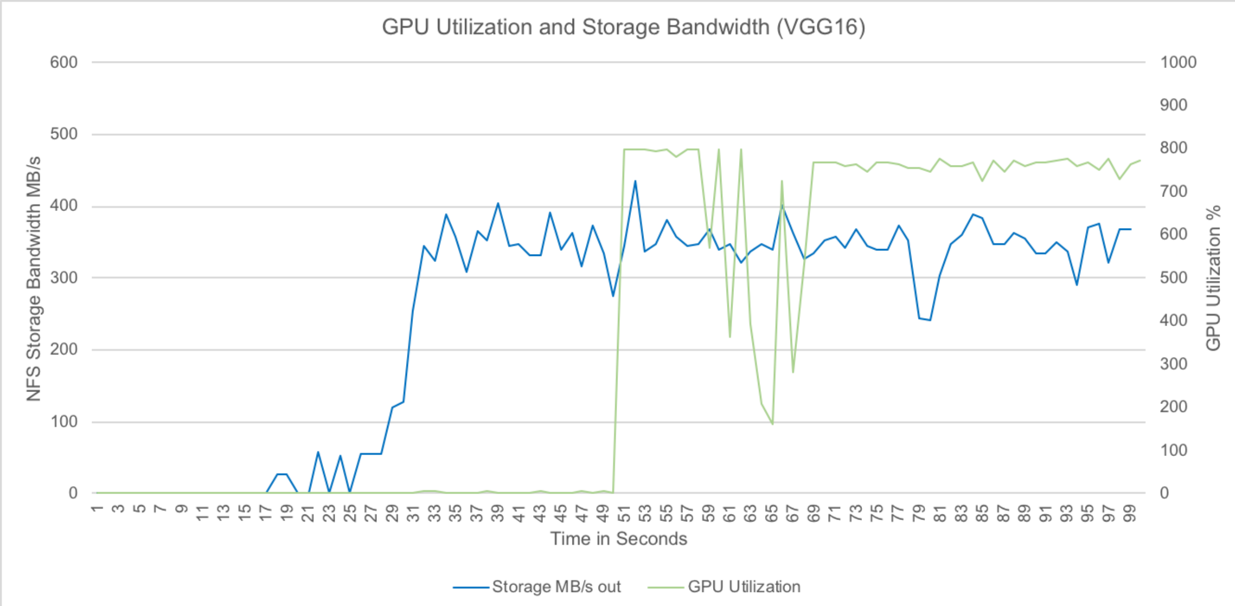


Figure 4 summarizes the results of GPU and the storage service to demonstrate the interaction between GPUs and the Cloud Volume Service when tested with the VGG16 model. The X-axis represents the run time of the TensorFlow benchmark, the left-Y-axis represents the overall network traffic from the storage to GPUs, and the right-Y-axis shows the overall all GPUs utilizations (the maximum value is 800% due to eight GPUs). From the results, we can conclude that the network traffic is lower than the theoretical throughput (~512MBps). This is due to the fact that the GPUs do not need more than this amount of traffic. It can be confirmed by the overall GPU utilizations, which has been pushed to their limitation (~95% utilization per GPU). This figure also shows that our storage service has enough head room to support more than 8 GPUs to execute simultaneously.

7 Conclusion

The cloud provides a rich software ecosystem to develop, deploy, and test AI/DL applications. It is common for small to large organizations to get started on the cloud by using GPUs for model training. GPUs are priced resources and need to be maintained at high utilizations in order to reduce training times, allow for faster experimentation, and as a result, minimize the cost of usage. A high-performance, easy to use file system that will prevent GPUs from waiting for data is imperative in accelerating model training on the cloud and to optimize cost. NetApp Cloud Volumes Service fits perfectly for DL model training on the cloud. It is a fully managed, cloud-native file storage service with an intuitive user interface (and CLI) that will enable you to get started quickly and deliver the high performance required by data intensive AI / DL applications. The Cloud Volumes Service, uniquely optimized to accelerate model training on the cloud, thus reducing costs is ideally suited for organizations starting their AI journey on the cloud as well as the ones with existing cloud deployments for AI.

Where to Find Additional Information

To learn more about the information that is described in this document, review to the following documents and/or websites:

- NetApp Cloud Volumes Services
  https://cloud.netapp.com/cloud-volumes-service
- NetApp product documentation
  http://docs.netapp.com

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

TR-4718-1018

NetApp