Microsoft

Run MPI workloads

with Azure Batch and Azure NetApp Files

By Jon Shelley
AzureCAT

March 2020

Contents

Authored by Jon Shelley. Edited by Nanette Ray. Reviewed by Azure Global CAT.

Overview

Message Passing Interface (MPI) workloads are a significant part of traditional high-performance computing (HPC) workloads. Azure provides MPI support for Linux virtual machines (VMs) and Azure Batch, a service that makes it easy to run large-scale jobs in parallel. You tell Batch what kind of VMs you need, how to configure them, and the jobs and tasks to run. The service takes care of the rest.

This step-by-step guide helps you get started running MPI workloads in Azure Batch when using Azure NetApp Files (ANF) as the shared file system. ANF provides several benefits—for example, you can use Azure Marketplace images for your VMs and then store your applications and results on ANF. This approach allows for faster startup times of the nodes, reduces the complexity in job submission files, and lowers the time spent by the compute nodes moving data around at the start and end of the batch task.

The following steps show you how to set up an Azure Batch account in your user subscription with Azure NetApp Files using a CentOS 7.7 HPC image from the Azure Marketplace. This example creates a jump host and deploys an ANF file system for the batch compute nodes. Then you can submit a simple HPC-X MPI task that runs an OSU latency and bidirectional bandwidth test on two batch compute nodes. After verifying the output, the last step is to clean up the resource groups so you don't incur additional charges.

To deploy the infrastructure shown in Figure 1, follow the steps in this guide.

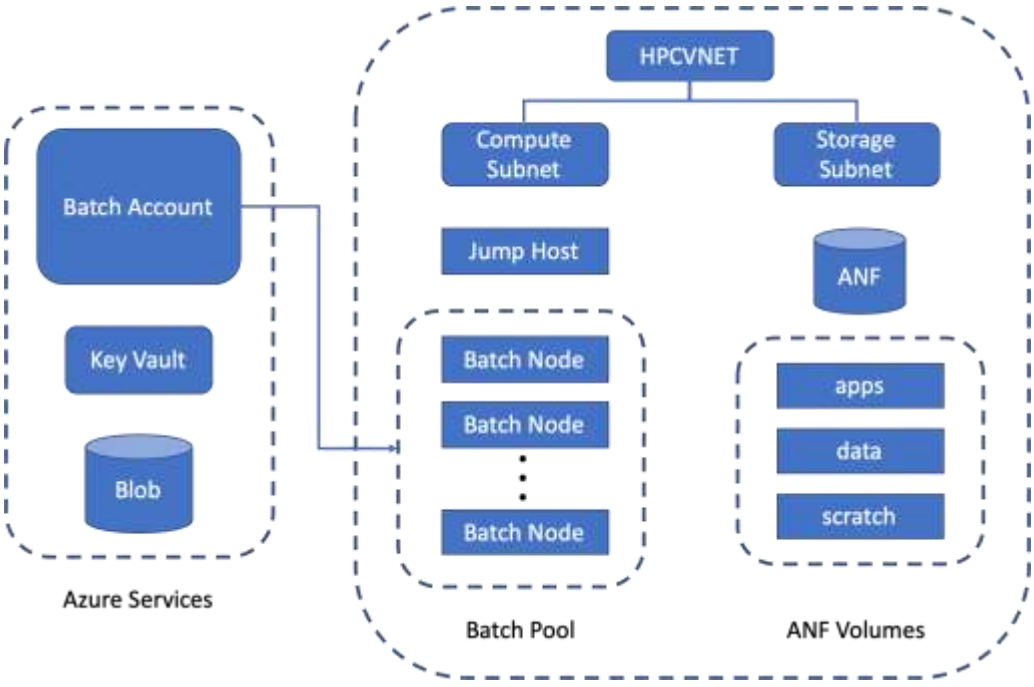


Figure 1. Deployment showing Azure Batch and Azure NetApp Files

Get started

To get started, set up these requirements:

1. Install the Azure CLI on your host (or, if you prefer, use Azure Cloud Shell from the Azure portal).
2. If it is not present, install Git on your host:

```
sudo yum install -y git
```

3. Verify that you have SSH keys generated on your system. To create your jump box, the script needs `~/.ssh/id_rsa.pub`. If you need to generate an SSH key, run the following command on your system:

```
ssh-keygen -t rsa -f ~/.ssh/id_rsa -q -P ""
```

4. Request and receive access to ANF using the Azure NetApp Files request form. The request is typically turned around in a day. When approved, you can proceed to set up the batch environment.
5. Verify your quota for the VM instance type in the desired region. To see the subscription quota in a given region, run the following commands, using your Azure subscription ID and specifying the region you want, such as South Central US.

```
az account set -s <Your Subscription ID>
az vm list-usage --location "South Central US" -o table
```

Set up the batch environment with ANF

To deploy the infrastructure, it is recommended that you use Azure Cloud Shell, but you can use your own Linux system running Red Hat Enterprise Linux (RHEL) 7 or CentOS 7 with Azure CLI version 2.0.80 or later.

1. To get the scripts needed to set up the batch environment and to run the MPI task in batch, go to Cloud Shell and run the following commands:

```
cd ~/
git clone https://github.com/JonShelley/azure.git
```

2. Change to the `anf` example directory:

```
cd ~/azure/blog-files/batch/examples/anf
```

3. Set the variables in the top section of setup_batch_with_anf.sh to the desired values for your subscription. The variables commonly changed are: `region, sub_id, vnet_2_octets` (the default is fine), `batch_name`, `pool_id` (the default is OK if you are using HC VM instances), and `pool_vm_size`.

```
# General variables
region=westus2
batch_rg=ex-batch-${region}
infra_rg=ex-infra-${region}
```

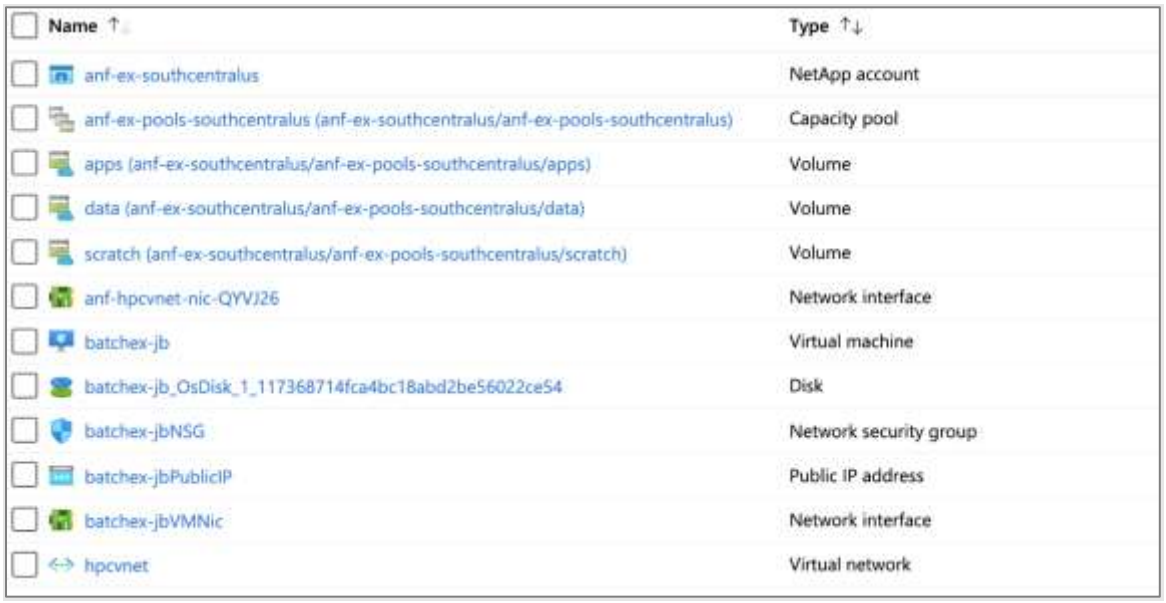
```
sub_id=<Replace with subscription id>
vnet_2_octets="10.2"

# Batch variables
# Note: batch_name and storage_account_name need to be unique and are
limited to 3-24 lowercase alphanumeric characters
# I recommend that you add your initials or 3 random letters at the end of
batch_name.
batch_name=batchex
storage_account_name=${batch_name}${region}
storage_blob=batch
pool_id=HC
pool_vm_size=Standard_HC44rs

# ANF variables
anf_account_name="anf-ex-${region}"
anf_pool_name="anf-ex-pools-${region}"
service_lvl="Premium"
apps_path=ex-apps
data_path=ex-data
scratch_path=ex-scratch
```

4. When the variables are set, run the following and wait for it to complete (about eight minutes) before going on to the next step.

```
bash setup_batch_with_anf.sh
```

5. After setup_batch_with_anf.sh completes, verify that it deployed correctly by scrolling through the terminal output for errors and reviewing the two resource groups in the Azure portal (`batch_rg=ex-batch-${region}` and `infra_rg=ex-infra-${region}`). The `batch_rg` group should look like this:

And `infra_rg` should look like this:



Name	Type
anf-ex-southcentralus	NetApp account
anf-ex-pools-southcentralus (anf-ex-southcentralus/anf-ex-pools-southcentralus)	Capacity pool
apps (anf-ex-southcentralus/anf-ex-pools-southcentralus/apps)	Volume
data (anf-ex-southcentralus/anf-ex-pools-southcentralus/data)	Volume
scratch (anf-ex-southcentralus/anf-ex-pools-southcentralus/scratch)	Volume
anf-hpcvnet-nic-QYVJ26	Network interface
batchex-jb	Virtual machine
batchex-jb_OsDisk_1_117368714fca4bc18abd2be56022ce54	Disk
batchex-jbNSG	Network security group
batchex-jbPublicIP	Public IP address
batchex-jbVMNic	Network interface
hpcvnet	Virtual network

6. Sign in to the newly created jump host from Azure Cloud Shell (or from the host that you used to run setup_batch_with_anf.sh). It is listed in the `$infra_rg` resource group as `batchex-jb` or something similar.

```
ssh hpcuser@ip-address-to-batchex-jb
cd ~/
git clone https://github.com/JonShelley/azure.git
cd ~/azure/blog-files/batch/tutorials/anf_mpi_tests
sudo chmod 777 /scratch
mkdir -p /scratch/ex1
cp run_hpcx_mpi_tests.sh /scratch/ex1/.
chmod 755 /scratch/ex1/run_hpcx_mpi_tests.sh
```

Optional

To create jobs and tasks from the jump host, follow the Azure CLI installation instructions. When complete, you will need the values used for `sub_id`, `batch_name`, and `batch_rg` that you used in the setup_batch_with_anf.sh file.

```
sub_id=<Replace with subscription id>
az account set -s $sub_id
az login
az batch account login --name $batch_name --resource-group $batch_rg
```

Create a job and a task

Using Azure Cloud Shell (or your host or the jump host if it is set up), change to the **anf_mpi_tests** directory:

```
cd ~/azure/blog-files/batch/tutorials/anf_mpi_tests
```

Edit the `pool_id` variable at the top of the submit_batch_tasks.sh file to match the value used in setup_batch_with_anf.sh. You may also need to edit the `ppn` variable to match the number of cores on the VM if you are not using a HC44rs VM instance.

After making these changes, you are now ready to submit the job and task:

```
bash submit_batch_tasks.sh
```

Review results

To see the results in the Azure portal:

1. Navigate to the Azure Batch account (for example, **batchex**) in the resource group defined by the `batch_rg` variable.
2. On the menu, select the **Jobs** link and then select the job that you created when you ran submit_batch_tasks.sh (for example, `myjobs-${pool_id}`).
3. Select the `test-$DATE` (that is, `test-XXXXXXXX-XXXXXX-XXXXXX`).
4. To see the results of your run, select the **stdout.txt** file. If all worked as expected, you should see something like the following.

```
b256727833514f23afc0fd3b26f14b5b000002
Src: 10.232.2.4
Dst: 10.232.2.6
# OSU MPI Latency Test v5.3.2
# Size          Latency (us)
0                       1.95
1                       1.97
2                       1.94
4                       1.95
8                       1.95
16                      1.96
.
.
.
1048576               158.73
2097152               300.02
4194304               581.99
# OSU MPI Bi-Directional Bandwidth Test v5.3.2
```

```
# Size      Bandwidth (MB/s)
1                      3.80
2                      7.84
4                     15.88
8                     31.42
16                    63.39
.
.
.
1048576            10600.76
2097152            10643.48
4194304            10600.19
```

Clean up

After you are finished, you can clean up the environment. To remove everything that you created in this example, run the following commands, replacing `batch_rg` and `infra_rg` with the values you used in the setup_batch_with_anf.sh file:

```
az group delete -n $batch_rg
az group delete -n $infra_rg
```

Learn more

For more information, check out the following resources:

- Get an overview of HPC on Azure
- Review the Azure Storage Performance and Scalability Checklist
- Read the HPC: Oil and Gas in Azure blog
- Check out Azure HPC sample scripts on GitHub