Run Star-CCM+ in an Azure HPC Cluster

By Jon Shelley, Paul Edwards, and Kanchan Mehrotra
Azure Global CAT

February 2020

Contents

Authored by Jon Shelley, Paul Edwards, and Kanchan Mehrotra from AzureCAT. Edited by Ed Price.

Overview

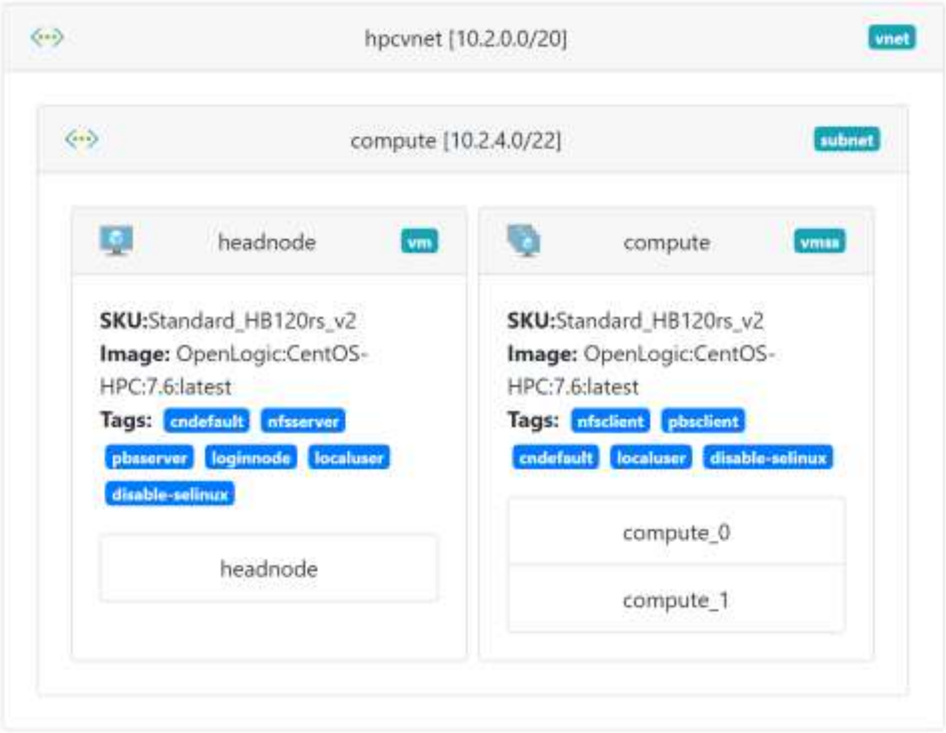
The goal of this document is to demonstrate how you can quickly set up and benchmark a high-performance computing (HPC) cluster in the Microsoft Azure environment as a proof of concept for running Star-CCM+ version 14 from CD-adapco, a Siemens Business. This example shows you how to create the cluster, provides benchmarking options, and installs the Star-CCM+ simulation software so you can test its performance on Azure.

This document elaborates on the example GitHub project, Install and run Star-CCM+ benchmarks using azurehpc cluster. You can use this document for more details as you run through the example.

NOTE: To run this example, you must have a valid Power-on-Demand (POD) license for Star-CCM+. For details, see the Licensing section later in this article. Your Azure subscription must also have sufficient quota to run this workload, which requires H-Series virtual machines (VMs). See this image for details on the resources deployed by the example, but if you want to run on larger scale, please plan accordingly.

Architecture

This example creates a simple architecture you can use to demonstrate Star-CCM+ running in a high-performance cluster that's based on virtual machine scale sets. This architecture is intended for test purposes.



headnode:

SKU: Standard_HB120rs_v2
Image: OpenLogic:CentOS-HPC:7.6:latest
Tags: cndefault nfsserver pbsserver loginnode localuser disable-selinux

compute:

SKU: Standard_HB120rs_v2
Image: OpenLogic:CentOS-HPC:7.6:latest
Tags: nfsclient pbsclient cndefault localuser disable-selinux

Figure 1. High-performance cluster for running Star-CCM+ in Azure

After creating the cluster, the script authenticates all the nodes in the cluster and creates a common share directory used by each node. One VM is designated as the jumpbox and head node. A public IP is assigned to the jumpbox NIC, and port 22 is left open for SSH.

Storage

For this architecture, the following NFS file shares are created:

- **/apps** and **/data** are created from the head node's operating system disk and shared with the scale set nodes.
- **/mnt/resource/scratch** is created on the temp disk of the head node. The temp disk is a physically attached disk that typically provides faster performance on each of the nodes. If the head node is deallocated, the data in this directory will be lost. This is mounted on all the nodes to **/scratch** and a symbolic link is created on the head node to match this.

Authentication

The VMs that are part of the scale set, which is inside a private network, are accessible only from the jumpbox by using a private IP address. You can access one of these nodes from the jumpbox.

Star-CCM+

The install_starccm.sh file contains the scripting commands to install Star-CCM+. You can use the environment variables to change the installation directory, temporary directory, and the directory where the Star-CCM+ installer is located (this must be present when the installer runs).

The add_reqs.sh file contains the scripting commands to install the prerequisite packages on the compute nodes. You will need to run this after deploying the cluster (or add as one of the installation steps in the cluster deployment).

The run_case.pbs file contains the scripting commands for Star-CCM+.

Licensing

During deployment, you are asked for the POD key associated with your Star-CCM+ license:

- Set your POD key (environment variable name: PODKEY) before running run_case.pbs.

Deploy the cluster and run Star-CCM+ benchmark

Below we present two options that you can use to deploy the cluster for running the Star-CCM+ benchmark, based on your preference.

Option 1: Simple PBS cluster using the AzureHPC tools

You can deploy a simple PBS cluster using the AzureHPC tools. The detailed steps are available in the example azurehpc project on GitHub.

Deploy a cluster and install Star-CCM+

You can install and run Star-CCM+ using the following steps (also available azurehpc readme).

1. Copy the apps directory to the cluster. The **azhpc-scp** command can be used to do this:

   ```
   azhpc-scp -r $azhpc_dir/apps hpcuser@headnode:.
   ```

   The **libXt** package is required on the compute nodes. This can be installed with the **add_reqs.sh** script that is provided here (alternatively add it to the install steps for the cluster, but with **azurehpc**):

   ```
   azhpc-run -u hpcuser -n compute apps/starccm/scripts/add_reqs.sh
   ```

   You must first obtain the **starccm** installer and copy it to the cluster. If it is available on the local machine you can copy as follows:

   ```
   azhpc-scp STAR-CCM+14.04.013_01_linux-x86_64-2.12_gnu7.1.zip
   hpcuser@headnode:/mnt/resource/.
   ```

2. Run the install script:

   ```
   azhpc-run -u hpcuser apps/starccm/install_starccm.sh
   ```

Run Star-CCM+ benchmark

You must copy the benchmark to the cluster. The default in the run script is **civil**. You can copy the **sim** file as follows:

```
azhpc-scp civil.sim hpcuser@headnode:/data/civil.sim
```

Now, connect to the head node:

```
azhpc-connect -u hpcuser headnode
```

The **run_case.pbs** script is ready to use. Environment variables can be passed to the PBS job with the **-v** flag.

Submit a job as follows (remember to substitute your PoD key value):

```
qsub -l select=2:ncpus=60:mpiprocs=60:place=scatter:excl \
    -v PODKEY=#INSERT_POD_KEY# \
    apps/starccm/run_case.pbs
```

NOTE: Multiple environment variables can be set if they are separated by commas. For example:

```
-v VAR1=x,VAR2=y
```

The output will be in the working directory where it was submitted.

Option 2: Simple PBS cluster using Azure CycleCloud

You can deploy a simple PBS cluster using Azure CycleCloud.

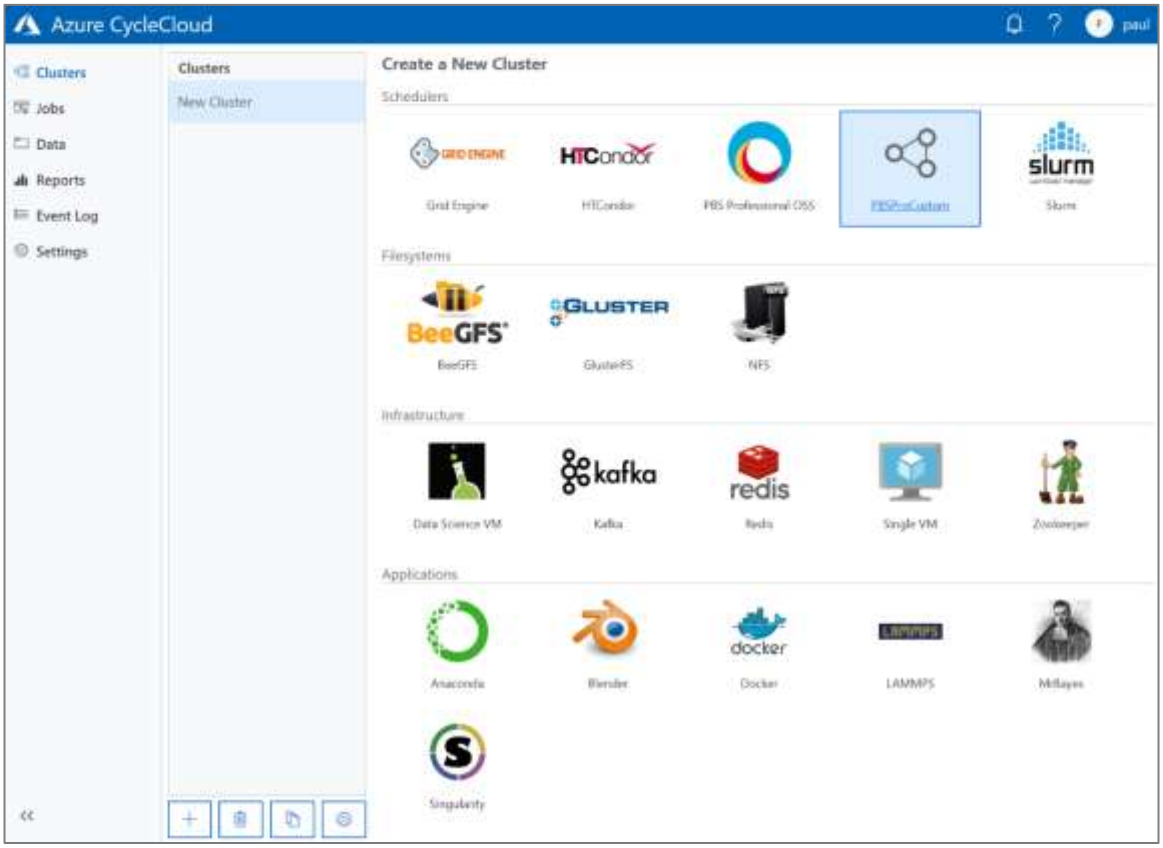
Deploy a cluster

1. Install Azure CycleCloud using the Azure CycleCloud Quickstarts or the **azurehpc** scripts in the GitHub project.
2. Use the template in our azurehpc/examples/cyclecloud_simple_pbs project to deploy PBSPro. Azure CycleCloud comes with a default template for deploying PBSPro, but we have made some tweaks to add some mount points needed for this example.

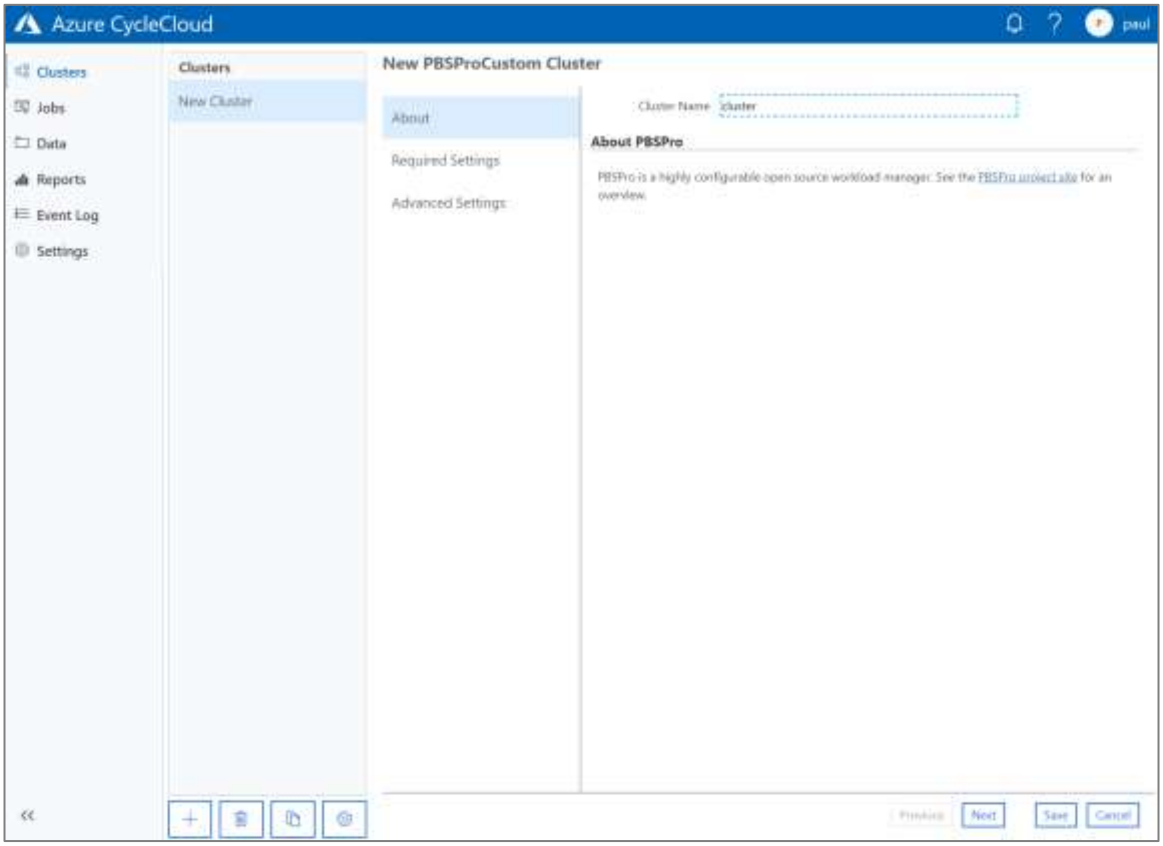
NOTE: Before importing the template using the details provided in the azurehpc/examples/cyclecloud_simple_pbs project, you need to update the default project spec to install the Star-CCM+ dependency. Add the following in your **specs/default/cluster-init/scripts/01_install_packages.sh** file:

```
#!/bin/bash
yum -y install libXt
```

3. Go to the Azure CycleCloud web page and create a new cluster.



4. Fill in the name of your cluster.



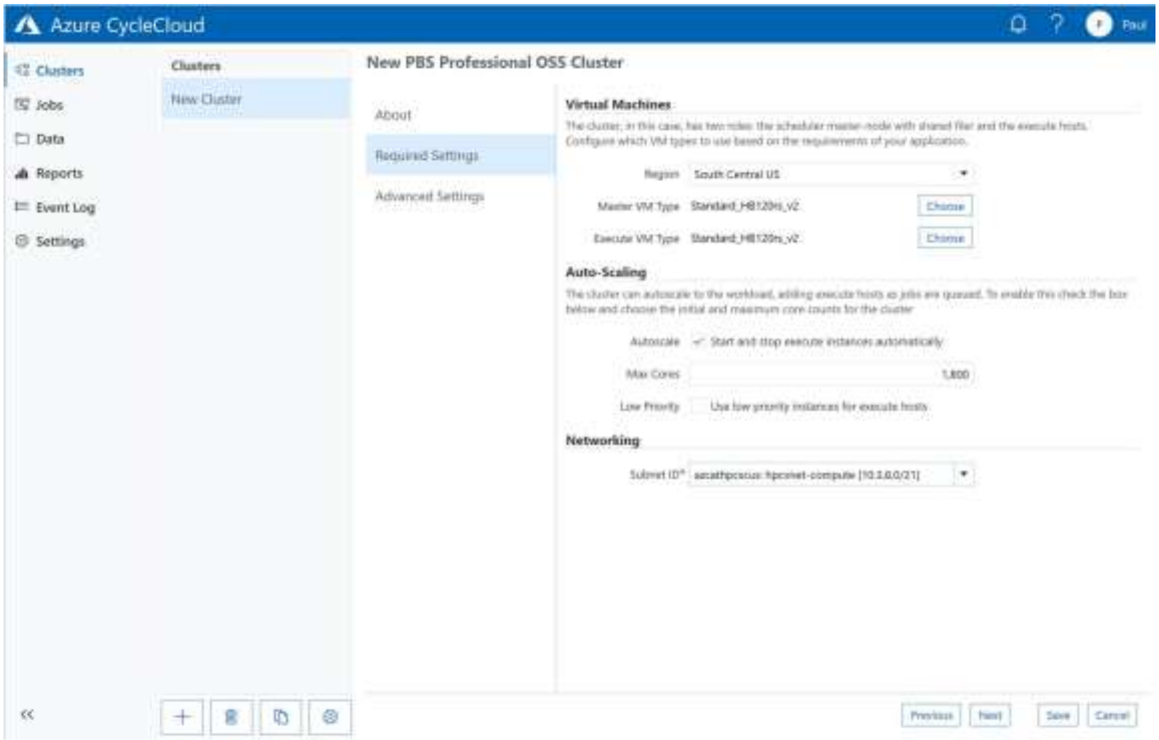
New PBSProCustom Cluster

Cluster Name cluster

About PBSPro

PBSPro is a highly configurable open source workload manager. See the PBSPro project site for an overview.

5. In Required Settings, set the node type to **Standard_HB120rs_v2**. (Or select HC44rs if you want to use an Intel-based VM). Choose your maximum number of cores, and select the virtual network you want to use.



New PBS Professional OSS Cluster

Virtual Machines

The cluster, in this case, has two roles: the scheduler master-node with shared filer and the execute hosts. Configure which VM types to use based on the requirements of your application.

Region South Central US

Master VM Type Standard_HB120rs_v2

Execute VM Type Standard_HB120rs_v2

Auto-Scaling

The cluster can autoscale to the workload, adding execute hosts as jobs are queued. To enable this check the box below and choose the initial and maximum core counts for the cluster

Autoscale ✓ Start and stop execute instances automatically

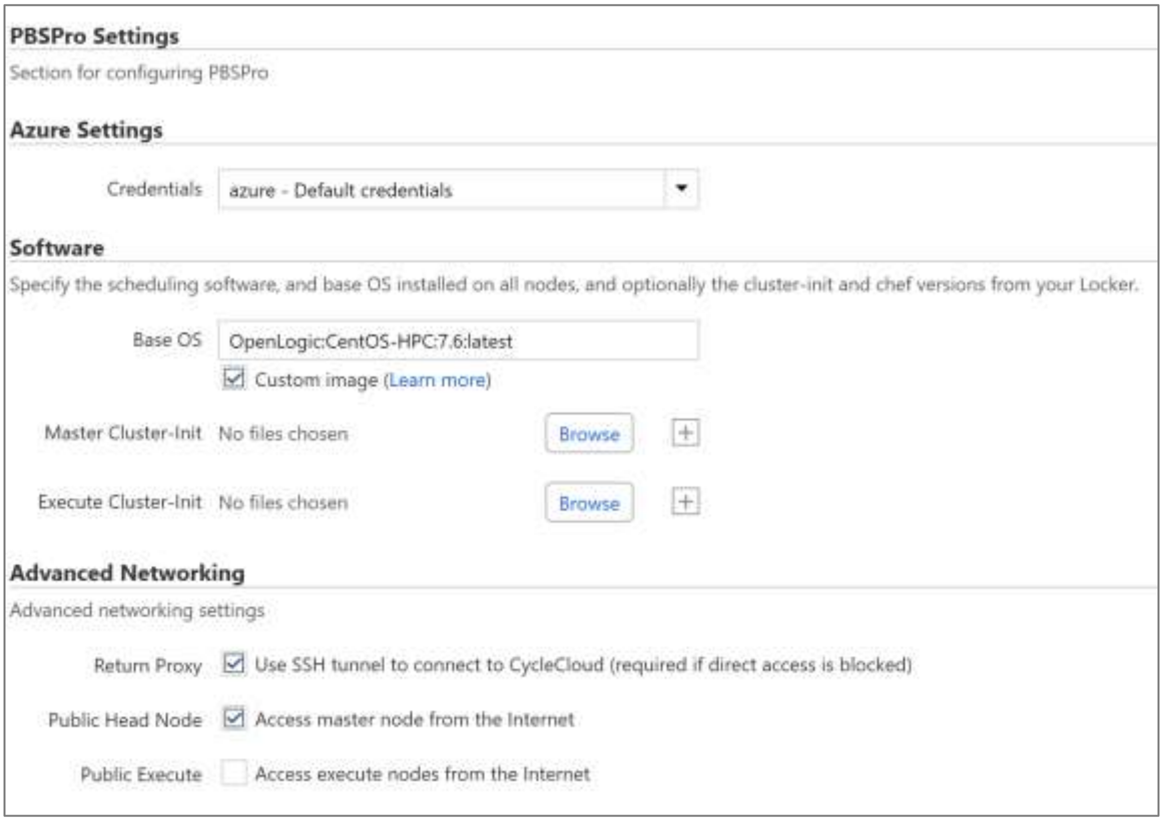
Max Cores 1,800

Low Priority □ Use low priority instances for execute hosts

Networking

Subnet ID* azcathpcscus: hpcvnet-compute [10.2.0.0/21]

6. In the **Advanced Settings** tab, set the **Base OS** to **OpenLogic:CentOS-HPC:7.6:latest** and select **Custom image**.



PBSPro Settings

Section for configuring PBSPro

Azure Settings

Credentials: azure - Default credentials

Software

Specify the scheduling software, and base OS installed on all nodes, and optionally the cluster-init and chef versions from your Locker.

Base OS: OpenLogic:CentOS-HPC:7.6:latest

☑ Custom image (Learn more)

Master Cluster-Init: No files chosen

Execute Cluster-Init: No files chosen

Advanced Networking

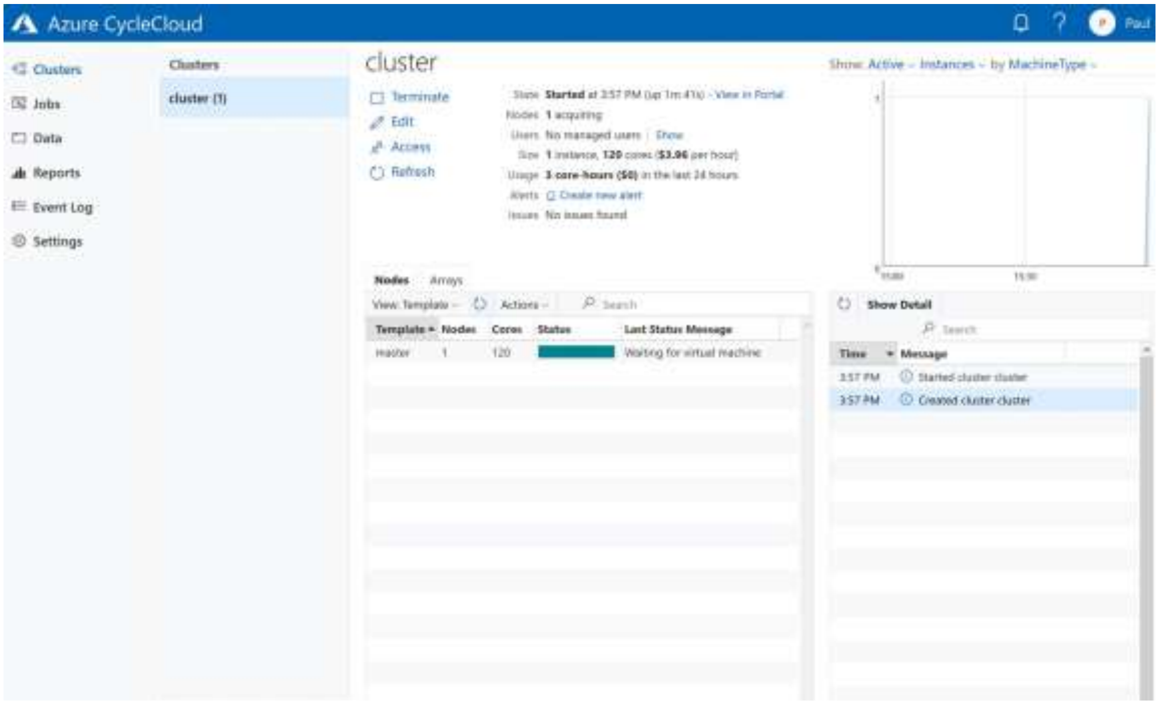
Advanced networking settings

Return Proxy: ☑ Use SSH tunnel to connect to CycleCloud (required if direct access is blocked)

Public Head Node: ☑ Access master node from the Internet

Public Execute: □ Access execute nodes from the Internet

7. Save and start the cluster. The web page will show progress as the head node for the cluster starts.



cluster

State: **Started** at 3:57 PM (up 1m 41s) - View in Portal
Nodes: **1** acquiring
Users: No managed users | Show
Size: **1** instance, **120** cores (**$3.96** per hour)
Usage: **3 core-hours ($0)** in the last 24 hours
Alerts: Create new alert
Issues: No issues found

Template	Nodes	Cores	Status	Last Status Message
master	1	120		Waiting for virtual machine

Time	Message
3:57 PM	Started cluster cluster
3:57 PM	Created cluster cluster

8. Log on to the head node of the cluster:

   ```
   $ cyclecloud connect master -c cluster
   ```

9. Set up Star-CCM+ and submit the jobs to the PBS cluster that was created using the following detailed steps for installing Star-CCM+ and running the benchmark (also available in the azurehpc project).

Install Star-CCM+

1. Copy the /apps/starccm folder to the head node.
2. Get the Star-CCM+ installer, **STAR-CCM+14.04.013_01_linux-x86_64-2.12_gnu7.1.zip**, and copy it to the cluster to **/mnt/resource**.
3. Run the following to install Star-CCM+ on the cluster:

   ```
   export APP_INSTALL_DIR=/scratch
   apps/starccm/install_starccm.sh
   ```

Run Star-CCM+ benchmark

1. Copy the benchmark to the cluster under **/scratch**. The default in the run script is **civil**. You can copy the **sim** file to the head node.

   The **run_case.pbs** script is ready to use. Environment variables can be passed to the PBS job with the **-v** flag.

2. Submit a job as follows (remembering to substitute your PoD key value):

   ```
   qsub -l select=2:ncpus=60:mpiprocs=60:place=scatter:excl \
   -v PODKEY=#INSERT_POD_KEY#,APP_INSTALL_DIR=/scratch,DATA_DIR=/scratch \
   apps/starccm/run_case.pbs
   ```

NOTE: You can set multiple environment variables if they are separated by commas. For example:

```
-v VAR1=x,VAR2=y
```

The output will be in the working directory where it was submitted.

See also

Run Star-CCM+ in an Azure HPC Cluster (azurehpc project on GitHub)

Azure CycleCloud documentation