Create Hpc Linux Node

10/30/2014

# Introduction

Hpc ComputeNode running on Microsoft® Azure Linux VM is now supported by the latest version of Microsoft® Windows HPC cluster (hereinafter referred to as HPC cluster) - Microsoft HPC Pack 2012 R2 Update 1. By adding Linux ComputeNodes to your Hpc cluster, you can then use compute resources in Azure to run your Linux jobs with Hpc cluster scheduler. A cloud-based HPC cluster provides a flexible, scalable platform for running Big Compute workloads without requiring investment in an on-premises compute cluster infrastructure.

This document demonstrates how to add Linux ComputeNodes to your Hpc cluster deployed in Azure.

In this topic:

* [Prerequisites](#_Prerequisites)
* [Configure HeadNode for Linux node](#_Configure_HeadNode_for)
* [Create Linux ComputeNode Image](#_Create_Linux_ComputeNode)
* [Add Linux nodes to Hpc cluster](#_Add_Linux_nodes)
* [Run a Linux job](#_Run_a_Linux)

# Prerequisites

The following things are required before we start.

**● Microsoft Azure subscription**

You must obtain or have access to a Microsoft Azure subscription.

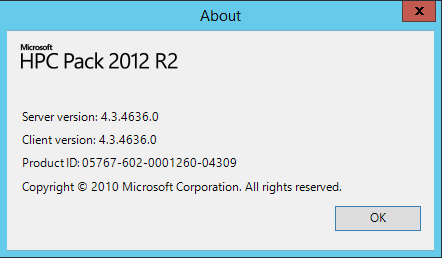
* To create a Microsoft Azure subscription, go to the [Microsoft Azure](http://www.windowsazure.com/) site.
* To access an existing subscription, go to the [Microsoft Azure Management Portal](https://manage.windowsazure.com/).

**● A Hpc cluster deployed in Azure**

You already have your Hpc cluster (Hpc HeadNode) deployed in Azure. And please make sure the version of your Hpc Pack software is 4.3.4636 or higher.

Note:

You can find the version of Hpc Pack in HPC Cluster Manager by selecting Menu Help -> About.



We strongly recommend to use the Out-of-box deployment scripts which released with Microsoft HPC Pack 2012 R2 Update 1 to deploy your Hpc cluster in Azure. You can download the deployment scripts and relative help documents from [Microsoft Connect](http://go.microsoft.com/fwlink/?LinkID=509800).

**● A new cloud service and a storage account**

The cloud service will be used to deploy your new Linux nodes. And it must be empty and in the same Affinity Group/Region as the one of your HeadNode.

Note:

Alternatively you can deploy your Linux nodes in the same cloud service as the one of your HeadNode. But for best practice, it is better to deploy ComputeNodes in separate cloud service.

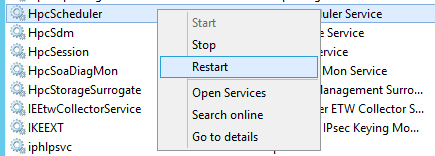
The storage account will be used to store your Linux node image and the VHD files of your new Linux nodes. And it must be in the same Affinity Group/Region as the one of your HeadNode.

**● Source codes and scripts**

You can download all source codes and scripts required in this document from [GitHub](https://github.com/coolmay/whpc-linux-communicator).

# Configure HeadNode for Linux node

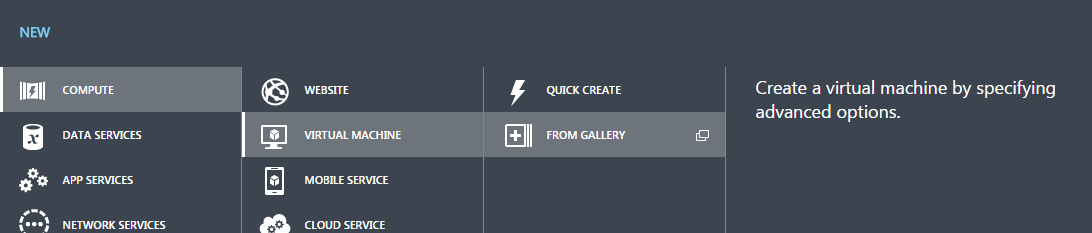
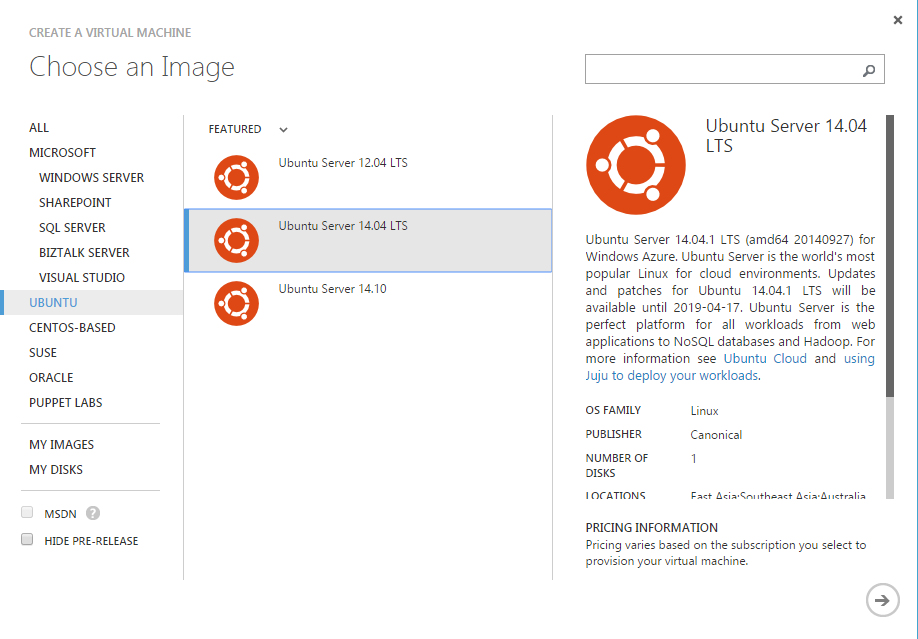
In this section, we will configure the HeadNode to support Linux node.

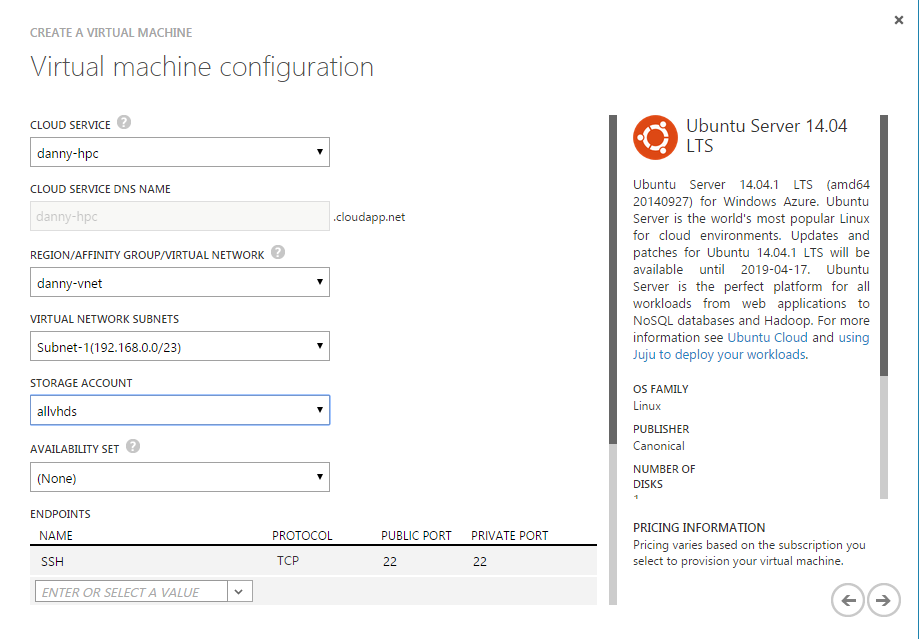
1. Connect to your HeadNode with Remote Desktop Connection.
2. Copy the entire folder **ConfigureLinux** under [Documents](https://github.com/coolmay/whpc-linux-communicator/tree/master/Documents) to **C:\** on HeadNode.
3. Copy the entire folder **UnmanagedResourceCommunicators** under [Documents](https://github.com/coolmay/whpc-linux-communicator/tree/master/Documents) to **%CCP\_HOME%\** on HeadNode.  
   Note1:   
   **%CCP\_HOME%** is the install location of Microsoft Hpc Pack. The default location is **C:\Program Files\Microsoft HPC Pack 2012\**.  
     
   Note2:   
   The dll files under **UnmanagedResourceCommunicators\Linux** are built from Project [*LinuxCommunicator*](https://github.com/coolmay/whpc-linux-communicator/tree/master/LinuxCommunicator).
4. Copy all PowerShell scripts under [Documents\Add-HPCIaasNode-Scripts](https://github.com/coolmay/whpc-linux-communicator/tree/master/Documents/Add-HPCIaasNode-Scripts) to **%CCP\_HOME%\bin\** on HeadNode and overwrite the existing files.
5. Restart the **HpcScheduler** service.  
   

# Create Linux ComputeNode Image

In this section, we will create a Linux VM, configure it as a Linux node and create an image of Linux node from it.

## Step 1: Create a Linux VM

1. Log into the [Microsoft Azure Management Portal](https://manage.windowsazure.com/).
2. Select the **VIRTUAL MACHINES** tab in the left panel and click the **NEW** button on the command bar.
3. In the pop-up menu, select **COMPUTE** -> **VIRTUAL MACHINE** -> **FROM GALLERY**.  
   
4. In the **Choose an Image** page, select **Ubuntu Server 14.04 LTS**, then click the arrow button.  
     
   Note:  
   You can also choose the other type of Linux OS as your wish. But in this document, the commands and scripts demonstrated are based on Ubuntu 14.04 LTS. So if you choose another Linux OS, these commands and scripts may not work correctly. You may have to modify them by yourself.
5. In the **Virtual machine configuration** page, make sure that you have selected
   1. The same virtual network as the one of your HeadNode in the **REGION/AFFINITY GROUP/VIRTUAL NETWORK** field.
   2. The storage account for the linux nodes in the **STORAGE ACCOUNT** field.



1. Complete the wizard to create a new Linux VM.

## Step 2: Configure Linux VM as Linux node

1. Connect to your Linux VM with SSH tool (Putty, WinSCP).
2. Copy all files and folders under [LinuxNodeFiles](https://github.com/coolmay/whpc-linux-communicator/tree/master/Documents/LinuxNodeFiles) to your user’s home directory on Linux VM. In the scenario of this document, it’s **/home/azureuser/**.
   * **nm**: contains the binary and lib files of the LinuxNodeManager program.   
     Note:  
     These files were built with Ubuntu 14.04 LTS. So if you have chosen another type of Linux OS in the previous step, you may have to build your own LinuxNodeManager by yourself. You can find the source codes from [*LinuxNodeMgr*](https://github.com/coolmay/whpc-linux-communicator/tree/master/LinuxNodeMgr). And the LinuxNodeManager program needs cpprest library to run. Please see [*here*](http://casablanca.codeplex.com/wikipage?title=Setup%20and%20Build%20on%20Linux&ANCHOR) for the steps of building cpprest library.
   * **nodemanager**: is the script to run the LinuxNodeManager as a Linux service.
3. Navigate to **nm** directory and modify the **startnm.sh** file with proper values. In the scenario of this document, the username is **azureuser.**  
   Note:  
   You can write your own **startnm.sh** to start the LinuxNodeManager program. But make sure to add path of cpprest library (libcpprest.so) to LD\_LIBRARY\_PATH.

#!/bin/bash

# nm/startnm.sh

# Add path of nm to PATH

PATH=$PATH:/home/azureuser/nm

export PATH

# Add path of cpprest library to LD\_LIBRARY\_PATH

LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/home/azureuser/nm

export LD\_LIBRARY\_PATH

# Set the default work directory for Linux node

cd /home/azureuser/nm

# Run NM and write output to log file

LOG=/home/azureuser/nm/NM\_`date +%y%m%d%H%M%S`.log

/home/azureuser/nm/NM > $LOG

echo "end of startnm" >> $LOG

1. Navigate to up directory and modify the **nodemanage** file to set the path of **startnm.sh** to **EXEC**.

#! /bin/sh

......

NAME=nodemanager

DESC="hpc node manager"

USERGROUP="root:root"

EXEC=/home/azureuser/nm/startnm.sh

#Helper functions

start() {

start-stop-daemon --start --quiet --background --make-pidfile \

--pidfile /var/run/$NAME.pid --chuid $USERGROUP \

--chdir /home/azureuser/nm \

--exec $EXEC

}

......

1. Run the following commands to install necessary packages and make the LinuxNodeManager runnable.

# Update and upgrade

$ sudo apt-get update -qq

$ sudo apt-get upgrade –y

# Install necessary packages for LinuxNodeManager

$ sudo apt-get install g++-4.8 g++ git make libboost1.54-all-dev libssl-dev cmake –y

# Install necessary package for Azure file service

$ sudo apt-get install cifs-utils –y

# Make NM runnable

$ chmod 777 ~/nm/\*

1. Run the following commands to install nodemanager as a Linux service.

# Copy script to init.d

$ cd ~

$ sudo cp nodemanager /etc/init.d

# Change owner and make it excutable

$ cd /etc/init.d

$ sudo chown root:root nodemanager

$ sudo chmod 755 nodemanager

# Install as Linux service

$ sudo update-rc.d nodemanager defaults$ sudo update-rc.d nodemanager enable

1. Reboot your Linux VM to validate whether the LinuxNodeManager starts up with Linux.

# After reboot

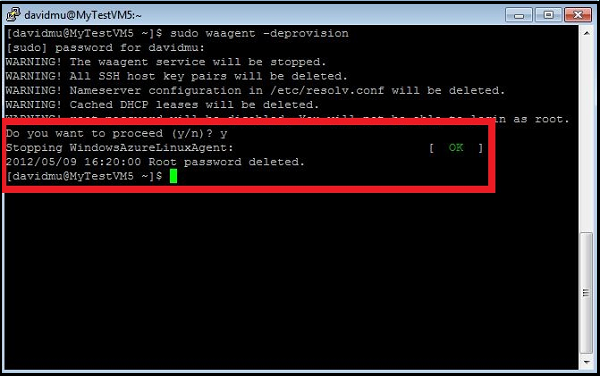
$ ps aux | grep NM

# You should see something like below

root 1292 0.0 0.0 2530012 6596 ? Sl Oct29 0:07 /home/azureuser/nm/**NM**

azureus+ 5765 0.0 0.0 10464 912 pts/0 S+ 09:39 0:00 grep --color=auto **NM**

## Step 3: Create an image of Linux node

1. Run the following command and then enter the password for your account. And type **y** to continue.  
     
   Note:  
   The output from waagent may vary slightly depending on the version of this utility. You may not see the last two lines in Ubuntu 14.04 LTS.  
   

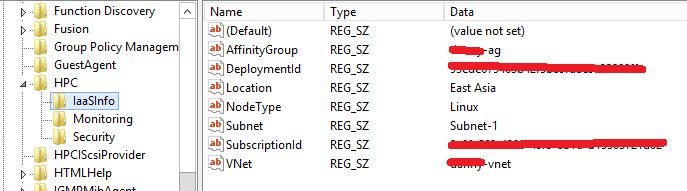
$ sudo waagent -deprovision

1. In the [Management Portal](http://manage.windowsazure.com/), select the virtual machine, and then click **Shut down**.
2. When the virtual machine is stopped, on the command bar, click **Capture** to create an image from this Linux VM.
3. For more information about capturing a Linux image, please see [How to Capture a Linux Virtual Machine to Use as a Template](http://azure.microsoft.com/en-us/documentation/articles/virtual-machines-linux-capture-image/).

# Add Linux nodes to Hpc cluster

In this section, we will run some PowerShell cmdlets on HeadNode to add Linux nodes into our HPC cluster automatically.

1. Connect to your HeadNode with Remote Desktop Connection if you haven’t done so.
2. If your Hpc cluster is not created from the Out-of-box deployment scripts which released with Microsoft HPC Pack 2012 R2 Update 1, you have to add some information of your Azure subscription into your Registry manually.
   1. Open Registry Editor by typing **regedit** in run window.
   2. Go to **HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\HPC** and see whether the Key **IaaSInfo** exists.
   3. If it does not exist, add a new Key and name it **IaaSInfo**. And add the following string value into it.

  
**SubscriptionId**: just your current subscription id.  
**Location**: the location of the datacenter current HPC cluster is.  
**VNet**:  name of the virtual network which is used for current HPC cluster.  
**Subnet**: the subnet name of the using virtual network.  
**DeploymentId**: the deployment id of the HeadNode VM deployment (You can get it from Azure Management portal).  
**AffinityGroup**: the affinity group used for HeadNode, if no affinity group is set, leave this value empty.

1. Open a Windows PowerShell client as administrator.
2. Type the following cmdlet to add 2 Linux nodes into Hpc cluster.  
     
   **ServiceName**: the cloud service we have prepared in [Prerequisites](#_Prerequisites). It could be an empty cloud service or the cloud service of HeadNode.  
   **ImageName**: the name of the image we created in previous steps.  
   **Quantity**: the number of Linux nodes which will be created.  
   **InstanceSize**: the size of the Linux VM.  
   **DomainUserName**: the username to login the Linux nodes.  
   **DomainUserPassword**: the password to login the Linux nodes.  
   **NodeNameSeries**: the name pattern of the Linux nodes.  
   Note:   
   Type **get-help Add-HPCIaaSNode.ps1** to see more details about this cmdlet.

Add-HPCIaaSNode.ps1 -ServiceName "danny-hpc" -ImageName "hpclinuxnode-namd" -Quantity 2 -InstanceSize "Large" -DomainUserName "azureuser" -DomainUserPassword “password" -NodeNameSeries NAMDNODE-%000%

1. Open **HPC Cluster Manager** and navigate to **Node Management**.
2. After a few minutes, in the **Nodes** list, you can see the new Linux nodes have been created.



1. Select the new nodes and make them online, then you can run your Linux jobs with Windows Hpc cluster.

Note:

If you want to stop the nodes, you can run

Stop-HPCIaaSNode.ps1 -Name NAMDNODE-004

Or

Stop-HPCIaaSNode.ps1 -Name NAMDNODE-00\*

This will stop all nodes whose name is in pattern NAMDNODE-00\*.

Then the VM in Azure will be Stopped-Deallocated state, and the health of the nodes in HPC Cluster Manager will be Error.

You can restart them by

Start-HPCIaaSNode.ps1 -Name NAMDNODE-004

And if you want to remove them,

Remove-HPCIaaSNode.ps1 –Name NAMDNODE-004 -DeleteVHD

# Run a Linux job

Now we have already had our Linux nodes ready for work, let us run our first Linux job on them. We can use Microsoft Azure File Service to share data files between HeadNode and Linux nodes. If you want to learn more about Microsoft Azure File Service, please see [Introducing Microsoft Azure File Service](http://blogs.msdn.com/b/windowsazurestorage/archive/2014/05/12/introducing-microsoft-azure-file-service.aspx).

1. Mount Azure File Share as Drive Z:\ on HeadNode.

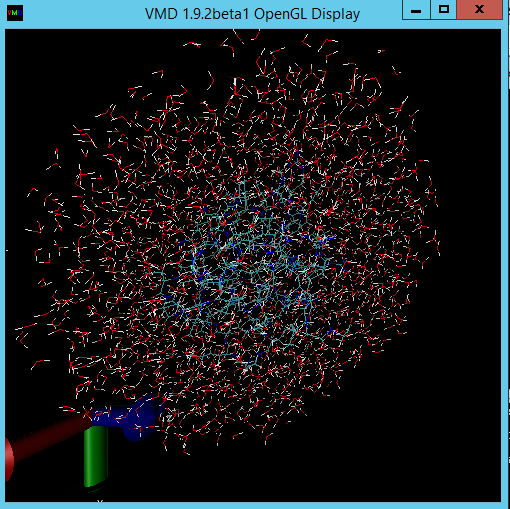
net use z: \\<fileshareaccount>.file.core.windows.net\<filesharename> /u:<fileshareaccount> <fileshareaccountkey>

1. Copy all folders under [NamdSample](https://github.com/coolmay/whpc-linux-communicator/tree/master/Documents/NamdSample) to Z:\.  
   **namd210**: the binaries of Namd (version 2.10 mutilcores).  
   **namdsample**: a set of sample data for running Namd.
2. Open **HPC Cluster Manager** and navigate to **Job Management**.
3. Create a new job and add the following tasks with the command lines.  
   **Node Preparation Task**:  
      
   **Node Release Task**:  
     
   Basic Task:

sudo mkdir -p /share1;sudo mount -t cifs //<fileshareaccount>.file.core.windows.net/<filesharename> /share1 -o vers=2.1,username=<fileshareaccount>,password=<fileshareaccountkey>,dir\_mode=0777,file\_mode=0777

sudo umount /share1;sudo rm -r /share1

/share1/namd210/namd2 +p4 /share1/namdsample/1-2-sphere/ubq\_ws\_eq.conf > /share1/namd.log

1. In the **Resource Selection** tab, select one of the Linux nodes you just created to ensure that this job will run on your Linux node.
2. Click **Submit** to run this job.
3. You can find your job results at **Z:\namd.log** and **Z:\namdsample\1-2-sphere\** after the job finished.
4. Open VMD and view your job results.  
   

Note:

**NAMD** is a parallel molecular dynamics code designed for high-performance simulation of large biomolecular systems. You can find more information about NAMD [*here*](http://www.ks.uiuc.edu/Research/namd/).

**VMD** is a molecular visualization program for displaying, animating, and analyzing large biomolecular systems using 3-D graphics and built-in scripting. You can find more information about VMD [*here*](http://www.ks.uiuc.edu/Research/vmd/).

END OF DOCUMENT