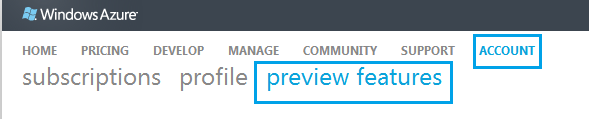
In this tutorial, you will learn to deploy a MongoDB replica set to CentOS VMs on Windows Azure and access it from outside Azure. The following are the tasks to achieve this:

1. Sign up for the feature on the Azure portal
2. Set up affinity group
3. Create and set up the virtual machines
4. Install and run MongoDB
5. Configure the replica set

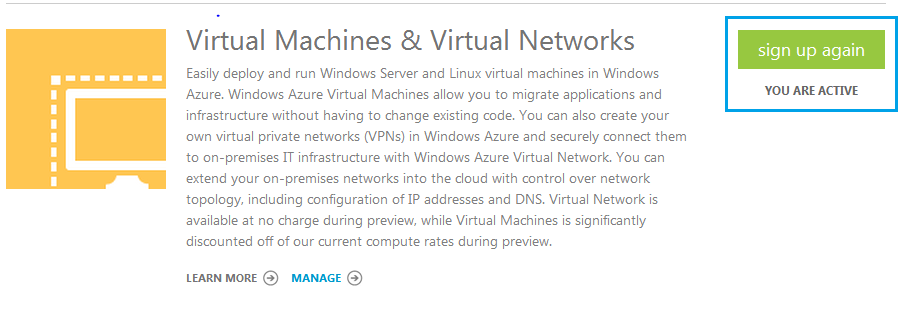
## 1. Signing up

Sign up for ‘Virtual Machines and Virtual Networks’ preview feature from the new Azure portal\*

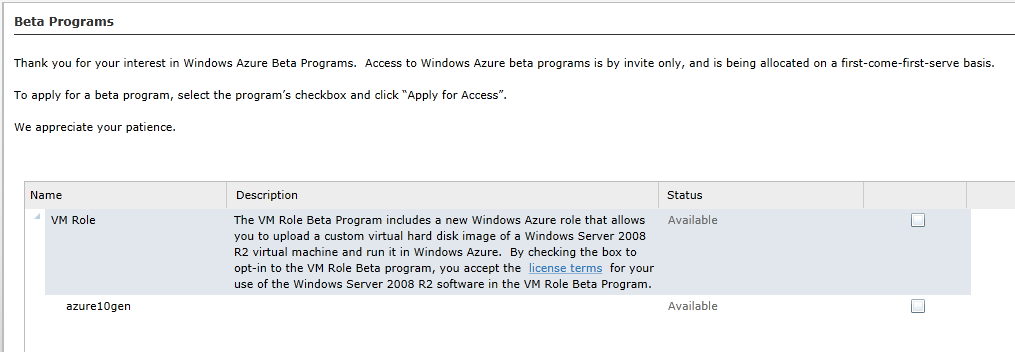
In the ‘Preview’ portal, go to Account->Preview Features.



Sign up for the “Virtual Machines and Virtual Networks”



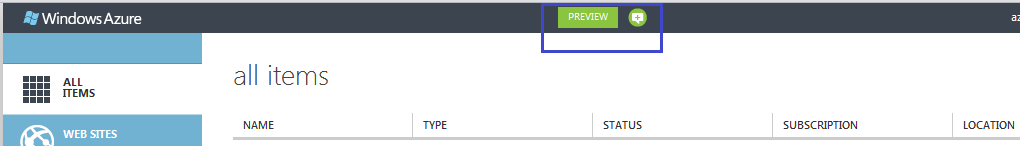
“\*” – The VM Role beta program from the old Azure portal (shown below) is not the same as the preview feature above



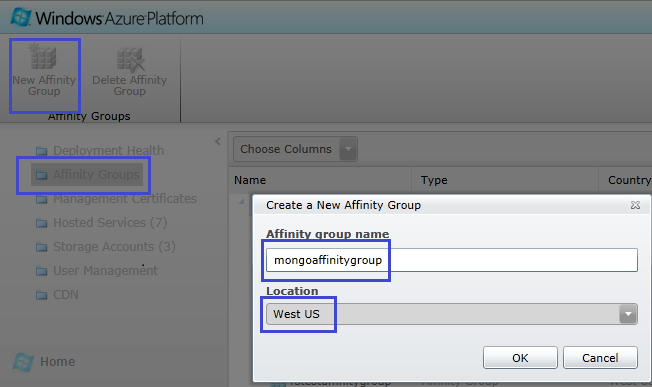
## 2. Set up affinity group

The affinity group needs to be set up from the old portal.

1. If in the new Azure Management preview portal, click ‘Preview’ at the top of the screen and then ‘Take me to the previous portal’ to switch to the old portal



1. Once in the old portal, click ‘Hosted Services, Storage Accounts & CDN’
2. Select ‘Affinity Groups’ and then ‘New Affinity Group’
3. In the “Create a New Affinity Group” dialog, enter an affinity group name such as “mongoaffinitygroup” and choose a location
   * Choose one of the newer Azure data centers such as “West US”, “East US” or “West Europe”



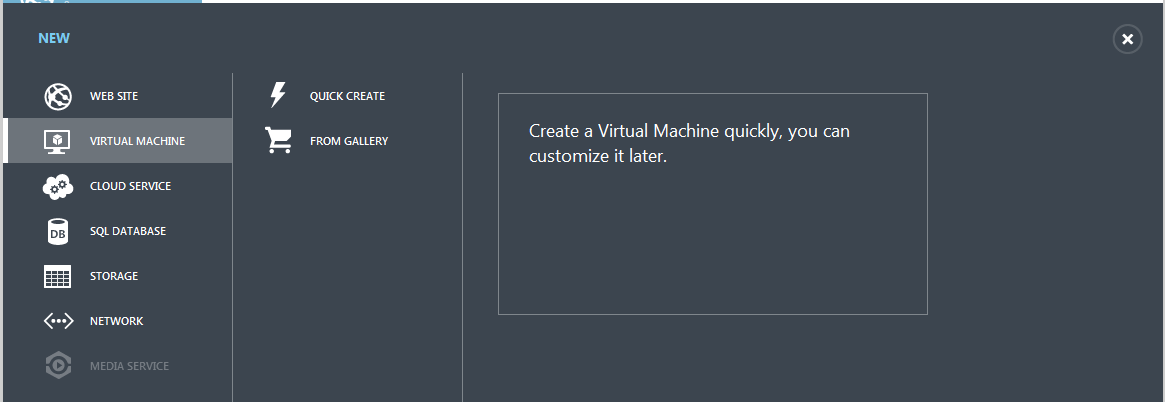
## 3. Create and set up the virtual machines

To create a 3-node MongoDB replica set you will be creating 3 VM instances. For this step, log in to the new preview portal, or from the old portal, click ‘Visit the Previous Portal’ at the bottom of the page

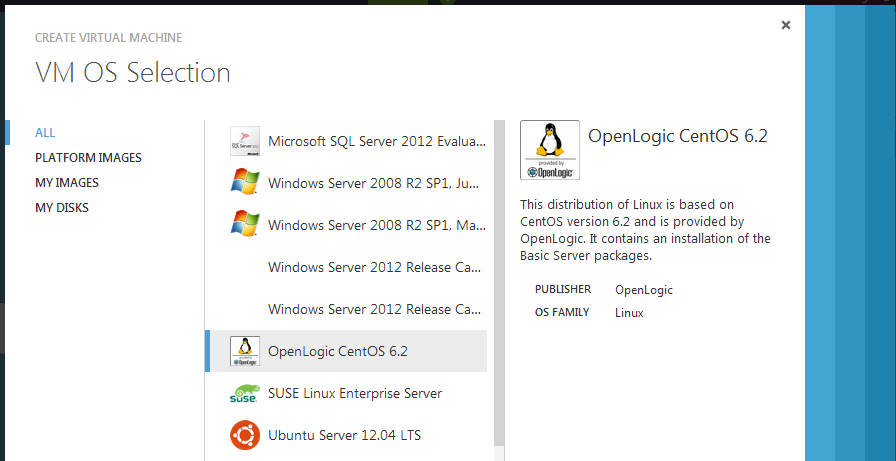
### Create instances

#### Instance 1

1. In the portal, click New->Virtual Machine->From Gallery

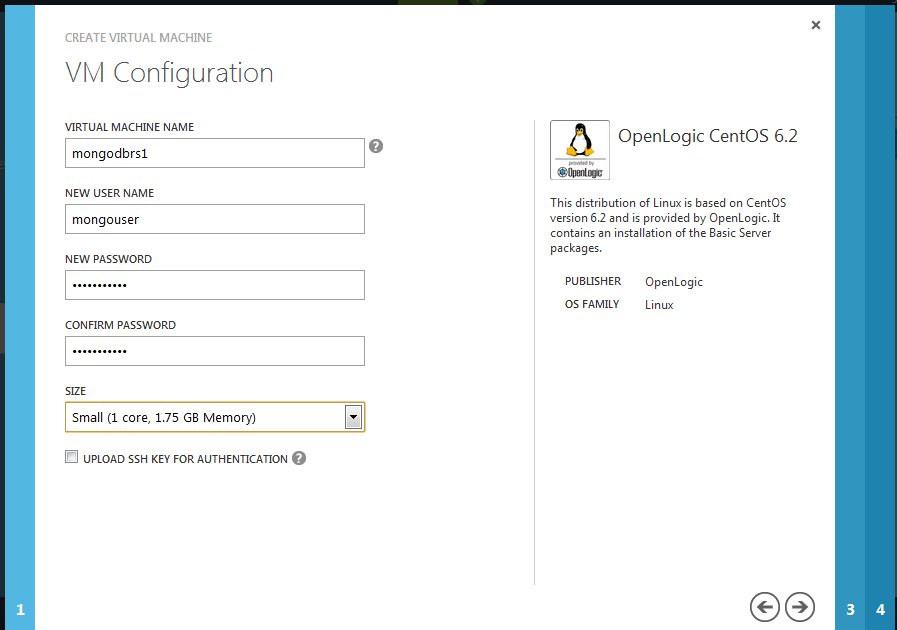


1. On the VM OS Selection page, choose ‘OpenLogic CentOS 6.2’ and click the next arrow to continue



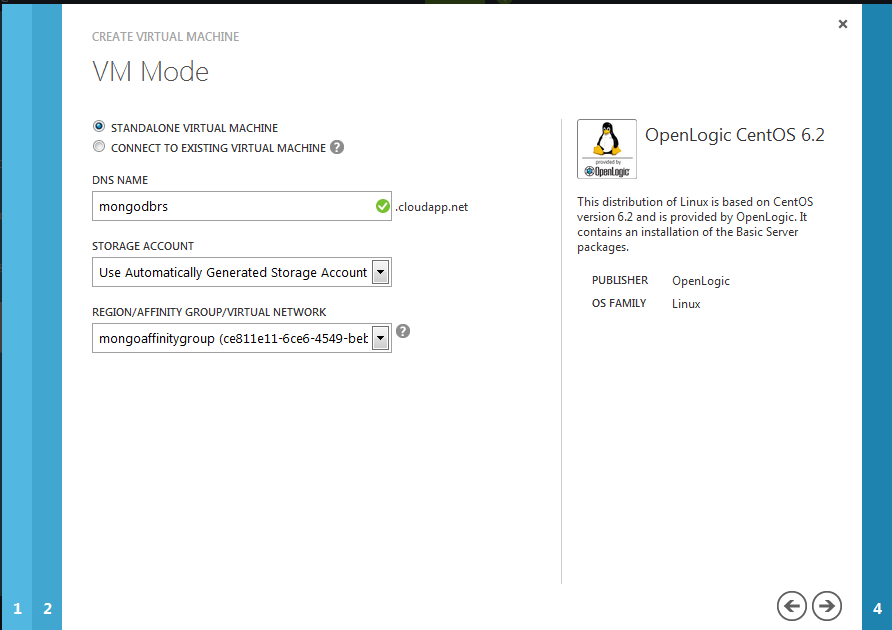
1. On the VM Configuration page, specify values for the parameters
   * Virtual Machine Name - "mongodbrs1"
   * New User Name – "mongouser". This user will be added to the ‘sudoers’ list file
   * New Password box – type a strong password that confirms to the Azure specifications
   * Retype the password to confirm
   * Size – Choose appropriate size from drop down list. For anything but small test instances, choose a size larger than Medium
   * Leave ‘Upload SSH key for Authentication’ unchecked

Click the next arrow to continue

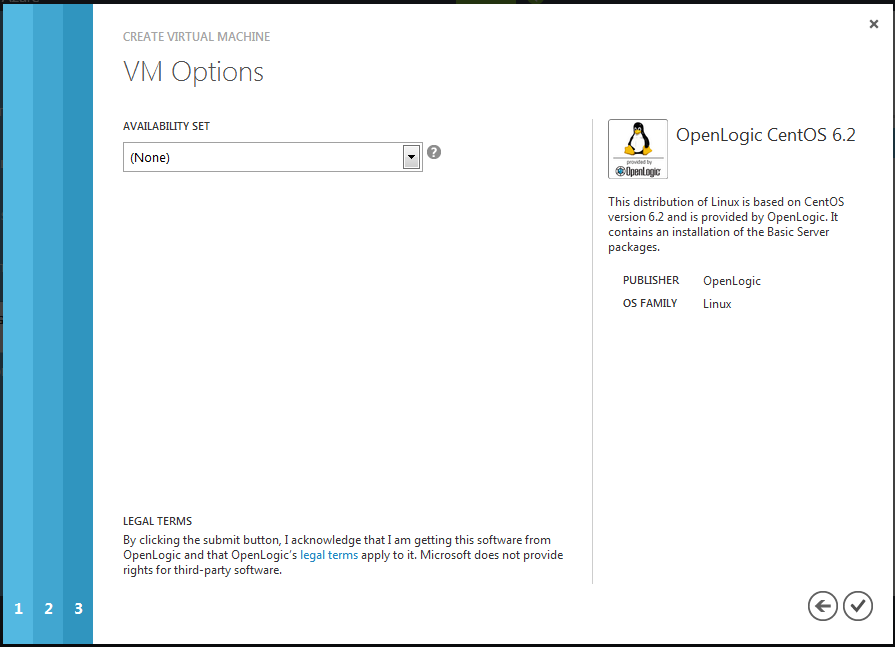


1. On the VM Mode page, specify values for the required parameters
   * Select ‘Standalone Virtual Machine’
   * DNS – a valid DNS prefix e.g. "mongodbrs"
   * Storage Account box – choose ‘Use Automatically Generated Storage Account’
   * In the ‘Region/Affinity Group/Virtual Network’ box, select the affinity group created previously “mongoaffinitygroup”

Click next arrow to continue



1. On the VM Options page select ‘None for Availability Set’



1. Click the check mark to create the VM instance

#### Instance 2

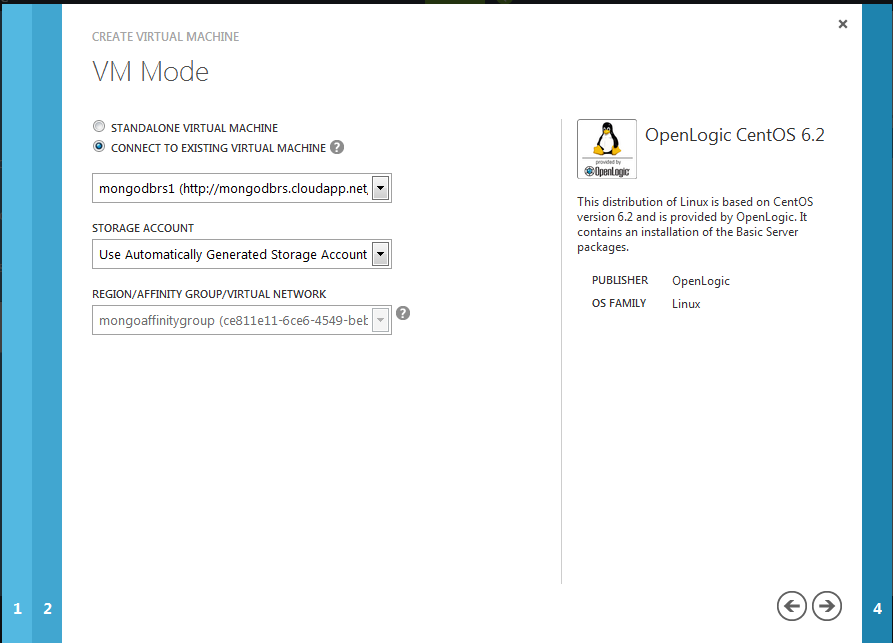
A similar process to creating instance 1.

1. In the portal, click New->Virtual Machine->From Gallery
2. On the VM OS Selection page, choose ‘OpenLogic CentOS 6.2’ and then the next arrow to continue
3. On the VM Configuration page, specify values for the parameters
   * Virtual Machine Name – "mongodbrs2"
   * New User Name – "mongouser". This user will be added to the Sudoers list file. For ease of use specify the same user name as for instance 1
   * New Password box – type a strong password that confirms to the Azure specifications
   * Retype the password to confirm
   * Size – Choose the same instance size as instance 1
   * Leave ‘Upload SSH key for Authentication’ unchecked

Click the next arrow to continue

1. On the VM Mode page, specify values for the required parameters
   * Select ‘Connect to Existing Virtual Machine’
   * In the dropdown choose instance 1 ‘mongodbrs1’
   * Storage Account box – choose ‘Use Automatically Generated Storage Account’
   * The ‘Region/Affinity Group/Virtual Network’ box should be automatically set to “mongoaffinitygroup”

Click the next arrow to continue



1. On the VM Options page select ‘None for Availability Set’
2. Click the check mark to create the VM instance

#### Instance 3

A similar process for creating instance 2. Choose ‘mongodbrs3’ to be the Virtual Machine Name in step 3.

### Configure endpoints

Once the virtual machines are connected you need to configure the endpoints to:

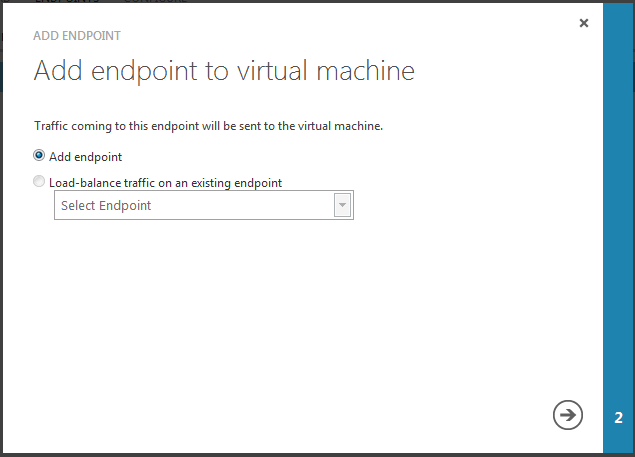
* Allow remote connection
* Allow mongo traffic

#### Instance 1:

1. In the management portal, click virtual machines and click the name of instance 1, ‘mongodbrs1’
2. Now click on endpoints
3. The ssh endpoint should be automatically created. For this endpoint ensure the following are set

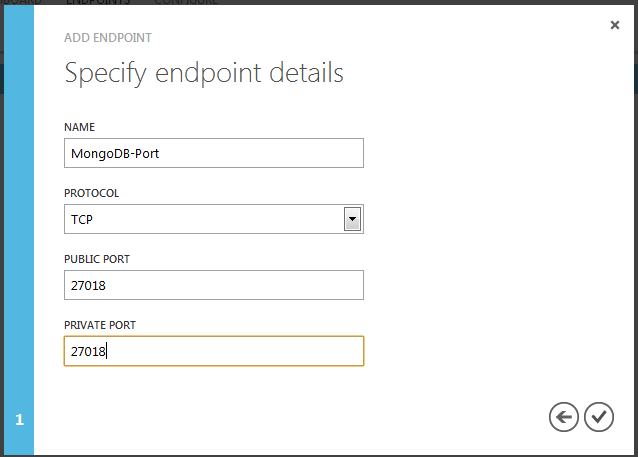
* Protocol – tcp
* Public Port – 22
* Private Port – 22
* Load Balanced – No

1. Create a new endpoint for MongoDB by clicking on ‘Add Endpoint’ at the bottom of the screen
2. Ensure ‘Add endpoint’ is selected and click the right arrow to go to the next screen

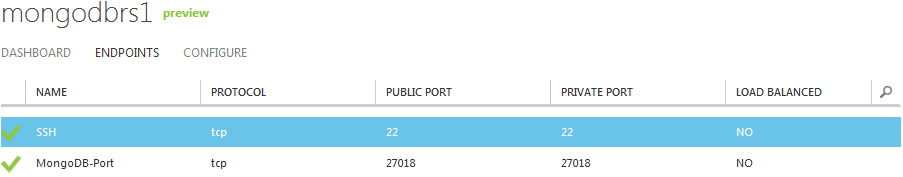


1. Specify the endpoint details as below:

* Name : MongoDB-Port
* Protocol : TCP
* Public Port : 27018
* Private Port : 27018
* Click on check mark to create endpoint



The instance now should have 2 endpoints, 1 for SSH and 1 for MongoDB



#### Instance 2:

We need to configure the endpoints for instance 2 similar to instance 1:

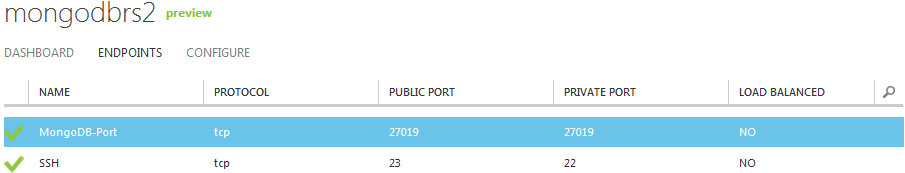
1. In the management portal, click ‘virtual machines’ and click the name of instance 2
2. Now click on ‘endpoints’
3. The ssh endpoint should be automatically created. Ensure that:

* Name – SSH
* Protocol – TCP
* Private Port – 22
* Load Balanced – No

1. Now click on Edit Endpoint at the bottom of the screen and set ‘Public Port’ to 23. Click on the ‘check mark’ to update
2. Create a new endpoint for MongoDB by clicking on ‘Add Endpoint’ at the bottom of the screen
3. Ensure ‘Add Endpoint’ is selected and click the right arrow to go to the next screen
4. Specify the endpoint details as below:

* Name : MongoDB-Port
* Protocol : TCP
* Public Port : 27019
* Private Port : 27019
* Click on check mark to create endpoint

The instance now should have 2 endpoints, 1 for SSH and 1 for MongoDB

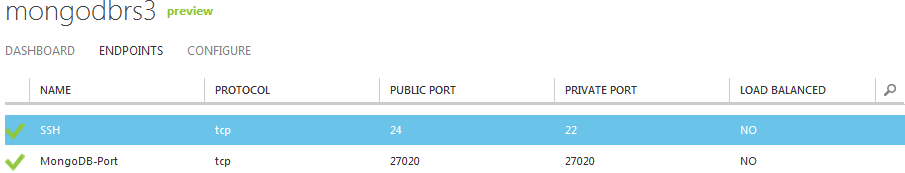


#### Instance 3:

Create endpoints for instance 3 similar to instance 2 with the following changes:

1. In step 4, set public port to 24
2. In step 7, set public and private ports to be 27020

The instance now should have 2 endpoints, 1 for SSH and 1 for MongoDB



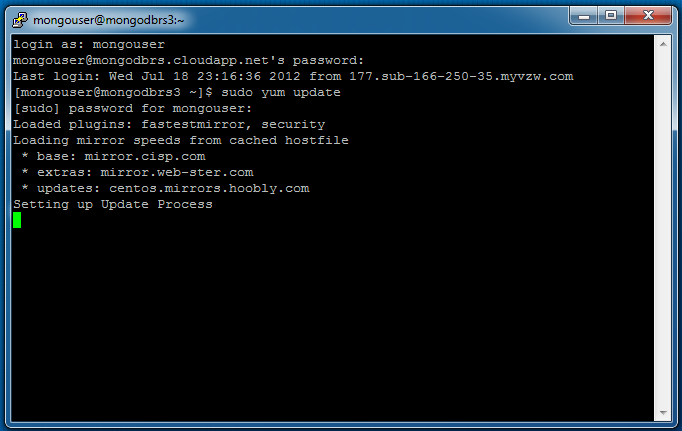
#### Update OS

Use this optional step to update the operating system on each of your VM instances. Once the machine endpoints have been configured above, you need to log into the machines to update them. More information on this can be found at ‘[How to Log on to a Virtual Machine Running Linux](https://www.windowsazure.com/en-us/manage/linux/how-to-guides/log-on-a-linux-vm/)’

Use the username and password you used when creating the virtual machine instances.

Once you are connected to the machine, update the operating system by running: sudo yum update and following the prompts. This could take some time.

**Note**: When connecting to instances 2 and 3, remember to use ports 23 and 24 and not the default ssh port of 22.



### Set up disks

Once the instances are updated, you can then attach a data disk to each of the instances. The data disks will be storing the actual mongodb data as part of –dbpath. More information on Azure data disks can be found at [Data Disk Concepts](http://www.windowsazure.com/en-us/manage/windows/how-to-guides/attach-a-disk/%23concepts).

To set up the data disk follow the steps outlined below for each of the instances you created:

1. Attach an empty disk to the instance as described in [How to: Attach an empty disk](http://www.windowsazure.com/en-us/manage/windows/how-to-guides/attach-a-disk/#attachempty)
   1. Create a data disk of at least 10 GB
2. Now initialize the data disk by following the steps described at [How to: Initialize a new data disk in Linux](http://www.windowsazure.com/en-us/manage/windows/how-to-guides/attach-a-disk/%23initializeinlinux)
   1. Ensure you use ext4 for the file system and not ext3
3. Also once mounted, create a mongodb data directory by:
   1. Log on onto the instance
   2. Then run *sudo chown `id -u` /mnt/datadrive/* to make the mongouser the owner of the data directory
   3. Run the following command: *mkdir -p /mnt/datadrive/data*

## 4. Install and run MongoDB

As part of this step, you will be using the official 10gen supplied packages to install, configure and run MongoDB as a service using YUM. You want to install as a service since this would ensure that mongod is started on machine restart also. More information can be found at [Install MongoDB on RedHat Enterprise, CentOS, or Fedora Linux](http://docs.mongodb.org/manual/tutorial/install-mongodb-on-redhat-centos-or-fedora-linux/) doc page.

### Install MongoDB

Repeat the following steps on each instance.

1. Log onto the instance
2. Create a */etc/yum.repos.d/10gen.repo* file to hold information about your repository using your favorite editor as sudo. Place the following configuration in */etc/yum.repos.d/10gen.repo* file:

*[10gen]*

*name=10gen Repository*

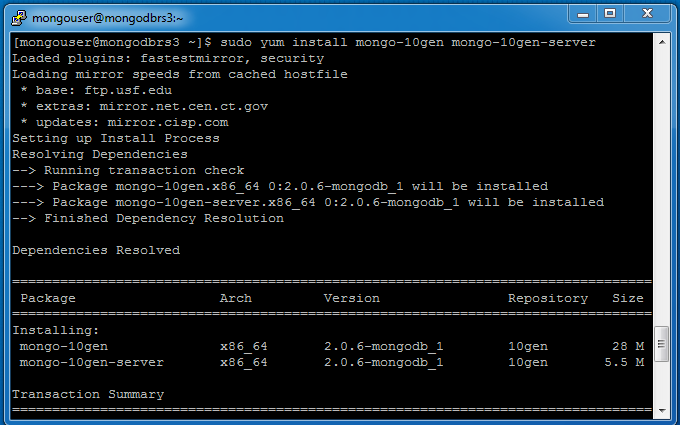
*baseurl=http://downloads-distro.mongodb.org/repo/redhat/os/x86\_64*

*gpgcheck=0*

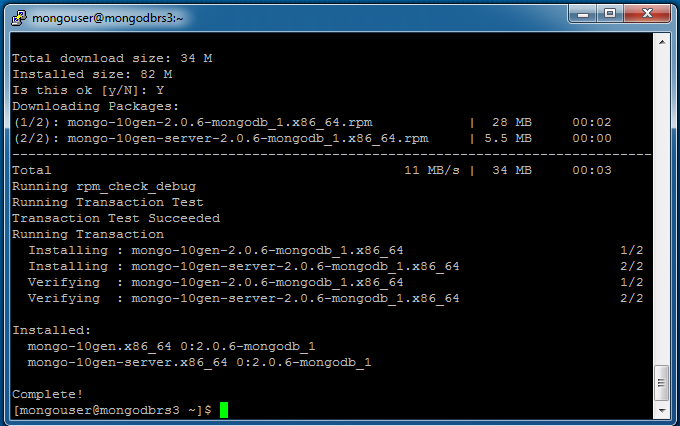
*enabled=1*

1. After saving the new .repo file, issue the following command to update the local package database: *sudo yum update*
2. Issue the following command (as root or with sudo) to install the latest stable version of MongoDB and the associated tools:

*sudo yum install mongo-10gen mongo-10gen-server*



1. When this command completes, you have successfully installed MongoDB.



### Configure MongoDB

The packages installed in the previous step configure MongoDB using the /etc/mongod.conf file in conjunction with the control script. You can find the init script at /etc/rc.d/init.d/mongod. As part of this step you will edit the mongod.conf file to set the appropriate parameters. If the parameters are commented, make sure to uncomment them.

#### Instance 1

1. Connect to the instance using ssh or PuTTY
2. As sudo, edit */etc/mongod.conf* to set the following parameters:

*port = 27018*

*dbpath = /mnt/datadrive/data*

*logpath = /mnt/datadrive/mongod.log*

*replSet = mongors*

1. Save the config file

#### Instance 2

1. Connect to the instance using ssh or PuTTY
2. As sudo, edit */etc/mongod.conf* to set the following parameter:

*port = 27019*

*dbpath = /mnt/datadrive/data*

*logpath = /mnt/datadrive/mongod.log*

*replSet = mongors*

1. Save the config file

#### Instance 3

1. Connect to the instance using ssh or PuTTY
2. As sudo, edit */etc/mongod.conf* to set the following parameter:

*port = 27020*

*dbpath = /mnt/datadrive/data*

*logpath = /mnt/datadrive/mongod.log*

*replSet = mongors*

1. Save the config file

### Start MongoDB

Once the configuration files have been edited, start the database process mongod on each instance by:

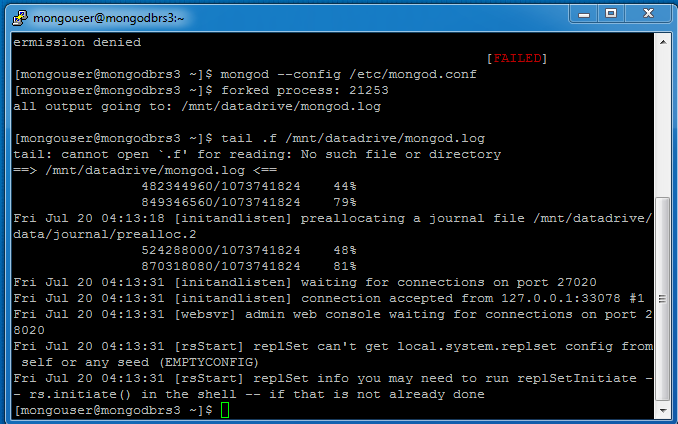
1. Log on onto the instance
2. Run the following command to start the process:

*mongod –config /etc/mongod.conf*

1. This should start the mongod process
2. Verify that mongod start by tailing the log file using the command

*tail –f /mnt/datadrive/mongod.log*

1. The *‘waiting for connections’* message in the log file indicates mongod is up and running and waiting for client connections. This may take a while as mongod preallocates its journal files



## 5. Configure the replica set

At this point in time, you should have mongod running on all 3 of your instances. You can now configure the 3 instances as a replica set by connecting to 1 of the 3 instances from within Azure or from outside.

Connect to the running mongod process using the mongo shell:

1. If connected to the VM instance type the following command where port is 27018 for instance 1, 27019 for instance 2 and 27020 for instance 3: *mongo –port <port number>*
2. If connecting to the mongod process in Azure from your local machine use the following command: *mongo –host* mongodbrs*.cloudapp.net –port <port number>*
3. In the mongo shell type the following:

*> conf = {*

*\_id = “mongors”,*

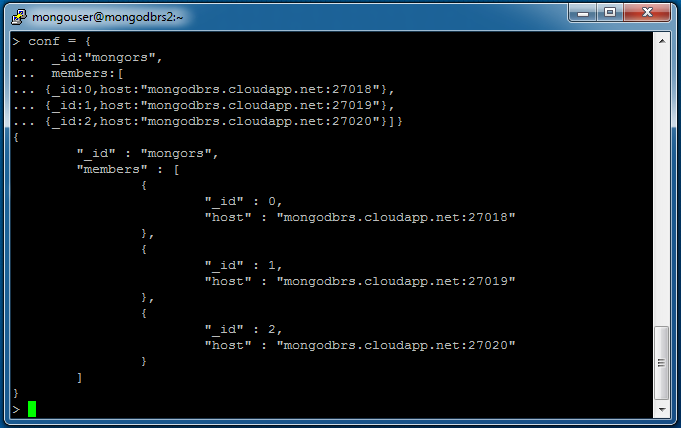
*members : [*

*{id:0, host:”mongodbrs.cloudapp.net:27018},*

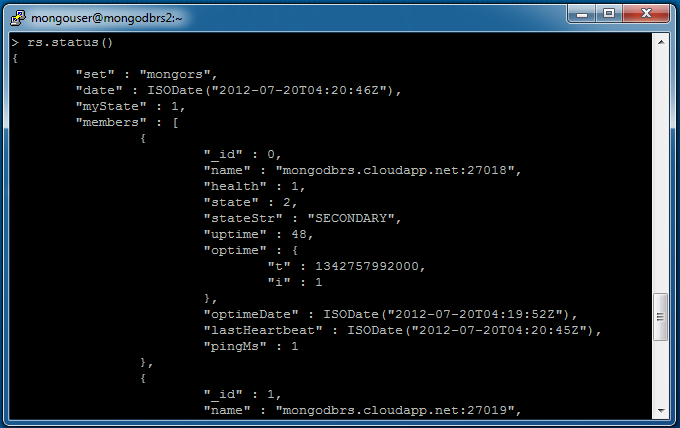
*{id:0, host:”mongodbrs.cloudapp.net:27019},*

*{id:0, host:”mongodbrs.cloudapp.net:27020}]}*

*>rs.initiate(conf)*



1. This will start the initialization of the mongodb replica set
2. Type the command ‘*rs.status()*’ to check the status of the replica set. Upon successful initialization, you should see 1 of the 3 instances being the ‘Primary’ of the set and the other 2 being the ‘Secondaries’



1. You have now successfully initiated the replica set

## Summary

In this tutorial you have learned how to create a set of CentOS virtual machines on Windows Azure, deploy MongoDB to them and create a replica set out of them. You can access this set from anywhere using the connection string *‘mongodb://mongodbrs.cloudapp.net:27018, mongodbrs.cloudapp.net:27019, mongodbrs.cloudapp.net:27020/?replicaSet=mongors’.*

More information on MongoDB can be found at <http://docs.mongodb.org/manual/>.

To create and deploy a replica set to Windows Virtual Machines on Azure you can use the [MongoDB Installer for Windows Azure](http://www.mongodb.org/display/DOCS/MongoDB+Installer+for+Windows+Azure).