

MySQL Clustering in CentOS 7

Content:

- Prerequisites
- Installation of CentOS(64x) in VMWare
- Step 1 - Setup Management Node
 - A. Download the MySQL Cluster software
 - B. Install and Remove Packages
 - C. Install MySQL Cluster
 - D. Configure MySQL Cluster
 - E. Start the Management Node
- Step 2 - Setup the MySQL Cluster Data Nodes
 - A. Login as root user and download the MySQL Cluster software
 - B. Install and Remove Packages
 - C. Install MySQL Cluster
 - D. Configure Data Node
 - E. Redo step 2.A - 2.D on db3 server
- Step 3 - Setup SQL Node
 - A. Log in and Download MySQL Cluster
 - B. Install and Remove Packages
 - C. Install MySQL Cluster
 - D. Configure the SQL Node
 - E. Redo step 3.A - 3.D on db5 server
- Step 4 - Monitor the Cluster
- Step 5 - Testing the Cluster
- Conclusion

How to Install and Configure MySQL Cluster on CentOS 7

MySQL Cluster is designed to provide a MySQL compatible database with high availability and low latency. The MySQL Cluster technology is implemented through the NDB (Network Database) and NDBCLUSTER storage engines and provides shared-nothing clustering and auto-sharing for MySQL database systems. In the shared-nothing architecture, each of nodes has its own memory and disk, the use of shared storage such as NFS, SANs are not recommended and supported.

To implement a MySQL Cluster, we have to install three types of nodes. Each node type will be installed on its own server. The components are:

1. **Management Node - NDB_MGMD/MGM**

The Cluster management server is used to manage the other node of the cluster. We can create and configure new nodes, restart, delete, or backup nodes on the cluster from the management node.

2. **Data Nodes - NDBD/NDB**

This is the layer where the process of synchronizing and data replication between nodes happens.

3. **SQL Nodes - MySQLD/API**

The interface servers that are used by the applications to connect to the database cluster. In this tutorial, I will guide you through the installation and configuration of a MySQL Cluster with CentOS 7. We will configure the management node, two data nodes, and two SQL nodes.

Prerequisites

5 CentOS servers or virtual machines. I will use the hostnames and IP addresses as shown below:

- **Management Node**
`centosmn` = 192.168.19.130
- **Data Nodes**
`datanode1` = 192.168.19.129
`datanode2` = 192.168.19.131
- **SQL Nodes**
`sqlnode1` = 192.168.19.132
`sqlnode2` = 192.168.19.133
- The OS is CentOS 7 - 64bit.

Installation of CentOS 7 (64X) in VMWare

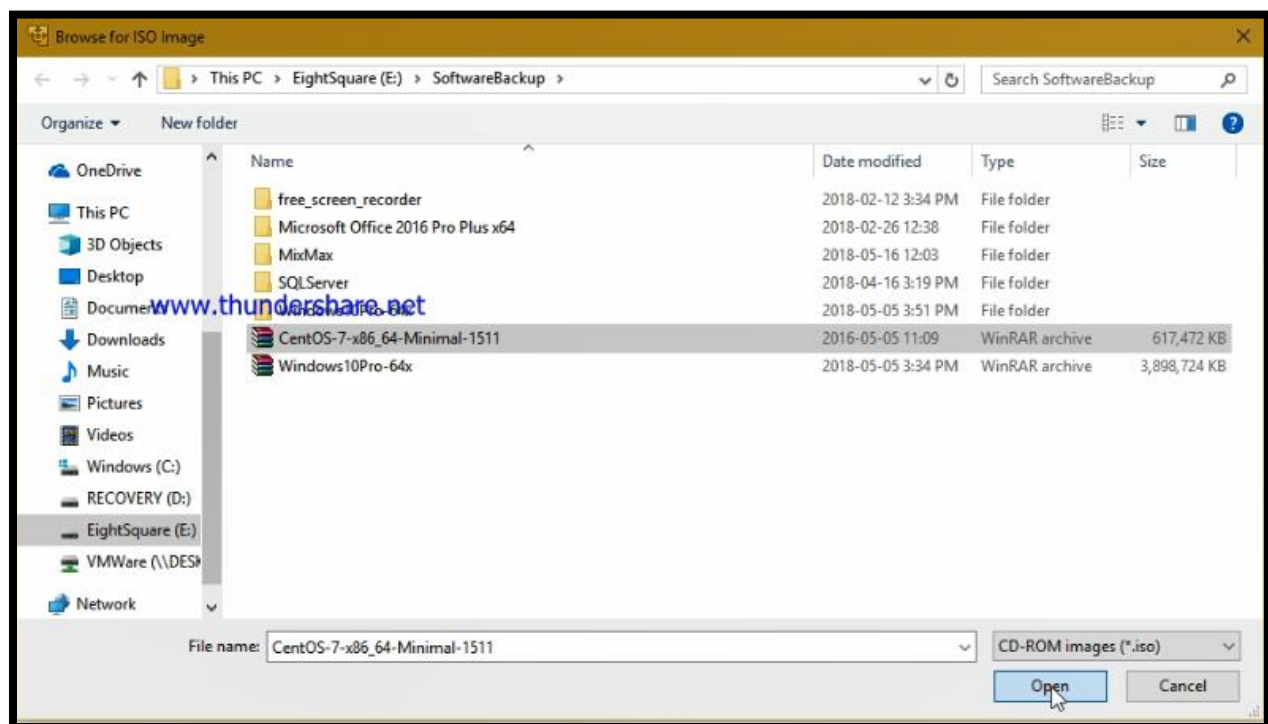
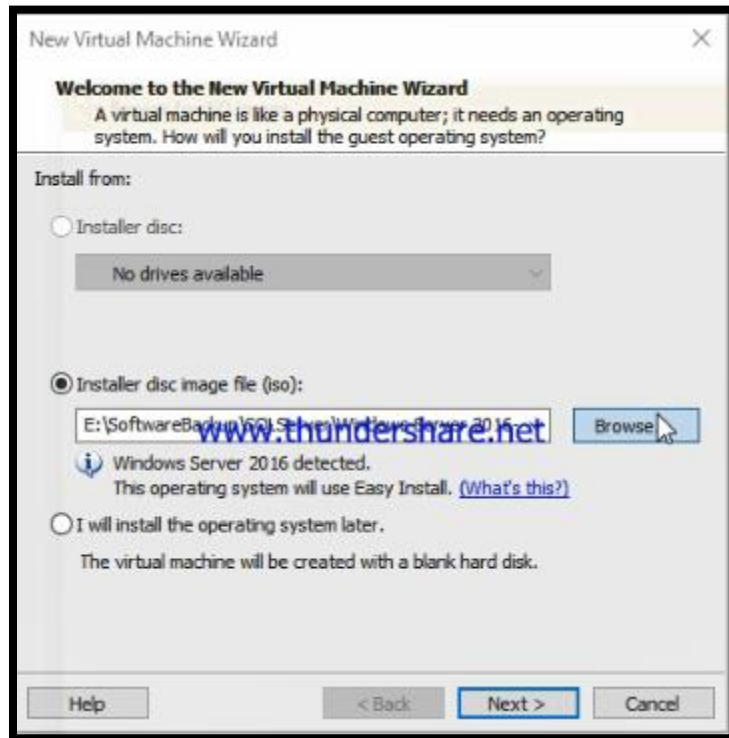
As we need 5 Virtual Machine to implement the MySQL Clustering so we need to first install Operation System (CentOS 7). The steps to install CentOS in VMWare is represented by the screenshot below. We need to install CentOS in five different VM. So, repeat the steps to install the OS on five different VM.

For the very first time we need to create the Virtual Machine for the OS.



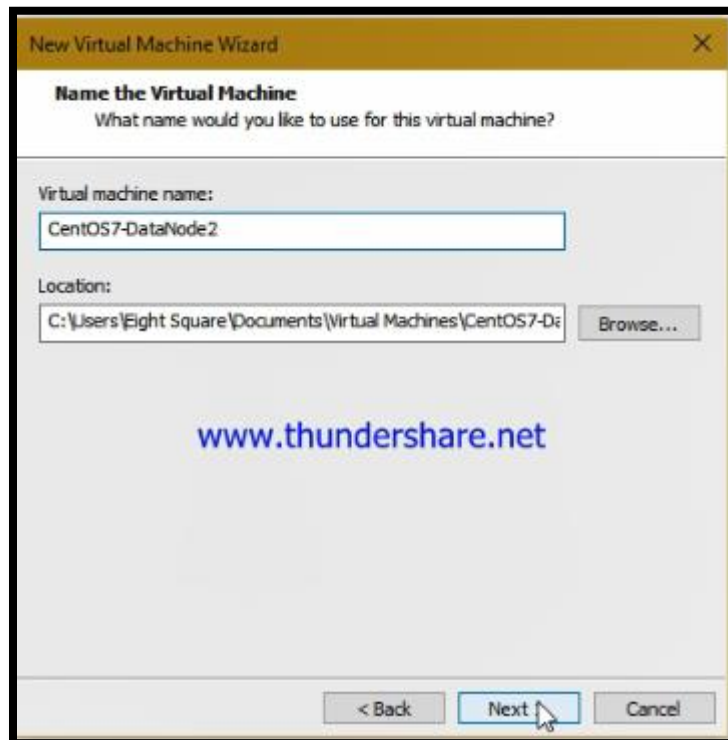
4 | MySQL Clustering

Browse the OS file and click Next to continue the installation.

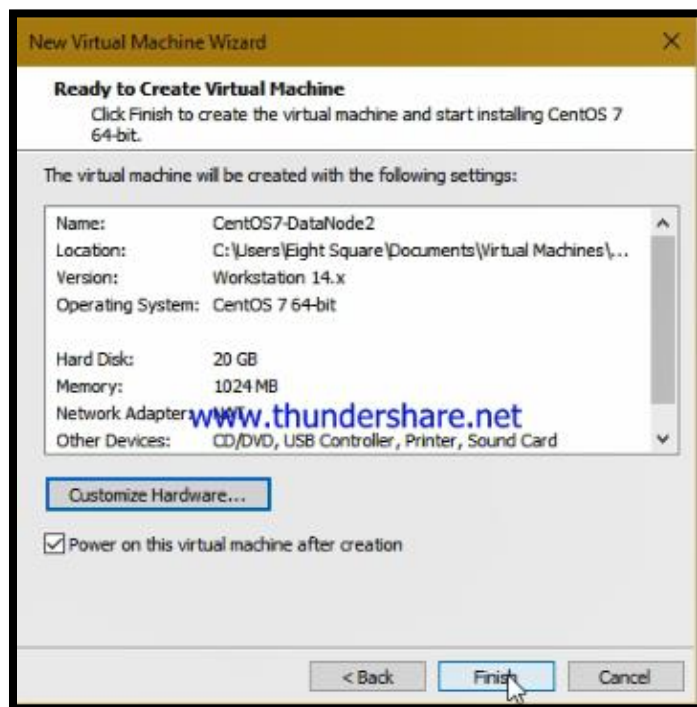




Give the Virtual Machine a Distinct name and click Next



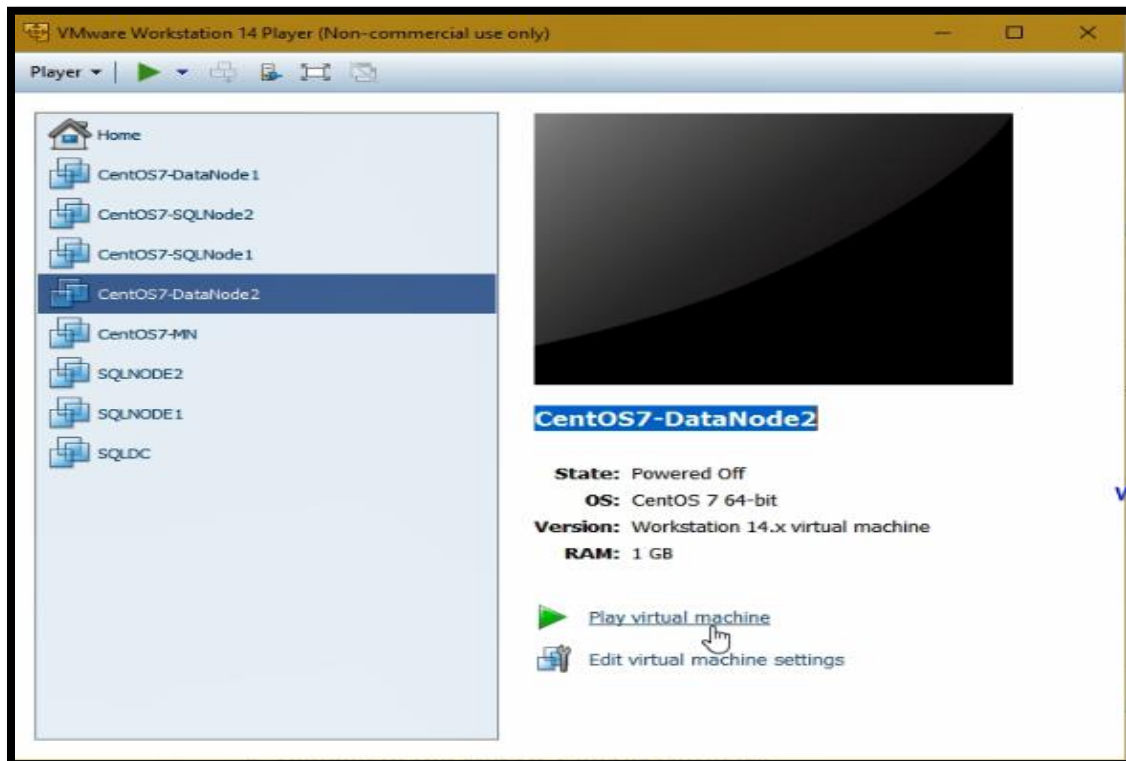
Instead of splitting the virtual disk files select to store in a single file and click Next to continue.

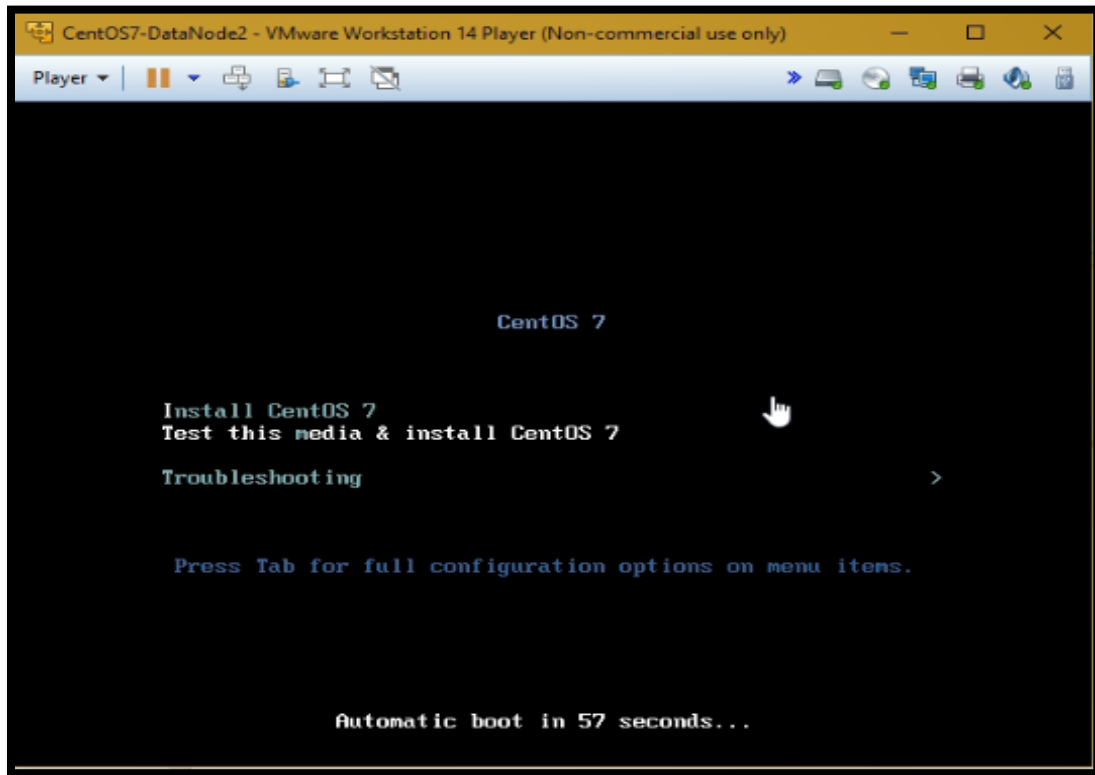


Click Finish to complete the creation of the Virtual Machine.

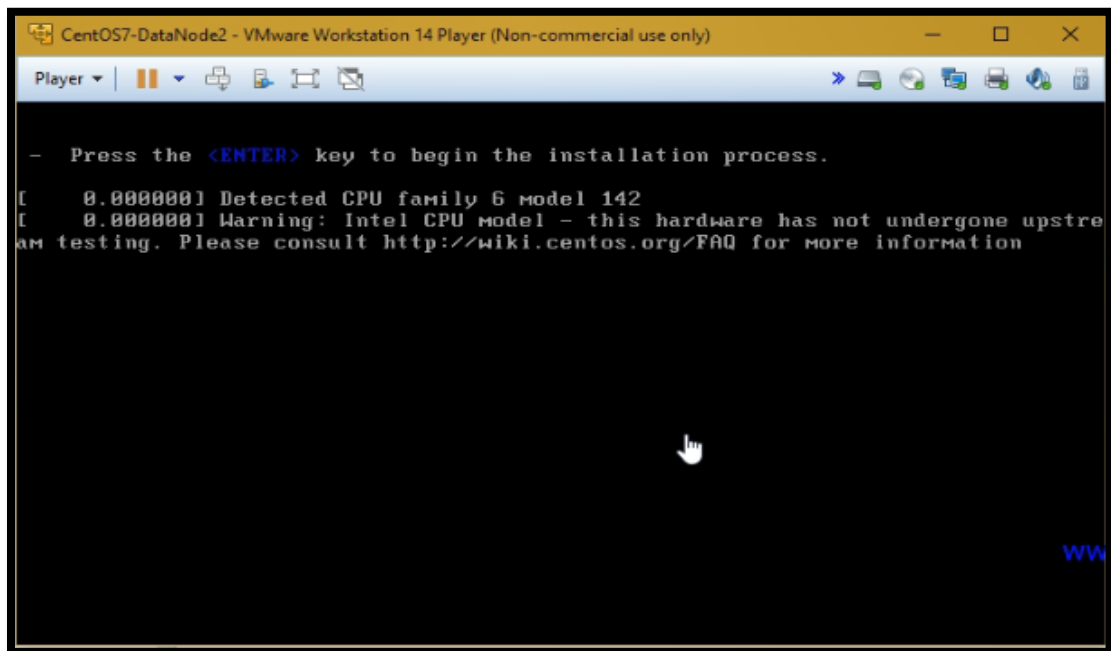


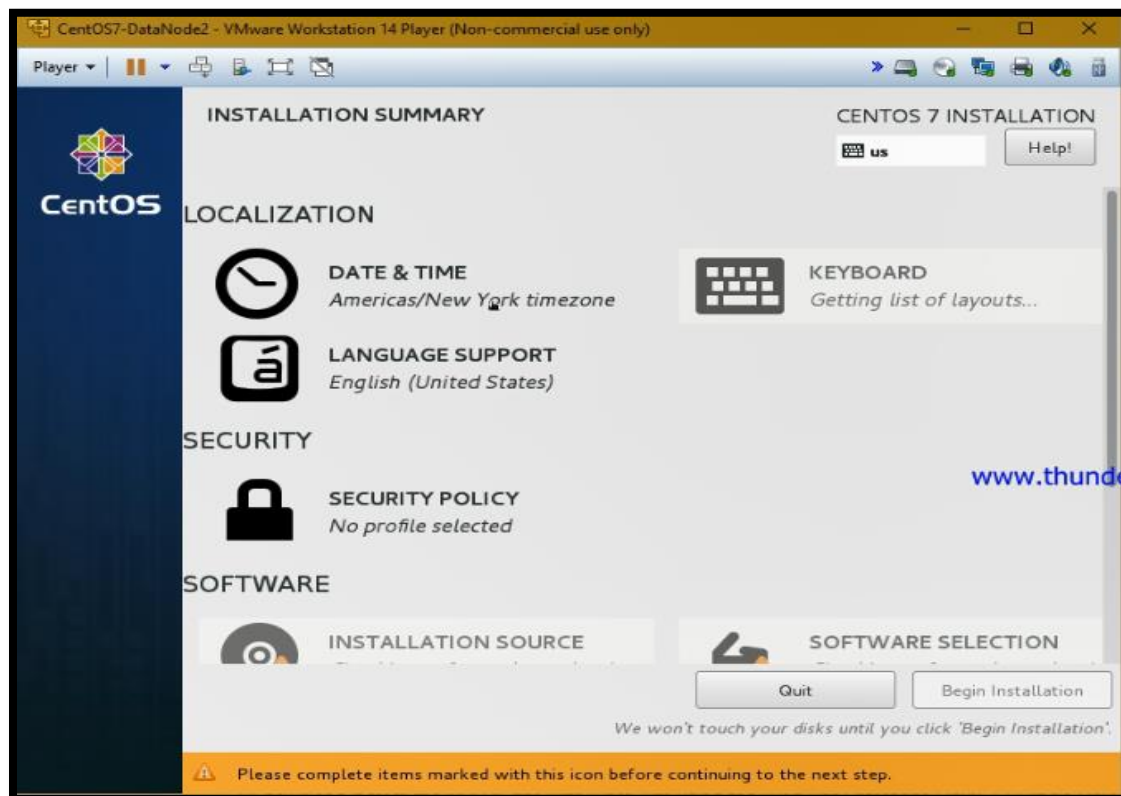
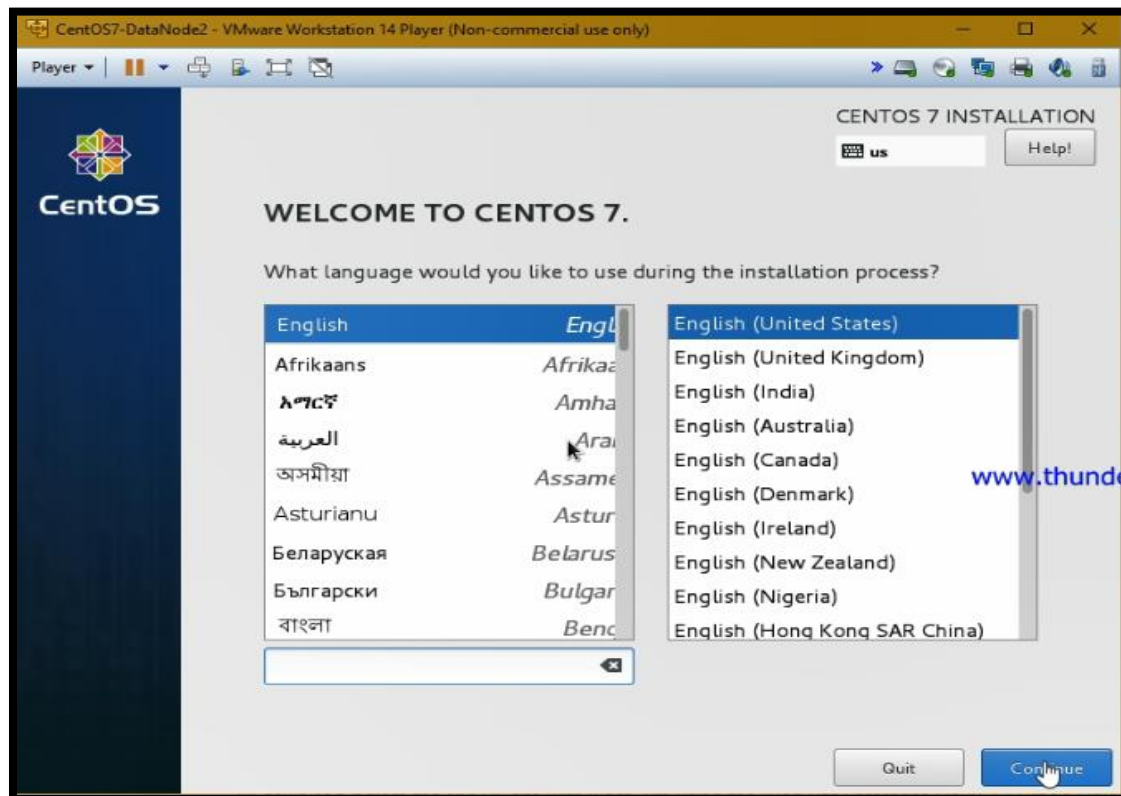
Select the Virtual Machine and Click on **Play virtual machine**

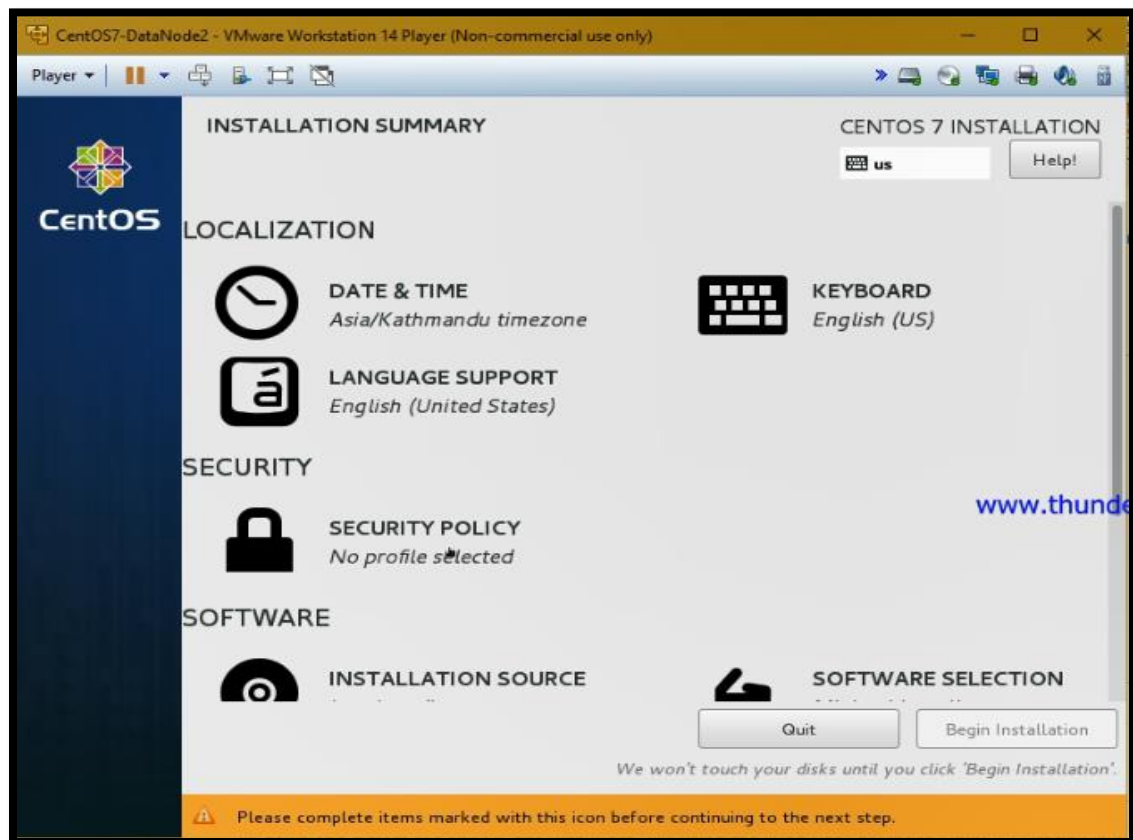
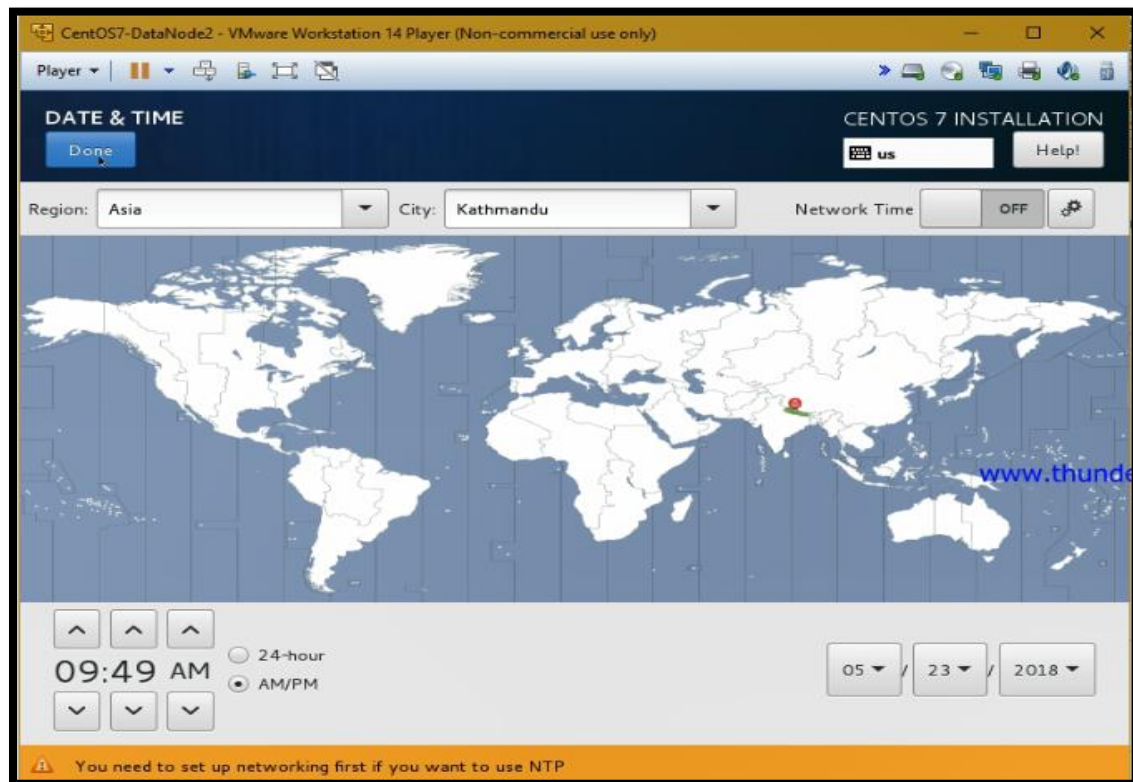


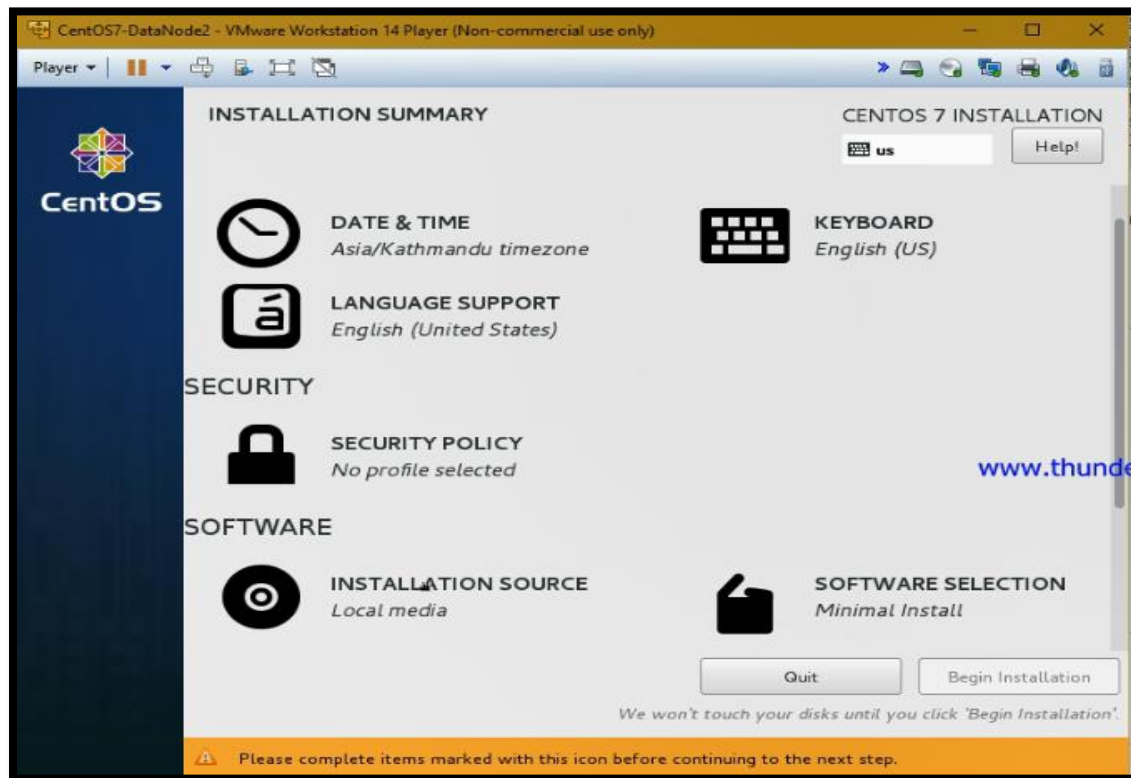
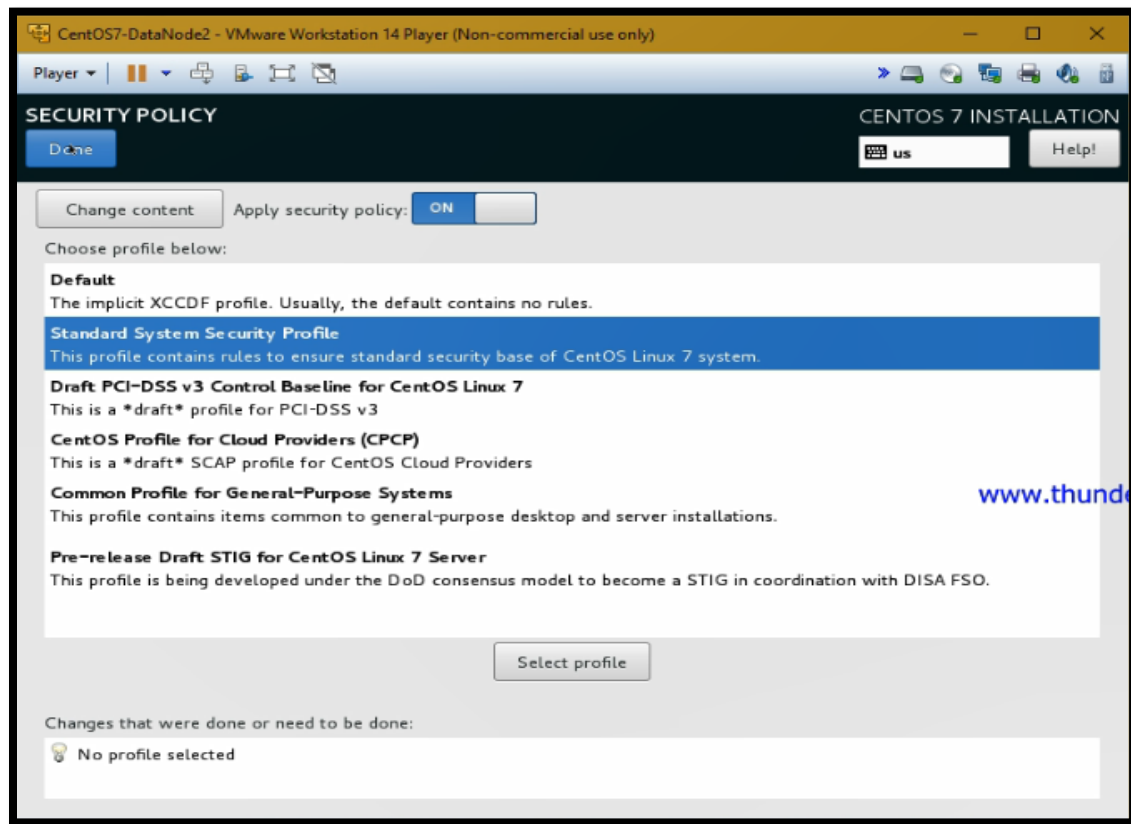


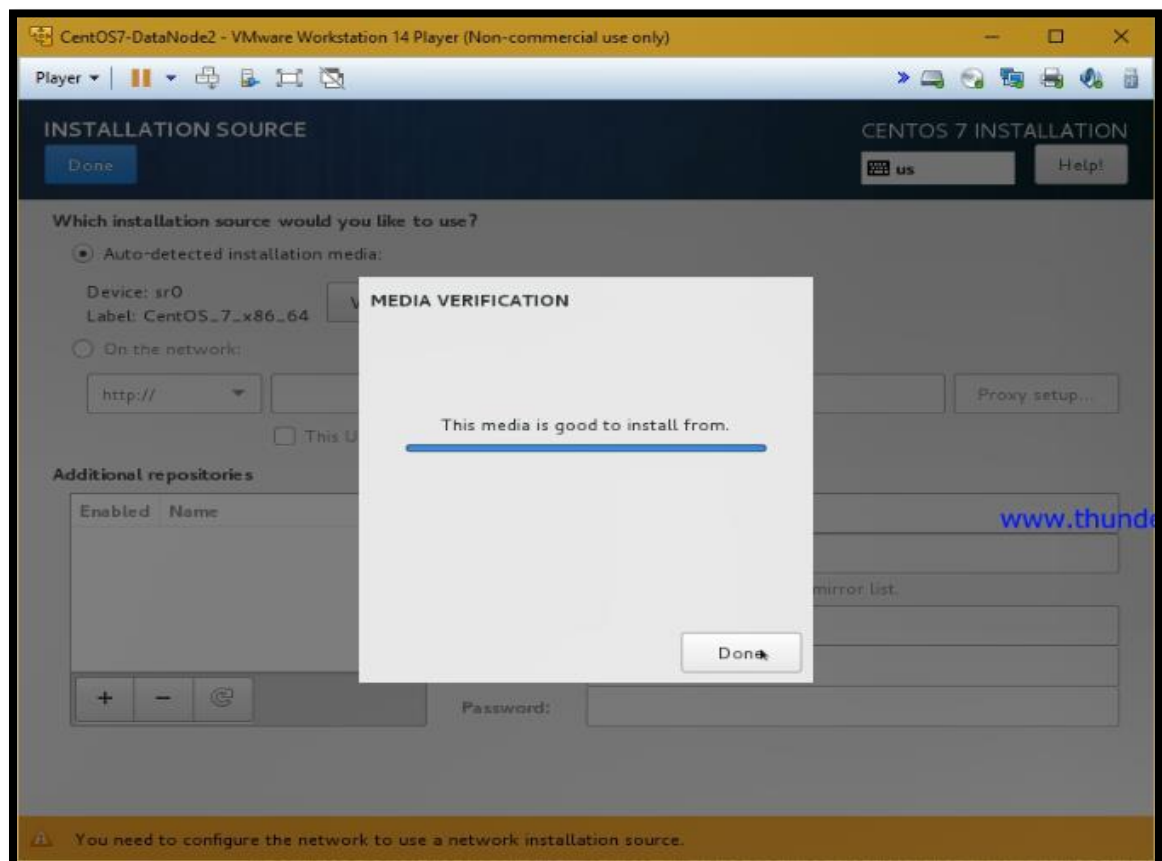
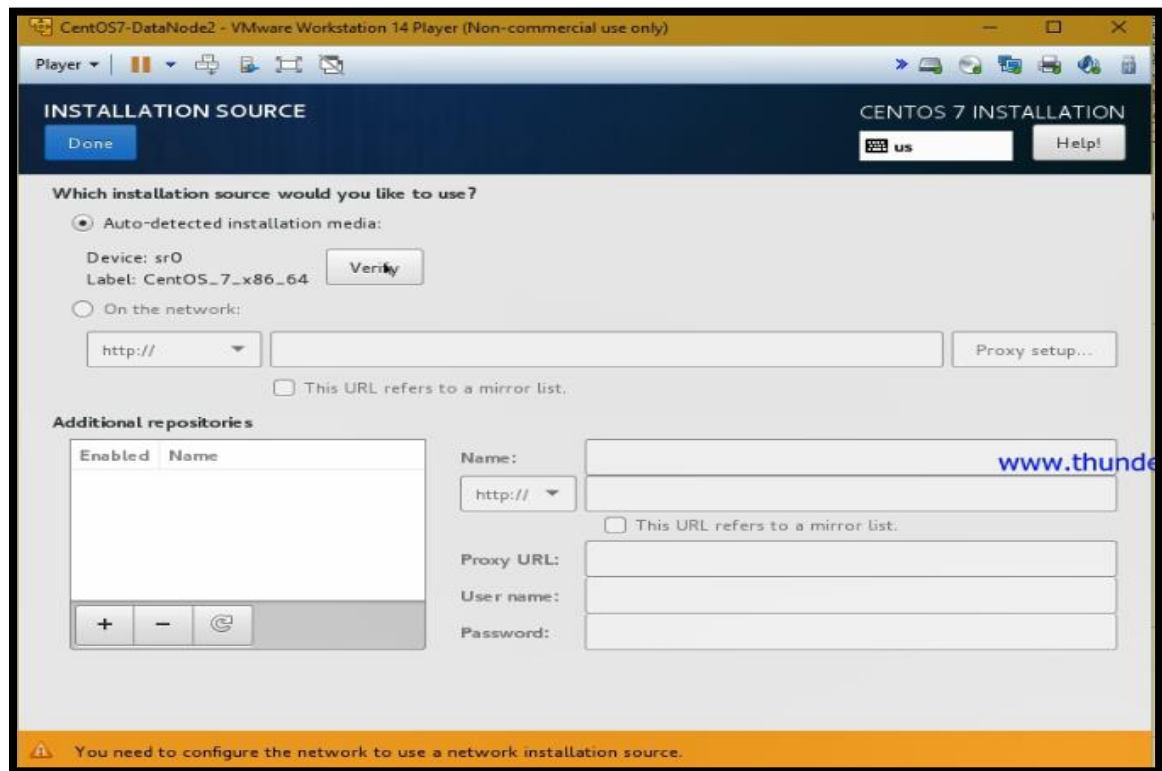
Press **Enter** key to continue.

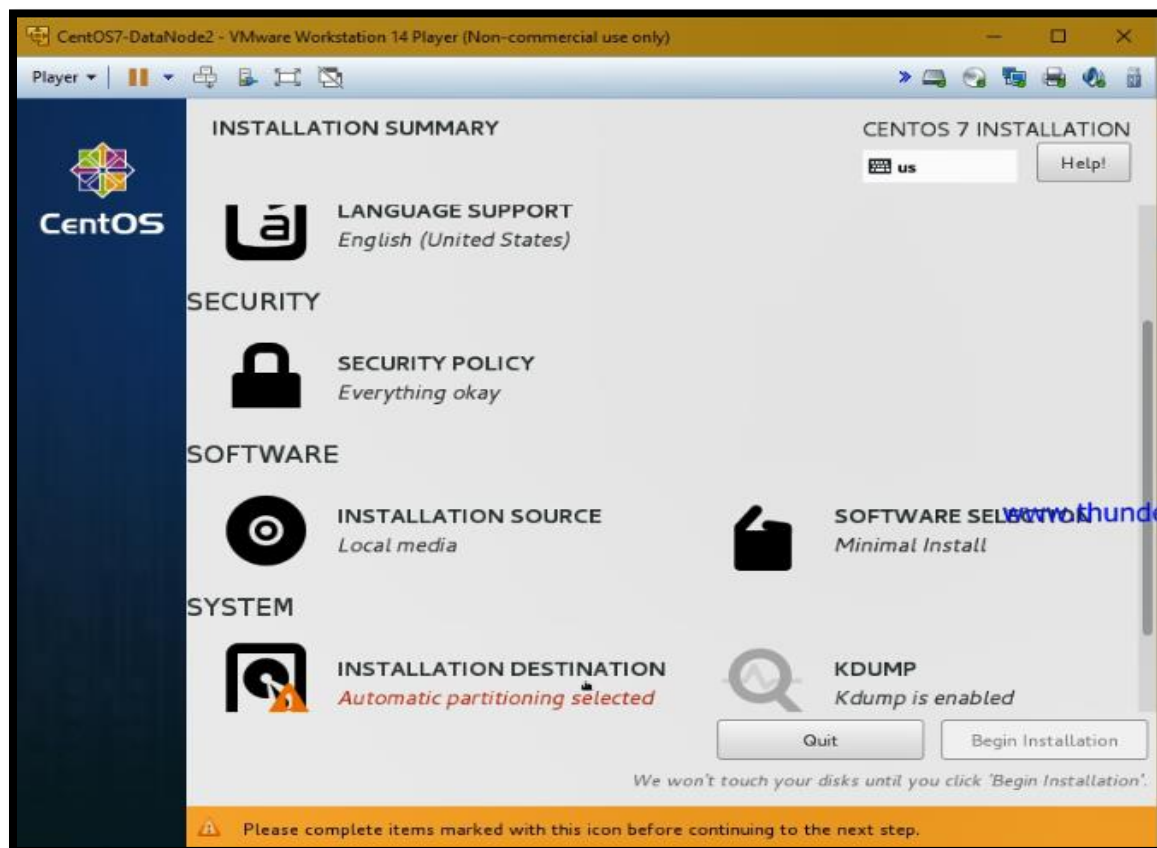
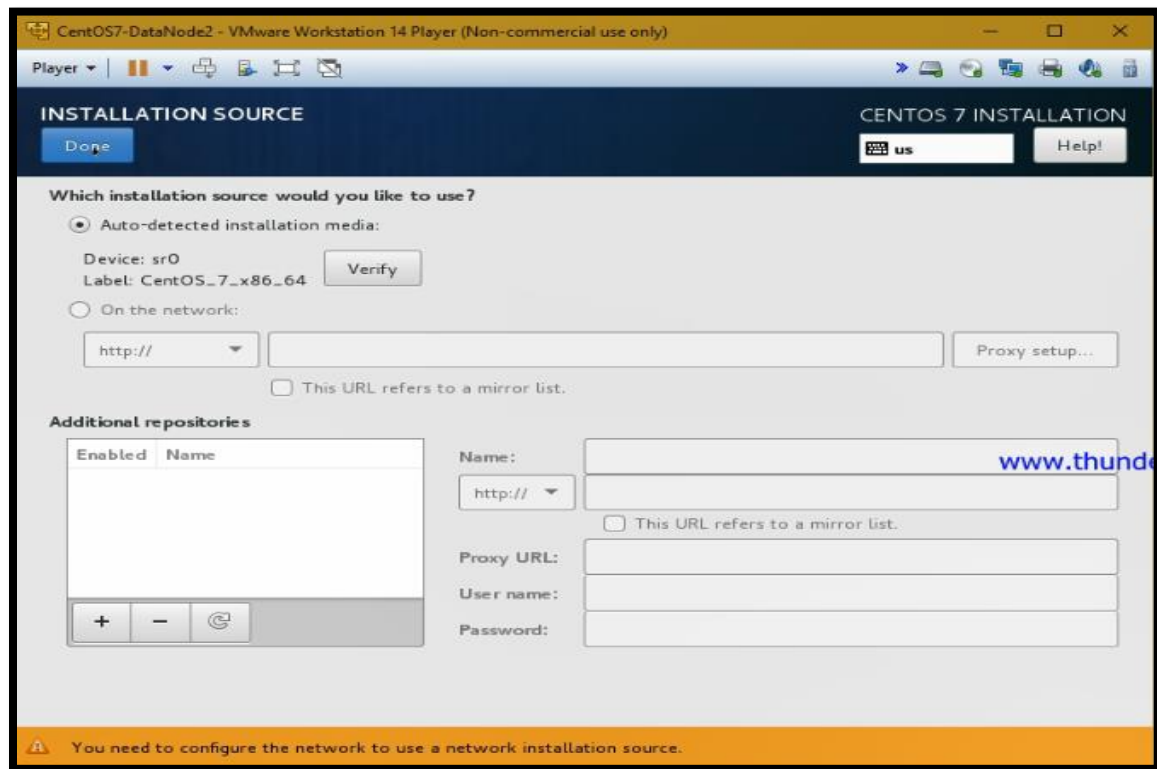


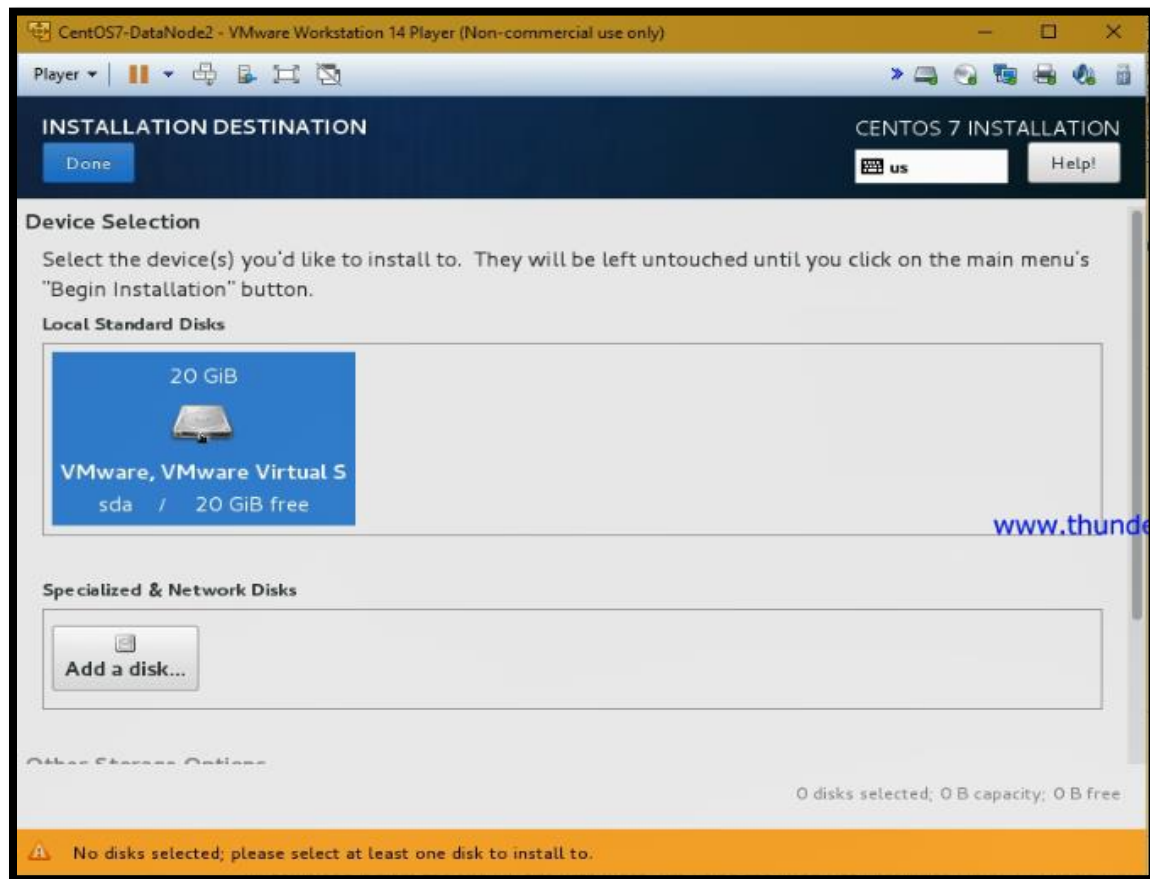


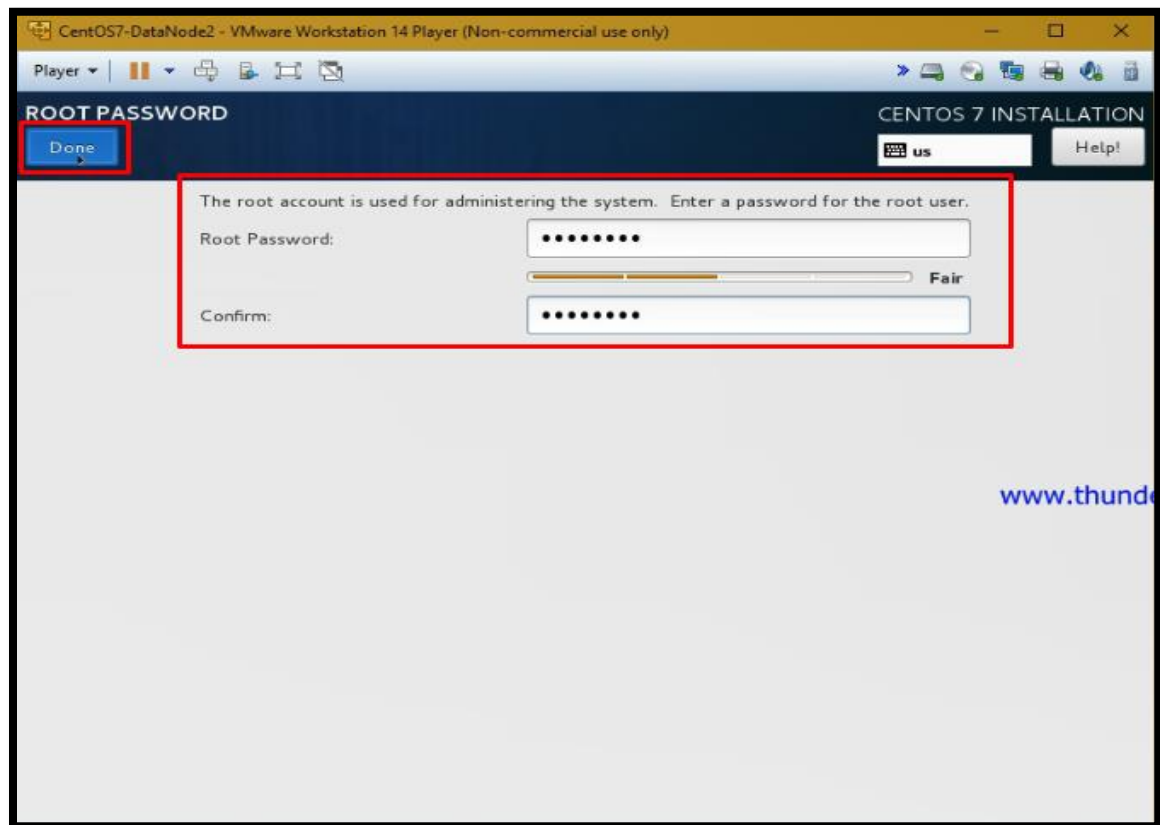
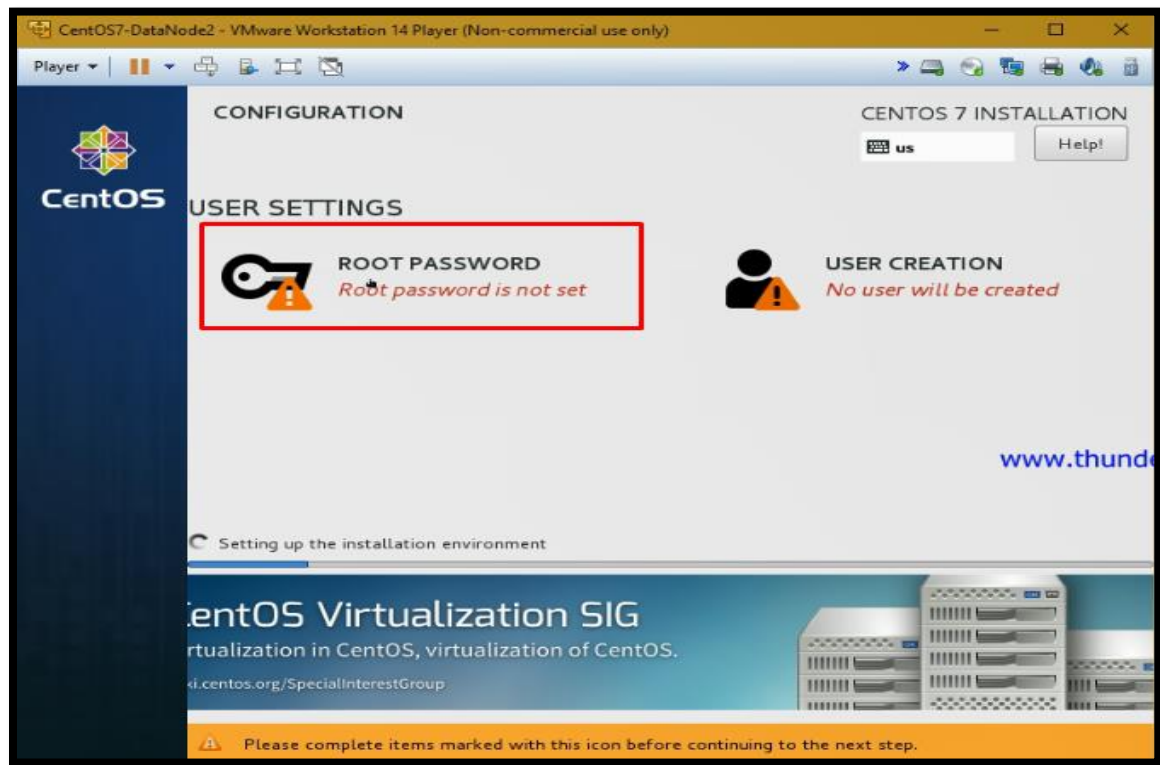


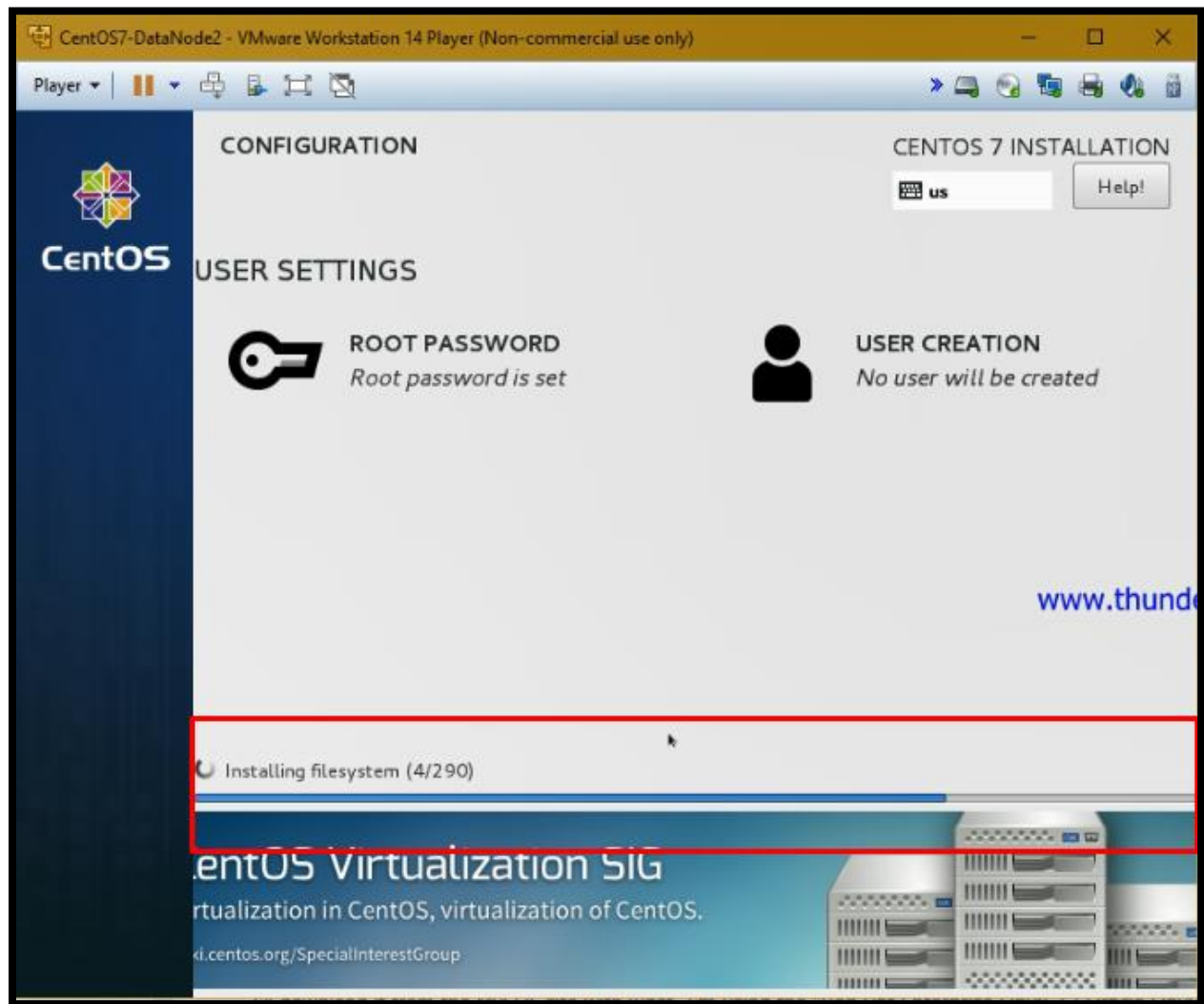












Change the host name before you start the process below to change the hostname as per the desired name follow: <https://github.com/sumanpantha/Linux-Research/blob/master/Change%20HostName> also update all the OS with command: `yum update`.
Syntax: `(root@localhost ~] # hostnamectl set-hostname your-new-hostname)`

For Management Nodes:

```
[root@localhost ~] # hostnamectl set-hostname centosmn
```

For Data Nodes:

```
[root@localhost ~] # hostnamectl set-hostname datanode1
```

```
[root@localhost ~] # hostnamectl set-hostname datanode2
```

For SQL Nodes:

```
[root@localhost ~] # hostnamectl set-hostname sqlnode1
```

```
[root@localhost ~] # hostnamectl set-hostname sqlnode2
```

- Need to Disable SELinux:
To Disable SELinux

```
[root@centos-mn ~] # vi /etc/sysconfig/selinux
```

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Then change the directive **SELinux=enforcing** to **SELinux=disabled** as shown in the below

```
SELINUX=disabled
```

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Then, save and exit the file, for the changes to take effect, you need to reboot your system and then check the status of SELinux using `sestatus` command as shown:

```
[root@centos-mn ~] # sestatus
```

```
[root@centosmn ~]# sestatus
SELinux status:                disabled
[root@centosmn ~]#
```

Step 1 - Setup Management Node

The first step is to create the **Management Node** with CentOS 7 `centosms` and IP `192.168.19.130`. Make sure you are logged into the `datanode1` server as root user.

A. Download the MySQL Cluster software

I'll download it from the MySQL site with `wget` before download we need to install `wget`. I'm using the "CentOS Linux 7 (x86, 64-bit), RPM Bundle " here. Then extract the tar file.

```
[root@centos-mn ~] # yum install wget

[root@centos-mn ~] # wget http://dev.mysql.com/get/Downloads/MySQL-Cluster-7.4/MySQL-Cluster-gpl-7.4.10-1.el7.x86_64.rpm-bundle.tar

[root@centos-mn ~] # tar -xvf MySQL-Cluster-gpl-7.4.10-1.el7.x86_64.rpm-bundle.tar
```

B. Install and Remove Packages

Before you install the rpm package for MySQL Cluster, you need to install *perl-Data-Dumper* that is required by the MySQL-Cluster server. And you need to remove *MariaDB-libs* before we can install MySQL Cluster.

```
[root@centos-mn ~] # yum install perl-Data-Dumper

[root@centos-mn ~] # yum remove mariadb-libs

[root@centos-mn ~] # yum install net-tools
```

C. Install MySQL Cluster

Install MySQL Cluster package with these rpm commands:

```
[root@centos-mn ~] # rpm -Uvh MySQL-Cluster-client-gpl-7.4.10-1.el7.x86_64.rpm

[root@centos-mn ~] # rpm -Uvh MySQL-Cluster-server-gpl-7.4.10-1.el7.x86_64.rpm

[root@centos-mn ~] # rpm -Uvh MySQL-Cluster-shared-gpl-7.4.10-1.el7.x86_64.rpm
```

Make sure there is no error.

D. Configure MySQL Cluster

Create a new directory for the configuration files. I will use the `/var/lib/mysql-cluster` directory.

```
[root@centos-mn ~] # mkdir -p /var/lib/mysql-cluster
```

Then create new configuration file for the cluster management named `config.ini` in the `mysql-cluster` directory.

```
[root@centos-mn ~] # cd /var/lib/mysql-cluster  
[root@centos-mn ~] # vi config.ini
```

Paste the configuration below:

```
[ndb_mgmd default]  
# Directory for MGM node log files  
DataDir=/var/lib/mysql-cluster  
  
[ndb_mgmd]  
#Management Node centosmn  
HostName=192.168.19.130  
  
[ndbd default]  
NoOfReplicas=2      # Number of replicas  
DataMemory=256M     # Memory allocate for data storage  
IndexMemory=128M    # Memory allocate for index storage  
#Directory for Data Node  
DataDir=/var/lib/mysql-cluster  
  
[ndbd]  
#Data Node datanode1  
HostName=192.168.19.129  
  
[ndbd]  
#Data Node datanode2  
HostName=192.168.19.131  
  
[mysqld]  
#SQL Node sqlnode1  
HostName=192.168.19.132  
  
[mysqld]  
#SQL Node sqlnode2  
HostName=192.168.19.133
```

Save the file and exit.

E. Start the Management Node

Next start the management node with the command below:

```
[root@centos-mn ~] # ndb_mgmd --config-file=/var/lib/mysql-cluster/config.ini
```

The result should be similar to this:

```
MySQL Cluster Management Server mysql-5.6.28 ndb-7.4.10
2018-05-24 19:26:08 [MgmtSrvr] INFO      -- The default config directory
'/usr/mysql-cluster' does not exist. Trying to create it...
2018-05-24 19:26:08 [MgmtSrvr] INFO      -- Successfully created config
directory
```

The management node is started, now you can use command **ndb_mgm** to monitor the node:

```
[root@centos-mn ~] # ndb_mgm
ndb_mgm> show
```

```
[root@centosmn mysql-cluster]# ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> show
Connected to Management Server at: localhost:1186
Cluster Configuration
-----
[ndbd(NDB)]      2 node(s)
id=2 (not connected, accepting connect from 192.168.19.129)
id=3 (not connected, accepting connect from 192.168.19.131)

[ndb_mgmd(MGM)]  1 node(s)
id=1 @192.168.19.130 (mysql-5.6.28 ndb-7.4.10)

[mysqld(API)]    2 node(s)
id=4 (not connected, accepting connect from 192.168.19.132)
id=5 (not connected, accepting connect from 192.168.19.133)
```

You can see the management node has been started with: mysql-5.6 and ndb-7.4.

Step 2 - Setup the MySQL Cluster Data Nodes

We will use 2 CentOS servers for the Data Nodes.

1. **datanode1** = 192.168.19.129
2. **datanode2** = 192.168.19.131

A. Login as root user and download the MySQL Cluster software

Login to the **datanode1** server with ssh:

```
[root@centos-mn ~] # ssh root@192.168.19.129
```

A. Login as root user and download the MySQL Cluster software

Login to the **datanode2** server with ssh:

```
[root@centos-mn ~] # ssh root@192.168.19.131
```

Then download the MySQL Cluster package and extract it:

```
[root@datanode1 ~] # yum install wget
```

```
[root@datanode1 ~] # wget http://dev.mysql.com/get/Downloads/MySQL-Cluster-7.4/MySQL-Cluster-gpl-7.4.10-1.el7.x86\_64.rpm-bundle.tar
```

```
[root@datanode1 ~] # tar -xvf MySQL-Cluster-gpl-7.4.10-1.el7.x86_64.rpm-bundle.tar
```

B. Install and Remove Packages

Install perl-Data-Dumper and remove the mariadb-libs also need to install net-tools for the connection between the nodes:

```
[root@datanode1 ~] # yum install perl-Data-Dumper
```

```
[root@datanode1 ~] # yum remove mariadb-libs
```

```
[root@centos-mn ~] # yum install net-tools
```

C. Install MySQL Cluster

Now we can install the MySQL Cluster packages for the Data Nodes with these rpm commands:

```
[root@datanode1 ~] # rpm -Uvh MySQL-Cluster-client-gpl-7.4.10-1.el7.x86_64.rpm

[root@datanode1 ~] # rpm -Uvh MySQL-Cluster-server-gpl-7.4.10-1.el7.x86_64.rpm

[root@datanode1 ~] # rpm -Uvh MySQL-Cluster-shared-gpl-7.4.10-1.el7.x86_64.rpm
```

Make sure there is no error.

D. Configure Data Node

Create a new configuration file in the /etc directory with the vi editor:

```
[root@datanode1 ~] # vi /etc/my.cnf
```

Paste configuration below:

```
[mysqld]
ndbcluster
ndb-connectstring=192.168.19.130      # IP address of Management Node

[mysql_cluster]
ndb-connectstring=192.168.19.130      # IP address of Management Node
```

Save the file and exit.

Then create the new directory for the database data that we defined in the management node config file **config.ini**.

```
[root@datanode1 ~] # mkdir -p /var/lib/mysql-cluster
```

Now start the data node/ndbd:

```
[root@datanode1 ~] # ndbd
```

Results:

datanode1

```
[root@datanode1 ~]# ndbd
2018-05-25 10:17:08 [ndbd] INFO      -- Angel connected to '192.168.19.130:1186'
2018-05-25 10:17:08 [ndbd] INFO      -- Angel allocated nodeid: 2
[root@datanode1 ~]# _
```

datanode2

```
[root@datanode2 ~]# ndbd
2018-05-25 10:16:52 [ndbd] INFO      -- Angel connected to '192.168.19.130:1186'
2018-05-25 10:16:52 [ndbd] INFO      -- Angel allocated nodeid: 3
[root@datanode2 ~]# _
```

Data Node **datanode1** connected to the management node IP **192.168.19.130**.

E. Redo step 2.A - 2.D on db3 server.

As we have 2 data nodes, please redo the steps 2.A - 2.D on our second data node.

Step 3 - Setup SQL Node

This step contains the setup for the SQL Node that provides the application access to the database. We use 2 CentOS servers for the SQL Nodes:

1. **sqlnode1** = 192.168.19.132
2. **sqlnode2** = 192.168.19.133

A. Log in and Download MySQL Cluster

Login to the **sqlnode1** server as **root** user:

```
[root@centosmn ~] # ssh root@192.168.19.132
```

And download MySQL Cluster package:

```
[root@sqlnode1 ~] # yum install wget  
  
[root@sqlnode1 ~] # wget http://dev.mysql.com/get/Downloads/MySQL-Cluster-7.4/MySQL-Cluster-gpl-7.4.10-1.el7.x86\_64.rpm-bundle.tar  
  
[root@sqlnode1 ~] # tar -xvf MySQL-Cluster-gpl-7.4.10-1.el7.x86_64.rpm-bundle.tar
```

B. Install and Remove Packages

Install perl-Data-Dumper and remove the mariadb-libs that conflict with MySQL Cluster.

```
[root@sqlnode1 ~] # yum install perl-Data-Dumper  
  
[root@sqlnode1 ~] # yum remove mariadb-libs  
  
[root@sqlnode1 ~] # yum install net-tools
```

C. Install MySQL Cluster

Install the MySQL Cluster server, client and shared package with the rpm commands below:

```
[root@sqlnode1 ~] # rpm -Uvh MySQL-Cluster-client-gpl-7.4.10-1.el7.x86_64.rpm  
[root@sqlnode1 ~] # rpm -Uvh MySQL-Cluster-server-gpl-7.4.10-1.el7.x86_64.rpm  
[root@sqlnode1 ~] # rpm -Uvh MySQL-Cluster-shared-gpl-7.4.10-1.el7.x86_64.rpm
```

D. Configure the SQL Node

Create a new **my.cnf** file in the **/etc** directory:

```
[root@sqlnode1 ~] # vi /etc/my.cnf
```

And paste configuration below:

```
[mysqld]
ndbcluster
ndb-connectstring=192.168.19.130      # IP address for server management node
default_storage_engine=ndbcluster    # Define default Storage Engine used by MySQL

[mysql_cluster]
ndb-connectstring=192.168.19.130      # IP address for server management node
```

Save the file and exit the editor.

Start the SQL Node by starting the MySQL server:

```
[root@sqlnode1 ~] # service mysql start
```

Start **sqlnode1**

```
[root@sqlnode1 ~]# mysqld_safe &
[1] 1711
[root@sqlnode1 ~]# 180525 22:32:09 mysqld_safe Logging to '/var/lib/mysql/sqlnode1.err'.
180525 22:32:09 mysqld_safe Starting mysqld daemon with databases from /var/lib/mysql
```

Start **sqlnode2**

```
[root@sqlnode2 ~]# mysqld_safe &
[1] 10649
[root@sqlnode2 ~]# 180525 10:36:57 mysqld_safe Logging to '/var/lib/mysql/sqlnode2.err'.
180525 10:36:57 mysqld_safe Starting mysqld daemon with databases from /var/lib/mysql
```

Disable Firewall or allow the port used for MySQL database (3306) and cluster port
(<https://bytfreaks.net/gnulinux/how-to-startstop-or-enabledisable-firewalld-on-centos-7>)
Allow mysql port on both the SQL Nodes **sqlnode1** and **sqlnode2**.

```
[root@sqlnode1 ~] # firewall-cmd --permanent --add-port=3306/tcp

[root@sqlnode1 ~] # firewall-cmd --reload
```

E. Redo step 3.A - 3.D on db5 server.

Please redo the steps 3.A - 3.D on the second SQL server (**sqlnode2**).

Step 4 - Monitor the Cluster

To see the cluster status, we have to log into the management node `centosmn`.

```
ssh root@192.168.19.130
```

We can use the `ndb_mgm` command to see the cluster status:

```
[root@centos-mn ~] # ndb_mgm
ndb_mgm> show
```

```
[root@centosmn ~]# ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> show
Connected to Management Server at: localhost:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2   @192.168.19.129  (mysql-5.6.28 ndb-7.4.10, Nodegroup: 0, *)
id=3   @192.168.19.131  (mysql-5.6.28 ndb-7.4.10, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1   @192.168.19.130  (mysql-5.6.28 ndb-7.4.10)

[mysqld(API)] 2 node(s)
id=4   @192.168.19.132  (mysql-5.6.28 ndb-7.4.10)
id=5   @192.168.19.133  (mysql-5.6.28 ndb-7.4.10)
```

Another useful command is:

```
ndb_mgm -e "all status"
ndb_mgm -e "all report memory"
```

Step 5 - Testing the Cluster

To perform a test on our new MySQL Cluster, we have to login to the SQL Nodes db4 or db5 servers.

Login to the `sqlnode1` server:

```
ssh root@192.168.19.132
```

Change the default MySQL password that stored in `.mysql_secret` file in root directory:

```
[root@sqlnode1 ~] # cat .mysql_secret
```

This is my sample:

```
# The random password set for the root user at Tue Mar 22 19:44:07 2016  
(local time): qna3AwbJMuOnw23T
```

Now change the password with command below:

```
[root@sqlnode1 ~] # mysql_secure_installation
```

Type your old MySQL password and then type the new one, press enter to confirm all.
If all is done, you can login to the MySQL shell with your password:

```
[root@sqlnode1 ~] # mysql -u root -p
```

After you logged in, create a new root user with host "`@`", so we will be able to access the MySQL from outside.

```
mysql> CREATE USER 'root'@'%' IDENTIFIED BY 'admin123';
```

Replace `admin123` with your own secure password! Now you can see the new root user with host "`@`" on the MySQL user list:

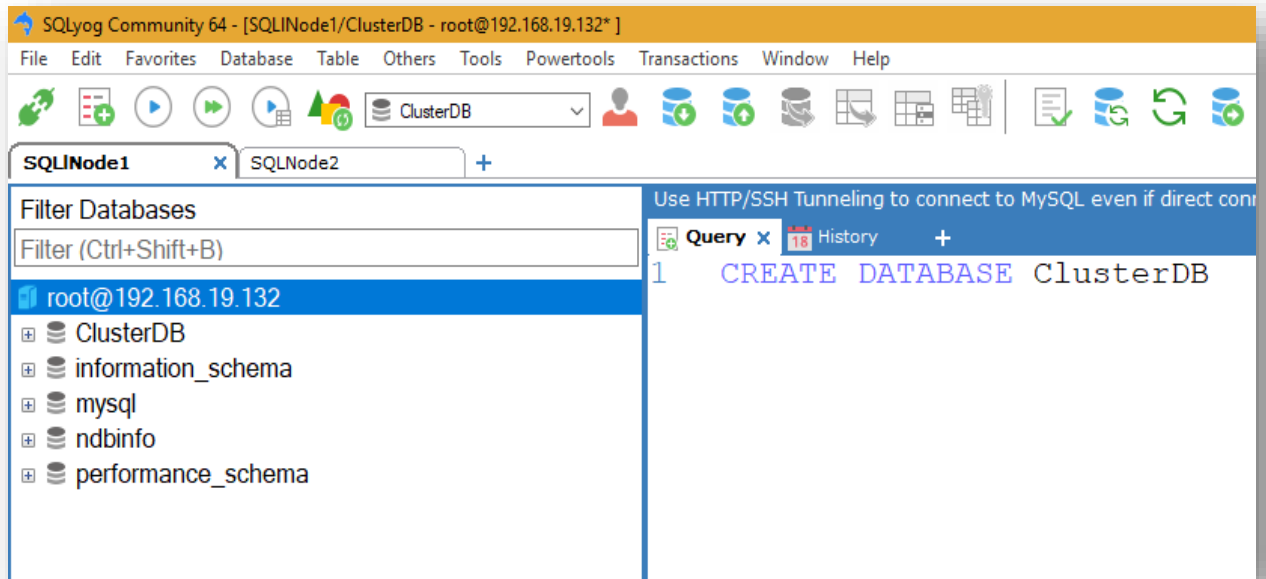
```
mysql> select user, host, password from mysql.user;
```

And grant the new root user read and write access from the remote node:

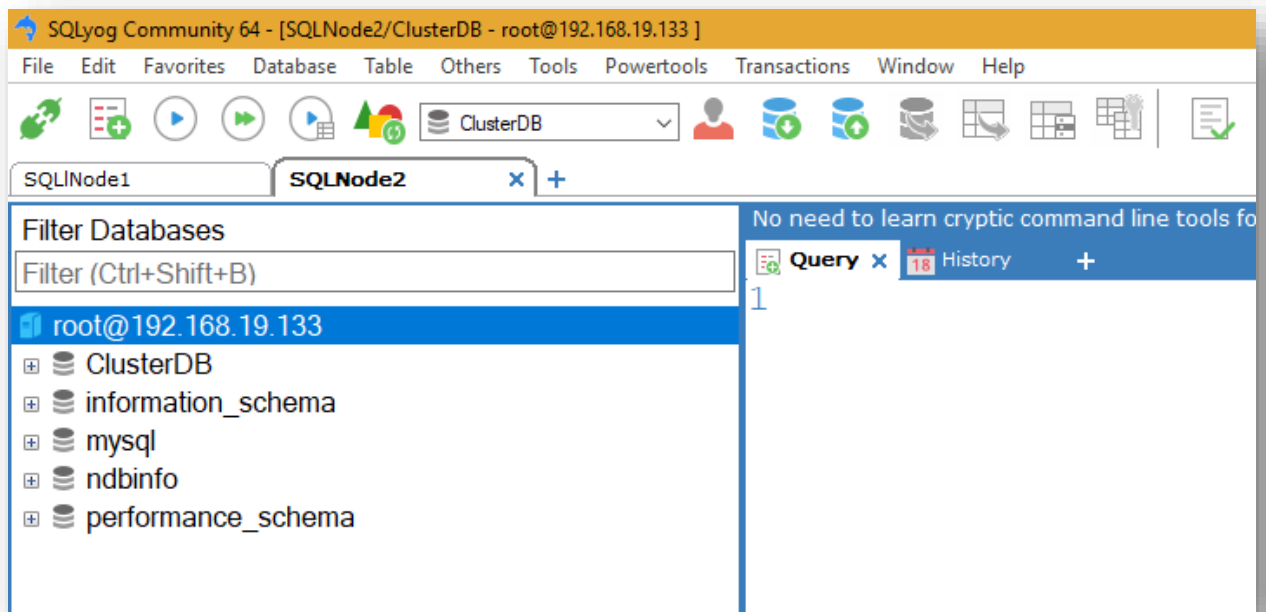
```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY PASSWORD  
'*94CC7BF027327993D738E11...(Encrypted PASSWORD)' WITH GRANT OPTION;
```

Now try to create a new database(ClusterDB) from **sqlnode1** server and you will see the database on **sqlnode2** too. This is just a sample result for testing the cluster data replication.

SQLNODE1



SQLNODE2



The MySQL Cluster has been setup successfully on CentOS 7 with 5 server nodes.

Conclusion

MySQL Cluster is a technology that provides High Availability and Redundancy for MySQL databases. It uses NDB or NDBCLUSTER as the storage engine and provides shared-nothing clustering and auto-sharing for MySQL databases. To implement the cluster, we need 3 components: Management Node(MGM), Data Nodes (NDB) and SQL Nodes (API). Each of node must have its own memory and disk. It is not recommended to use network storage such as NFS. To install MySQL Cluster on a CentOS 7 minimal system, we have to remove the mariadb-libs package, mariadb-libs conflict with MySQL-Cluster-server and you have to install the perl-Data-Dumper package, it's needed by MySQL-Cluster-server. A MySQL Cluster is easy to install and configure on multiple CentOS servers.