Prerequisite to add a new node

- To have the same cluster name
- IP address and access to at least one node

Common ports in Cassandra

- 7000 - Node communication
- 9160 - Thrift
- 7199 - JMX monitoring
- 9042 - Native client

Step 1 - Check the IP address to be added machine

```
ifconfig
```

```
[ubuntu@primary:~$ ifconfig                                                    ]
enp0s1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.64.2  netmask 255.255.255.0  broadcast 192.168.64.255
        inet6 fe80::5054:ff:fe77:ba12  prefixlen 64  scopeid 0x20<link>
        inet6 fd16:f2cb:163f:ba0d:5054:ff:fe77:ba12  prefixlen 64  scopeid 0x0<g
lobal>
        ether 52:54:00:77:ba:12  txqueuelen 1000  (Ethernet)
        RX packets 21540  bytes 13185624 (13.1 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13173  bytes 1554242 (1.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 7642  bytes 2938841 (2.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7642  bytes 2938841 (2.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

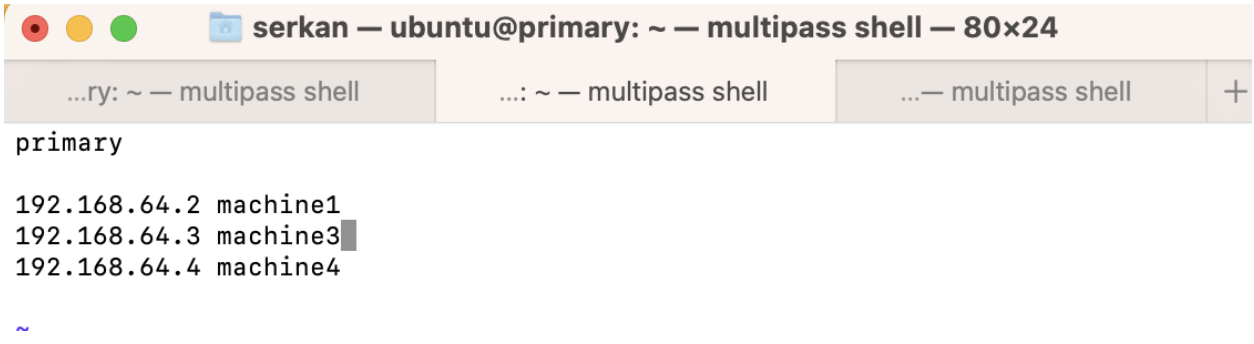
Step 2 - Open the node in the cluster, and check whether there is a network connectivity

```
ping 192.168.64.2
```

```
ubuntu@primary:~$ ping 192.168.64.2
PING 192.168.64.2 (192.168.64.2) 56(84) bytes of data.
64 bytes from 192.168.64.2: icmp_seq=1 ttl=64 time=0.063 ms
64 bytes from 192.168.64.2: icmp_seq=2 ttl=64 time=0.122 ms
64 bytes from 192.168.64.2: icmp_seq=3 ttl=64 time=0.125 ms
^C
--- 192.168.64.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2043ms
rtt min/avg/max/mdev = 0.063/0.103/0.125/0.028 ms
```

Step 3 - Define the hostname

```
sudo vim /etc/hostname
```



Define same list to other machines

Step 4 - Check whether the hostname is working

```
ubuntu@primary:~$ ping machine1
```

Step 5 - Install Apache Cassandra

https://cassandra.apache.org/doc/latest/cassandra/getting_started/installing.html

Step 6 - Change cluster name from cassandra yaml

```
vim conf/cassandra.yaml
```

```
# Cassandra storage config YAML

# NOTE:
#   See https://cassandra.apache.org/doc/latest/configuration/
#   full explanations of configuration directives
# /NOTE

# The name of the cluster. This is mainly used to prevent mach
# one logical cluster from joining another.
cluster_name: 'Test Cluster'
```

Step 7 - Change the listen_address conf to `192.168.64.2`

```
# Set listen_address OR listen_interface, not both.
#
# Leaving it blank leaves it up to InetAddress.getLocalHost(). This
# will always do the Right Thing _if_ the node is properly configured
# (hostname, name resolution, etc), and the Right Thing is to use the
# address associated with the hostname (it might not be). If unresolv
# it will fall back to InetAddress.getLoopbackAddress(), which is wro
uction systems.
#
# Setting listen_address to 0.0.0.0 is always wrong.
#
listen_address: localhost

/listen_address
```

Step 7 - Change the rpc_address conf to `192.168.64.2`

```
# Set rpc_address OR rpc_interface, not both.
#
# Leaving rpc_address blank has the same effect as on listen_address
# (i.e. it will be based on the configured hostname of the node).
#
# Note that unlike listen_address, you can specify 0.0.0.0, but you must also
# set broadcast_rpc_address to a value other than 0.0.0.0.
#
# For security reasons, you should not expose this port to the internet.  Fir
ll it if needed.
rpc_address: localhost
```