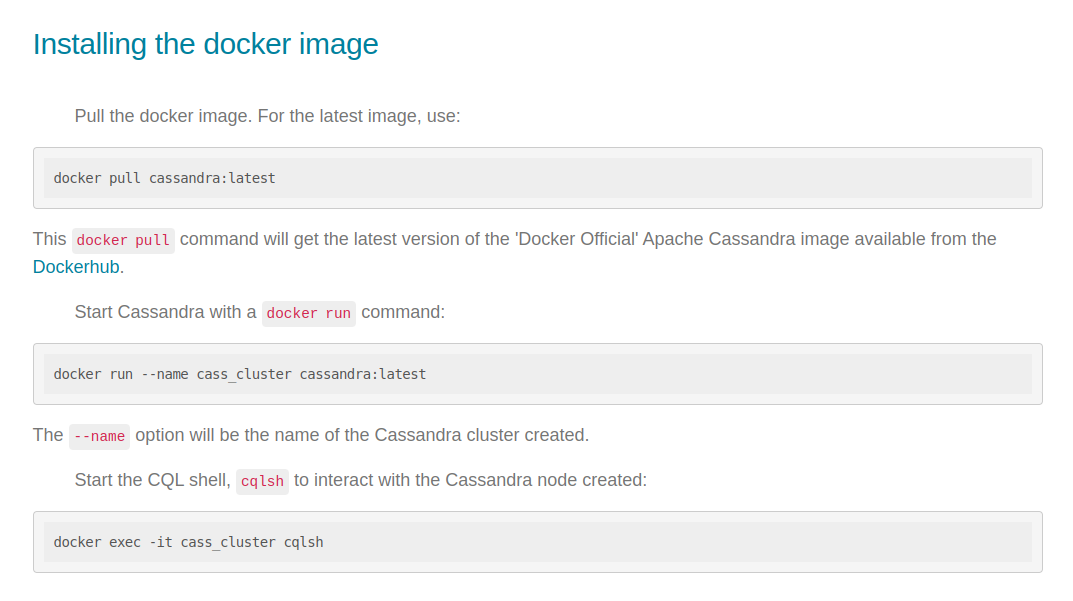
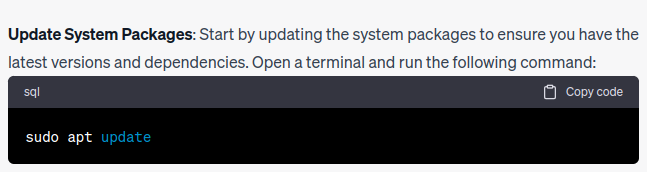
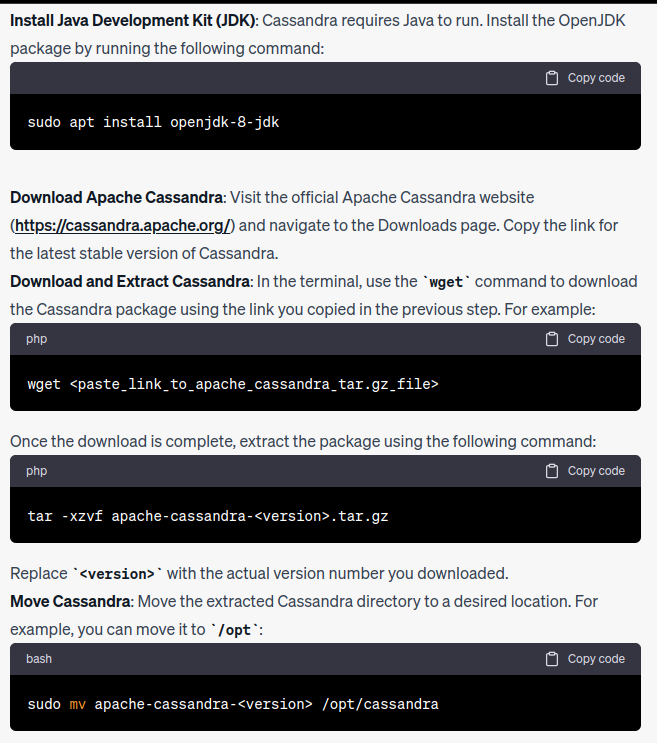
**How to setup cassandra cluster**

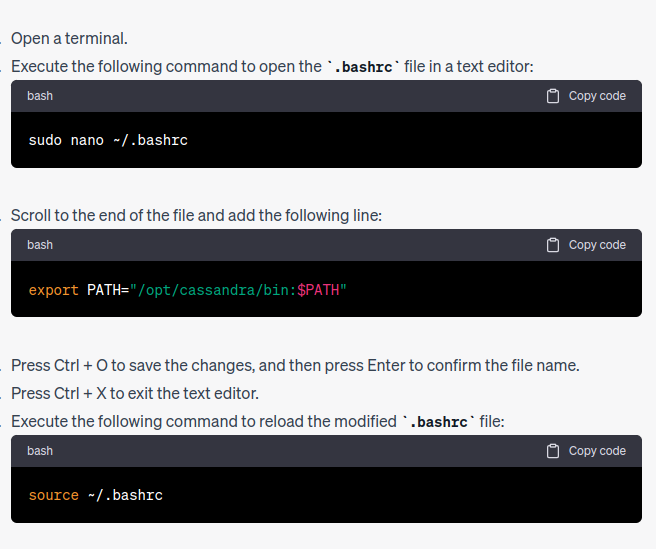
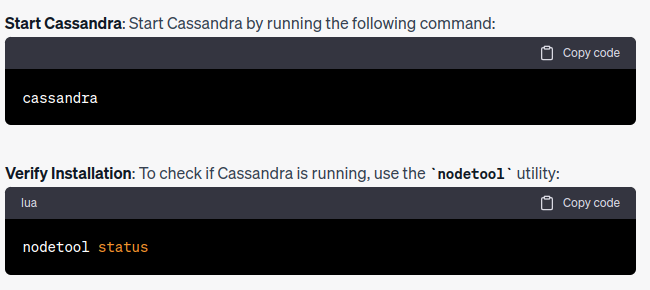
**Single node cluster:**

For a single node cluster running on a single machine we can just make use of docker and pull cassandra image using following commands  
  


**Multi node cluster:**

**Step 1**: install and run cassandra single node cluster using the following steps  


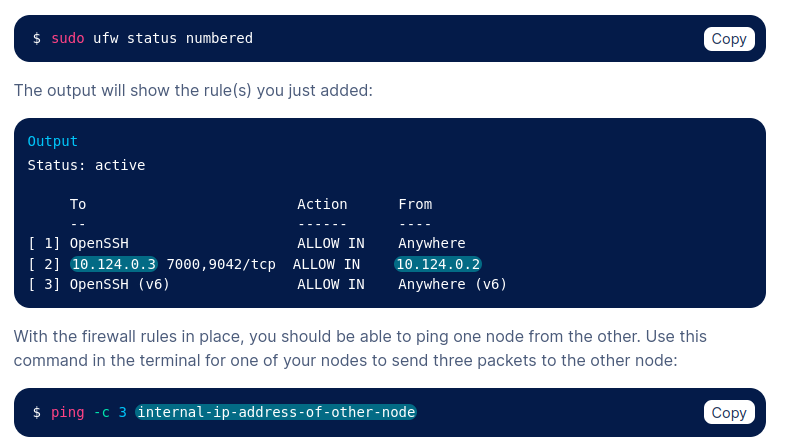
****

**  
  
  
  
Step 2**: Configuring the firewall to allow Cassandra traffic  
  
**7000** is the TCP port for commands and data.

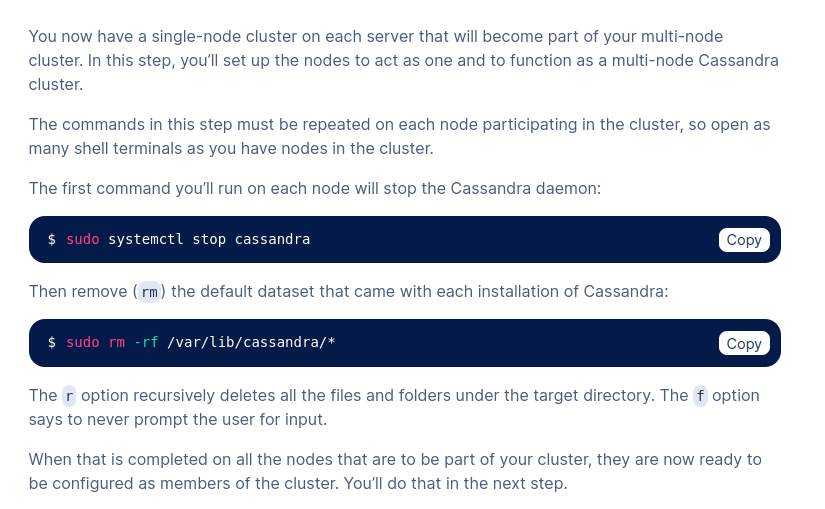
**9042** is the TCP port for the native transport server. The Cassandra command line utility, cqlsh, will connect to the cluster through this port.

For this tutorial, the first node will be called node1, and the second node will be called node2. Where the prompts refer to node1-internal-ip-address or node2-internal-ip-address, replace that section with the IP addresses you just extracted.  
  
On node1, execute the following command:

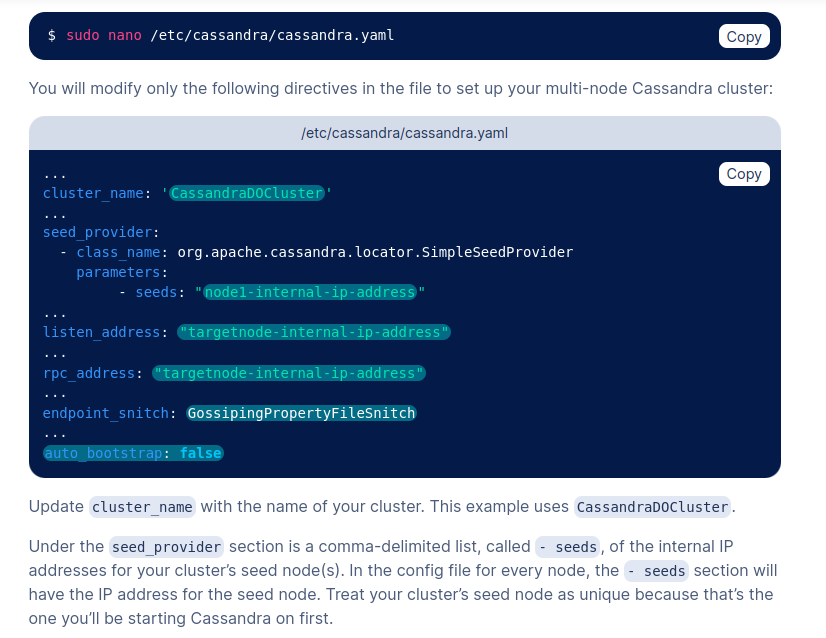
sudo ufw allow from node2-internal-ip-address to node1-internal-ip-address proto tcp port 7000,9042

On node2, reverse the IP addresses like so:  
  
sudo ufw allow from node1-internal-ip-address to node2-internal-ip-address proto tcp port 7000,9042  
  
Repeat the command for as many nodes in your cluster, only changing the sequence of IP addresses. If you have N nodes in your cluster, you will need to run N - 1 of that command on each node.  
The rules take effect immediately, so you don’t need to reload the firewall. You can view the firewall rules on each node with the following command:  
  


**Step 3**: Deleting Cassandra’s pre-installed data

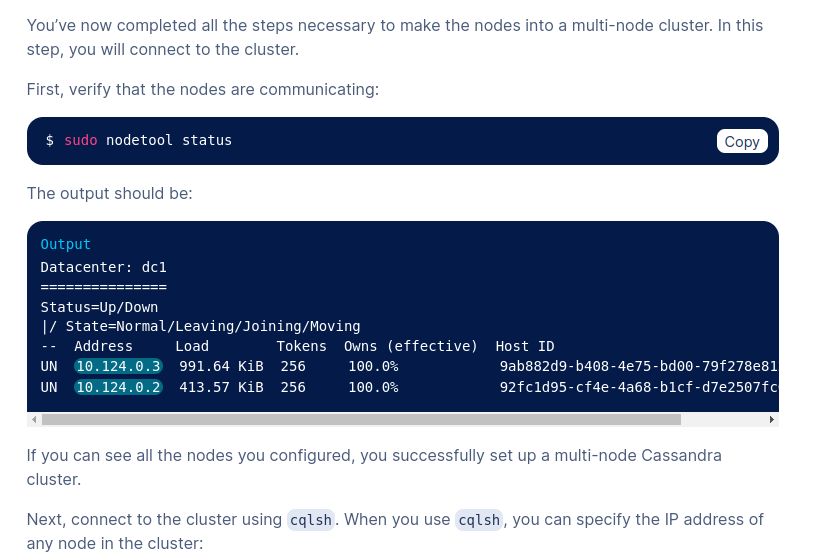


**Step 4**: Configuring the Cassandra Cluster

The /opt/cassandra/conf/cassandra.yaml configuration file contains many directives and is very well commented. Starting with your designated seed node (node1 in this tutorial), open the configuration file using:  
  
  
  
The listen\_address and rpc\_address default to localhost, but both need to be changed to the internal IP address of the target node. For the file on node1, you will put the same node1-internal-ip-address in all three places. For the file on node2, you will put the node1-internal-ip-address under seeds and use the node2-internal-ip-address for the listen\_address and rpc\_address. Do this for all nodes on your cluster.  
  
The [endpoint\_snitch](https://cassandra.apache.org/doc/trunk/cassandra/configuration/cass_yaml_file.html#endpoint_snitch) gives the name of a snitch class that will be used for locating nodes and routing requests within your Cassandra cluster. By default, it is set to [SimpleSnitch](https://cassandra.apache.org/doc/trunk/cassandra/configuration/cass_yaml_file.html#endpoint_snitch), which will only work for a Cassandra cluster within a single datacenter. For production deployments, [GossipingPropertyFileSnitch](https://cassandra.apache.org/doc/trunk/cassandra/configuration/cass_rackdc_file.html#gossipingpropertyfilesnitch) is recommended.  
  
The auto\_bootstrap directive is not in the configuration file, so it will need to be added and set to false. It is optional if you’re adding nodes to an existing cluster but required when you’re initializing a new cluster (one with no data).When you’re finished modifying the file, save and close it. Repeat this step for all the servers you want to include in the cluster, ensuring that the list of seed node(s) is the same and that the listen\_address and rpc\_address match the internal IP address of the target node.

**Step 4**: Restart Cassandra



**Step 5:** Connecting to Multi-node Cassandra Cluster  
  
  
  
**Multi node cluster on local machine using dockers:  
  
# Run the first node and keep it in background up and running**

**docker run --name cassandra-1 -p 9042:9042 -d cassandra**

**$INSTANCE1=$(docker inspect --format="{{ .NetworkSettings.IPAddress }}" cassandra-1)**

**echo "Instance 1: ${INSTANCE1}"**

**# Run the second node**

**docker run --name cassandra-2 -p 9043:9042 -d -e CASSANDRA\_SEEDS=$INSTANCE1 cassandra**

**$INSTANCE2=$(docker inspect --format="{{ .NetworkSettings.IPAddress }}" cassandra-2)**

**echo "Instance 2: ${INSTANCE2}"**

**echo "Wait 60s until the second node joins the cluster"**

**sleep 60**

**# Run the third node**

**docker run --name cassandra-3 -p 9044:9042 -d -e CASSANDRA\_SEEDS=$INSTANCE1,$INSTANCE2 cassandra**

**$INSTANCE3=$(docker inspect --format="{{ .NetworkSettings.IPAddress }}" cassandra-3)**