

In this document, I record the 'Oracle cloud specific' issues that I encountered while configuring our Hadoop cluster.

### **Only private IP addresses**

Public elastic IP addresses won't work for Hadoop connections on Oracle cloud instances.

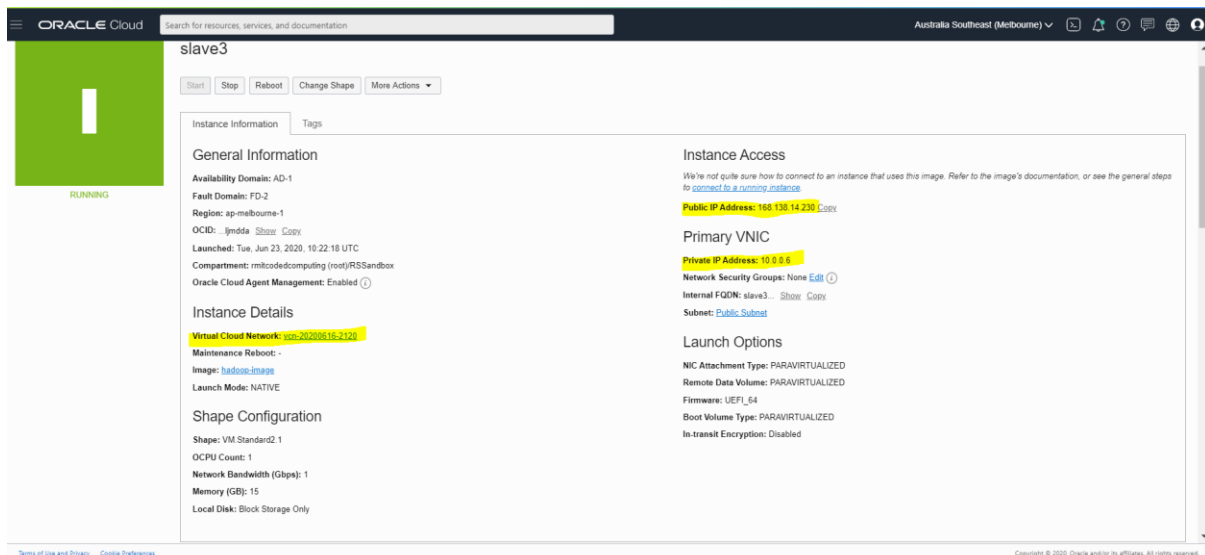
(Keeping public IP address was making the name node unable to bind on port 9000, throwing java.net.BindException).

I struggled with this for a couple of days until finally got a hint from apache's wiki on Hadoop exceptions:

<https://cwiki.apache.org/confluence/display/HADOOP2/BindException>

I tried changing public IP addresses to private IPs in /etc/hosts file, and it worked.

Private IPs can be found on the instance's details page as shown below:



### **No 127.0.1.1 entry in /etc/hosts file**

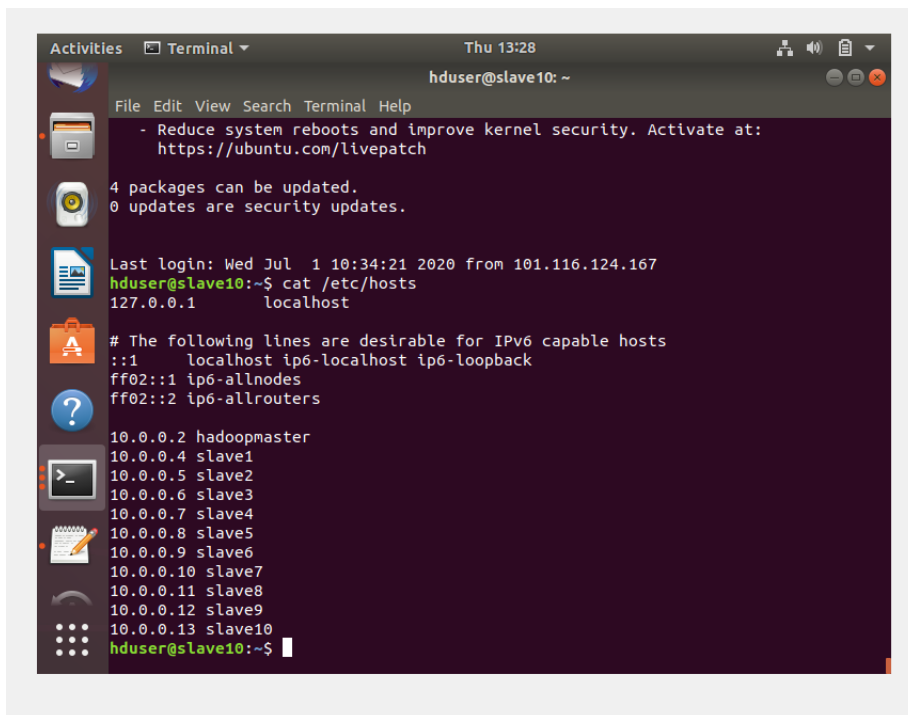
Also, delete the default entry 127.0.1.1 from /etc/hosts file on master and slaves.

(Keeping it was making master node listen on 127.0.1.1 instead of its IP)

Hint from this stack overflow discussion:

<https://stackoverflow.com/questions/8872807/hadoop-datanodes-cannot-find-namenode>

Snap shot of the /etc/hosts file (same for all nodes)

A terminal window titled 'Terminal' showing the configuration of a slave node. The user is 'hduser' on 'slave10'. The terminal displays the contents of the /etc/hosts file, which lists various IP addresses and their corresponding hostnames, including localhost, hadoopmaster, and several slave nodes (slave1 through slave10). The terminal also shows a message about system reboots and kernel security, and a notification that 4 packages can be updated.

```
Thu 13:28
hduser@slave10: ~

File Edit View Search Terminal Help

- Reduce system reboots and improve kernel security. Activate at:
  https://ubuntu.com/livepatch

4 packages can be updated.
0 updates are security updates.

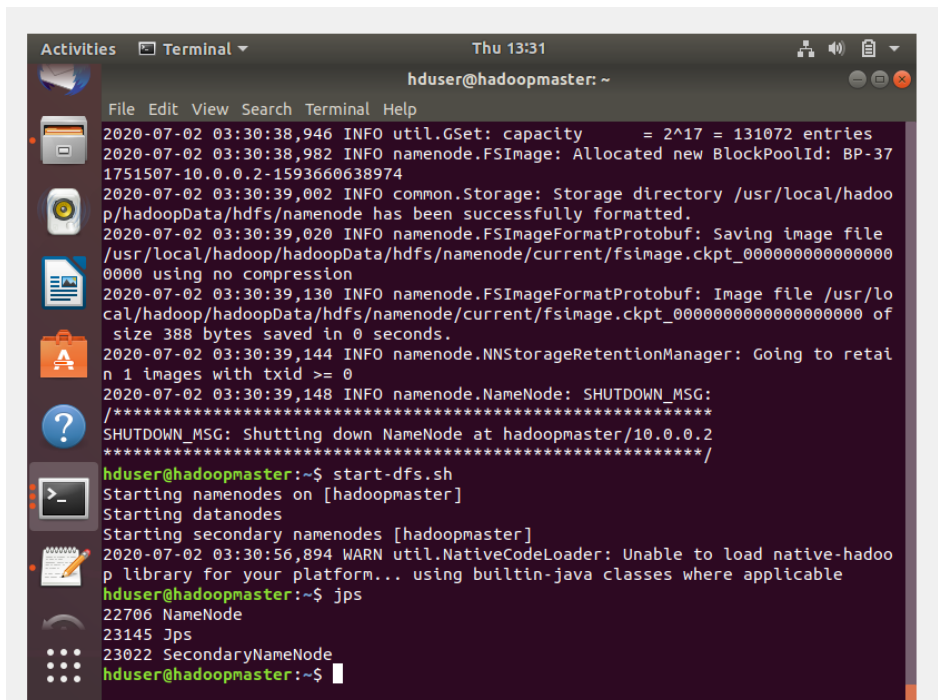
Last login: Wed Jul  1 10:34:21 2020 from 101.116.124.167
hduser@slave10:~$ cat /etc/hosts
127.0.0.1        localhost

# The following lines are desirable for IPv6 capable hosts
::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters

10.0.0.2        hadoopmaster
10.0.0.4        slave1
10.0.0.5        slave2
10.0.0.6        slave3
10.0.0.7        slave4
10.0.0.8        slave5
10.0.0.9        slave6
10.0.0.10       slave7
10.0.0.11       slave8
10.0.0.12       slave9
10.0.0.13       slave10
hduser@slave10:~$
```

### Data nodes unable to connect to master node on port 9000

Name node on the master was up and listening on port 9000, but slave nodes were not able to connect to it. (see the screenshots)

A terminal window titled 'Terminal' showing the configuration of a master node. The user is 'hduser' on 'hadoopmaster'. The terminal displays the output of the 'start-dfs.sh' script, which includes messages about the capacity of the util.GSet, the allocation of a new BlockPoolId, the successful formatting of the namenode, the saving of the fsimage file, and the shutdown of the NameNode. The user then runs 'jps' to check the status of the NameNode and SecondaryNameNode.

```
Thu 13:31
hduser@hadoopmaster: ~

File Edit View Search Terminal Help

2020-07-02 03:30:38,946 INFO util.GSet: capacity = 2^17 = 131072 entries
2020-07-02 03:30:38,982 INFO namenode.FSImage: Allocated new BlockPoolId: BP-37
1751507-10.0.0.2-1593660638974
2020-07-02 03:30:39,002 INFO common.Storage: Storage directory /usr/local/hadoo
p/hadoopData/hdfs/namenode has been successfully formatted.
2020-07-02 03:30:39,020 INFO namenode.FSImageFormatProtobuf: Saving image file
/usr/local/hadoop/hadoopData/hdfs/namenode/current/fsimage.ckpt_0000000000000000
0000 using no compression
2020-07-02 03:30:39,130 INFO namenode.FSImageFormatProtobuf: Image file /usr/lo
cal/hadoop/hadoopData/hdfs/namenode/current/fsimage.ckpt_00000000000000000000
of size 388 bytes saved in 0 seconds.
2020-07-02 03:30:39,144 INFO namenode.NNStorageRetentionManager: Going to retai
n 1 images with txid >= 0
2020-07-02 03:30:39,148 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at hadoopmaster/10.0.0.2
*****/
hduser@hadoopmaster:~$ start-dfs.sh
Starting namenodes on [hadoopmaster]
Starting datanodes
Starting secondary namenodes [hadoopmaster]
2020-07-02 03:30:56,894 WARN util.NativeCodeLoader: Unable to load native-hadoo
p library for your platform... using builtin-java classes where applicable
hduser@hadoopmaster:~$ jps
22706 NameNode
23145 Jps
23022 SecondaryNameNode
hduser@hadoopmaster:~$
```

```

hduser@hadoopmaster: ~
File Edit View Search Terminal Help
hduser@hadoopmaster:~$ sudo netstat -tulnp | grep LISTEN
tcp        0      0 10.0.0.2:9000        0.0.0.0:*           LISTEN
22706/java
tcp        0      0 0.0.0.0:9868        0.0.0.0:*           LISTEN
23022/java
tcp        0      0 0.0.0.0:9870        0.0.0.0:*           LISTEN
22706/java
tcp        0      0 0.0.0.0:111         0.0.0.0:*           LISTEN
759/rpcbind
tcp        0      0 127.0.0.53:53       0.0.0.0:*           LISTEN
917/systemd-resolve
tcp        0      0 0.0.0.0:22         0.0.0.0:*           LISTEN
1135/sshd
tcp6       0      0 :::111              :::*                LISTEN
759/rpcbind
tcp6       0      0 :::22              :::*                LISTEN
1135/sshd
hduser@hadoopmaster:~$

```

I first tried creating security rules in the VCN, enabling the ports that Hadoop uses.

Ingress rules specify rules for letting in incoming traffic through the boundary of a network and egress rules are for network traffic going out.

By default, egress is enabled for all traffic for all ports on Oracle. So, we don't have to change anything with egress.

I added an Ingress rule first, enabling specific ports needed by the master node and slave nodes.

It did not sort the connectivity issue.

Then, I enabled all TCP ports for all ingress traffic, hoping it works this way. (see the screenshot)

Default Security List for vcn-20200616-2120

Instance traffic is controlled by firewall rules on each Instance in addition to this Security List

Security List Information

OCID: ...hram2a Show Copy  
Created: Tue, Jun 16, 2020, 11:23:03 UTC

Compartment: RSSandbox

Ingress Rules

	Stateless	Source	IP Protocol	Source Port Range	Destination Port Range	Type and Code	Allows	Description
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	22		TCP traffic for ports: 22 SSH Remote Login Protocol	
<input type="checkbox"/>	No	0.0.0.0/0	ICMP			3, 4	ICMP traffic for: 3, 4 Destination Unreachable, Fragmentation Needed and Don't Fragment was a Set	
<input type="checkbox"/>	No	10.0.0.0/16	ICMP			3	ICMP traffic for: 3 Destination Unreachable	
<input type="checkbox"/>	No	0.0.0.0/0	TCP	All	All		TCP traffic for ports: All	hadoop nodes
<input type="checkbox"/>	No	0.0.0.0/0	ICMP			8	ICMP traffic for: 8 Echo	ping enable

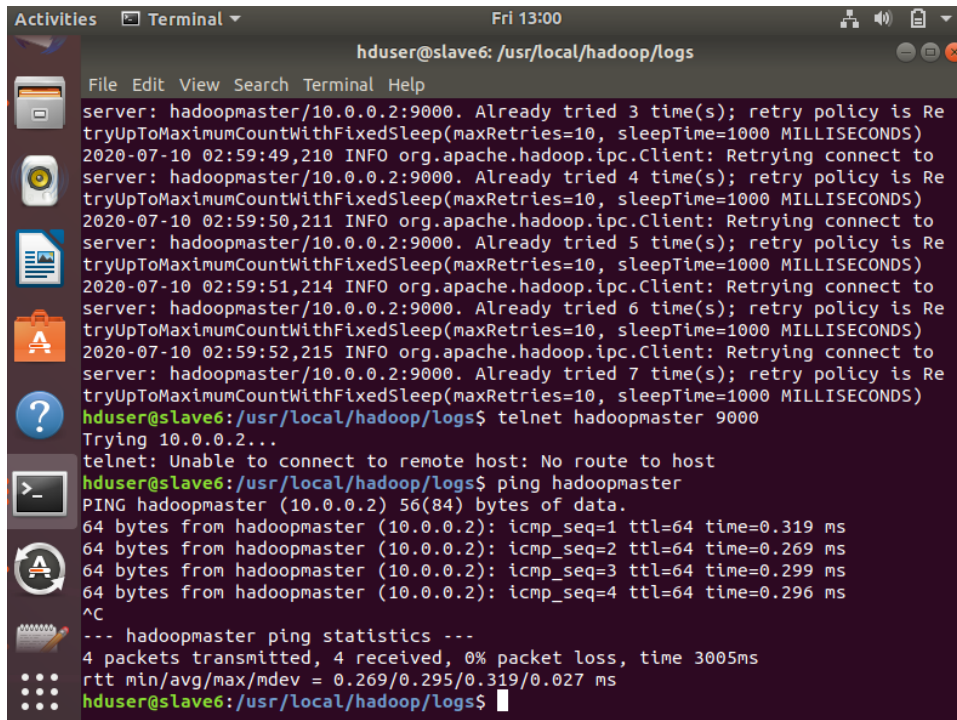
0 Selected

Showing 5 items < 1 of 1 >

Still, the issue persisted.

Telnet and traceroute from datanode to master were not working as well.

See the output of TELNET below:

A screenshot of a terminal window titled 'hduser@slave6: /usr/local/hadoop/logs'. The window shows a series of log messages from the Hadoop NameNode, indicating that the DataNode is retrying to connect to the NameNode at 10.0.0.2:9000. The logs show the retry policy and the number of retries (up to 7). After the logs, the user enters the command 'telnet hadoopmaster 9000', which results in 'Trying 10.0.0.2...' and 'telnet: Unable to connect to remote host: No route to host'. Then, the user enters 'ping hadoopmaster', which shows successful ping results with 4 packets transmitted and received, 0% packet loss, and a round-trip time of approximately 0.269ms. The terminal window also shows a sidebar with various application icons and a top bar with system information like 'Fri 13:00'.

### **Reporting the issue via an SR to Oracle cloud support team**

I decided to contact tech support and they suggested to file a Service Request (SR).

Before raising a technical SR, I posted this question to the Oracle community. I did not get a reply.

I decided to file a technical SR, uploading as much as detail as possible so that I am explaining my issue clearly to the SR team.

After about a week, they got back saying that since Oracle doesn't own the product Hadoop, they can't provide support with this issue.

They gave me an option to redirect the request to another team to check if there is a better team to help with this issue. I went with that option.

### **'iptables' is the solution!**

Finally, one support guy asked me to check the entries in the 'iptables' of the master node.

Kindly note: 'iptables' is a command-line firewall utility in which one can specify policies to allow or reject network traffic.

Commands for working with 'iptables' in Ubuntu:

<https://help.ubuntu.com/community/IptablesHowTo>

I added a new rule to permit port 9000 TCP traffic.

(see the screenshot after adding the rule)

```
Chain INPUT (policy ACCEPT)
target prot opt source destination state
ACCEPT all -- anywhere anywhere state RELATED,ESTABLISHED
ACCEPT icmp -- anywhere anywhere
ACCEPT all -- anywhere anywhere
ACCEPT udp -- anywhere anywhere udp spt:ntp
ACCEPT tcp -- anywhere anywhere state NEW tcp dpt:ssh
REJECT all -- anywhere anywhere reject-with icmp-host-prohibited
ACCEPT tcp -- anywhere anywhere tcp dpt:9000

Chain FORWARD (policy ACCEPT)
target prot opt source destination
REJECT all -- anywhere anywhere reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target prot opt source destination
InstanceServices all -- anywhere link-local/16

Chain InstanceServices (1 references)
target prot opt source destination
ACCEPT tcp -- anywhere 169.254.0.2 owner UID match root tcp dpt:iscsi-target /* See the Oracle-Provided Images section in the Oracle Cloud Infrastructure document
ACCEPT tcp -- anywhere 169.254.2.0/24 owner UID match root tcp dpt:iscsi-target /* See the Oracle-Provided Images section in the Oracle Cloud Infrastructure document
ACCEPT tcp -- anywhere 169.254.0.2 tcp dpt:http /* See the Oracle-Provided Images section in the Oracle Cloud Infrastructure documentation for security impact of
ACCEPT udp -- anywhere 169.254.169.254 udp dpt:domain /* See the Oracle-Provided Images section in the Oracle Cloud Infrastructure documentation for security impact c
ACCEPT tcp -- anywhere 169.254.169.254 tcp dpt:domain /* See the Oracle-Provided Images section in the Oracle Cloud Infrastructure documentation for security impact c
ACCEPT tcp -- anywhere 169.254.0.3 owner UID match root tcp dpt:http /* See the Oracle-Provided Images section in the Oracle Cloud Infrastructure documentation fc
ACCEPT tcp -- anywhere 169.254.0.4 tcp dpt:http /* See the Oracle-Provided Images section in the Oracle Cloud Infrastructure documentation for security impact of
ACCEPT tcp -- anywhere 169.254.169.254 tcp dpt:http /* See the Oracle-Provided Images section in the Oracle Cloud Infrastructure documentation for security impact of
ACCEPT udp -- anywhere 169.254.169.254 udp dpt:bootps /* See the Oracle-Provided Images section in the Oracle cloud Infrastructure documentation for security impact c
ACCEPT udp -- anywhere 169.254.169.254 udp dpt:tftp /* See the Oracle-Provided Images section in the Oracle Cloud Infrastructure documentation for security impact of
ACCEPT udp -- anywhere 169.254.169.254 udp dpt:ntp /* See the Oracle-Provided Images section in the Oracle Cloud Infrastructure documentation for security impact of a
REJECT tcp -- anywhere link-local/16 tcp /* See the Oracle-Provided Images section in the Oracle Cloud Infrastructure documentation for security impact of modifying
REJECT udp -- anywhere link-local/16 udp /* See the Oracle-Provided Images section in the Oracle Cloud Infrastructure documentation for security impact of modifying
```

Even after this, the connectivity issue was still there.

I deleted the highlighted entires (2 REJECT rules) assuming that they might be causing the issue.

Still, no success. I was reporting this to the tech support.

I was then suggested to shift the rule for port 9000 to before the REJECT rule.

To insert a rule to the top of a chain, please see the web page:

<https://www.cyberciti.biz/faq/linux-iptables-insert-rule-at-top-of-tables-prepend-rule/>

I did this. Like a magic, now data nodes started connecting to the namenode.

Later, I also added rules for enabling extra ports on the masternode for yarn, mapreduce and web UI (8025, 8030, 8050, 54311, 9870) and some ports (9864, 9865, 9866, 9867) on the data node.

See the below web site for the default ports used by Hadoop 3:

<https://www.stefaanlippens.net/hadoop-3-default-ports.html>

We need to have all nodes that are needed by master and data nodes to be enabled separately on their respective iptables.

New separate images for Hadoop master node and Hadoop slave node have been created on our Oracle cloud tenancy.

(Earlier, we had a single Hadoop image)

These can be used for spawning new master nodes and data nodes as and when we require.