



## Introduction

The purpose of this document is to describe how to stand-up a Carbon Black cluster containing a Master node and any number of Minion nodes.

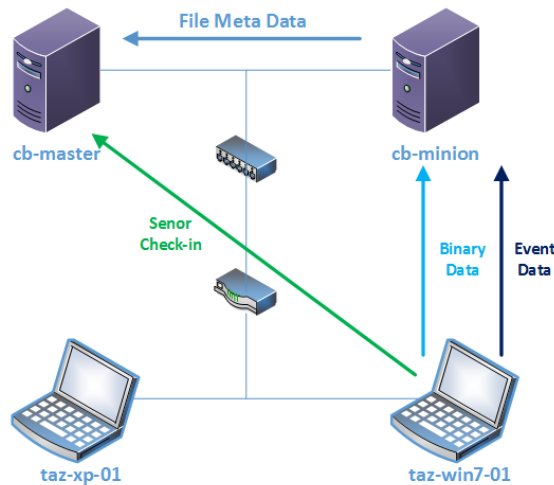


Figure 1 Original Configuration

Figure 1 represents the network utilized for this document with two generic Carbon Black server installs. They have no client software configured on them initially. The goal is to create a Carbon Black Cluster with *cb-master* as the master node and *cb-minion* as the minion node. The client sensor install software will be pulled from *cb-master*. The initial *cbinit* ran on the master node should be ran utilizing the `--proc-store-shards=PROC_STORE_SHARDS` where `PROC_STORE_SHARDS` is the number of SOLR shards. The default is one; however, on larger implementations at least two shards per minion should be utilized and one minion per 5K nodes. Using this example a 10K node network would require one master node and two minion nodes. The number of shards for the master node creation would be 4 with shards equally distributed across the minion nodes resulting in 2 shards per minion.

## Sections

### Cluster Configuration

The assumption for this section is that you have installed the Carbon Black RPM and ran the “yum install cb-enterprise” command on the master node. The minion node is simply a generic install of CentOS.

1. On the master node run *cbinit* with the correct number of shards defined based on your installation.

In our example that would be 2.

```
/usr/share/cb/cbinit --proc-store-shards=2
```

2. Command options for `cb_cluster` command utilized to initiate cluster configuration.

```
[root@cb-master ~]# /usr/share/cb/cbcluster
```

Usage: `cbcluster COMMAND [CMD OPTIONS]`

*Carbon Black Enterprise Server cluster management command line tool*

Available commands are:

```
help    Display this help screen
start   Start the cluster
stop    Stop the cluster
add-node Add a slave node to the cluster
reshard Split single process event store shard into N shards
```

3. From `cb-master` run the following command to initiate cluster configuration:

**Best Practice:** a good best practice is to have two shards per minion node to both distribute the load and allow for equal distribution of shards across all nodes. For best performance the master node should contain no shards. Many factors affect the number of minion nodes but a good rule of thumb is one minion per every 5000 nodes. Factors that influence the number of nodes are sensor activity, hardware configuration, proximity to the master node, and network latency.

1. Sensor activity - A default Windows 7 install with a normal office user will create X processes per day, and each of those processes will contain about 3 kB of event activity after indexing and compression. A typical managed desktop with normal office use will create Y processes per day, and each of those processes will contain about 10 kB of event activity after indexing and compression. These parameters vary from installation to installation and can substantively impact sizing.

2. Hardware - A fully loaded Carbon Black server needs about 8 cores, 64 GB RAM and a disk fast enough to keep up. A steady-state Cb server will have 4-8 threads each with 15-25 IOPS sustained, for a total of about 100-200 IOPS total. Disk writes will briefly burst in both synchronicity (more threads writing) and throughput (same threads writing more) up to 15 threads and a total of 500-1000 IOPS. Any reduction in synchronicity (i.e., an array of four disks with four physical spindles) or throughput will cause processing to slow. Any reduction in CPU or RAM will also cause processing to slow. Any performance reductions will likewise reduce the number of sensors a single server can support, increasing the overall number of servers required.

3. Proximity to Master - If the minion cannot be placed in the same rack on the same high-speed network segment, then a new master and minion will be required for sites with a density greater than 5K nodes. Sites with less than 5K nodes can be handled with a single master model without clustering.

4. Network Latency - The master and minion nodes should be placed on the same network segment with a minimum on a 1GB ethernet connection.

If you will not be utilizing the default /var/cb/data directory to store data on the minion, you will need to add --storage-path=STORAGE\_PATH to specify the new path.

```
[root@cb-master ~]# /usr/share/cb/cbcluster add-node
```

*What is the hostname or IP of the remote node: 172.16.100.110*

*Which shards will be managed by the new cluster node. Leave blank for none.*

*Example: 0-2,3,5,6-10: 0-1*

You'll then be asked for the password of the server to become a minion node. This is the root password for the box in question and not the Carbon Black application password.

4. Once the minion nodes have all been configured, go ahead and start the cluster services:

```
[root@cb-master ~]# /usr/share/cb/cbcluster start
```

5. You should be able to view the results in the config file and in the associated UI on each server.

```
[root@cb-master ~]# cd /etc/cb/
```

```
[root@cb-master ~]# less cluster.conf
```

```
#####  
#  
# /etc/cb/cluster.conf:  
# This file contains Carbon Black Enterprise Server cluster configuration,  
# which includes the list of participating nodes and Solr shards present on  
# every one of those nodes.  
#  
# NOTE: The contents of this file are being managed by  
# /usr/share/cb/cbcluster command line tool and any changes made here may  
# be overwritten next time that tool is used.  
#  
#####
```

```
[Cluster]  
NodeCount=2  
ShardCount=1  
NextSlaveAutoInc=2
```

```
[Master]  
ProcSolrShards=  
Host=172.16.100.110
```

```
[Slave1]
```

ProcSolrShards=0  
Host=172.16.100.111

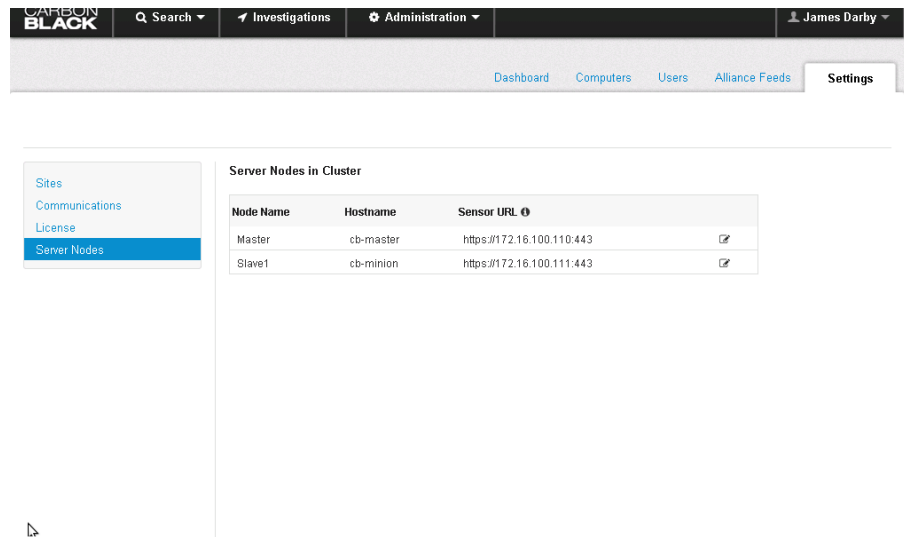


Figure 2 Master Node Post cbcluster add-node

You should now see both servers listed in Administration->Settings->Server Nodes section of the UI.

### Sensor Install and Verification

6. The next step is to add clients to the node by downloading the installer of choice.

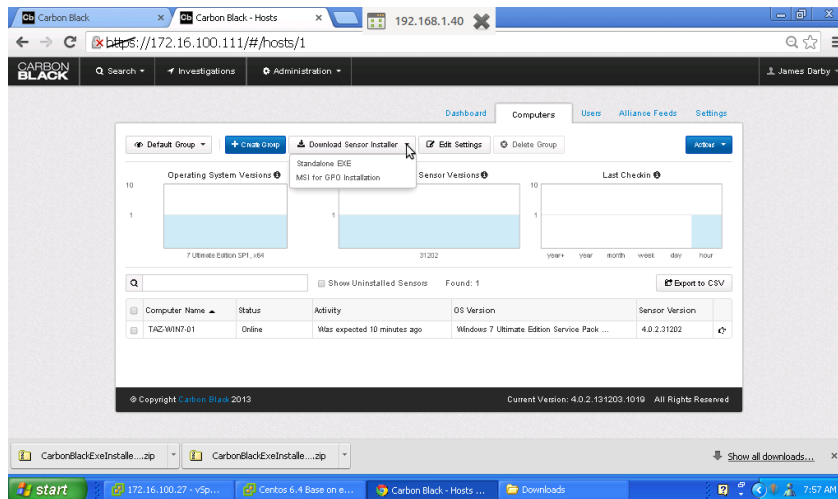
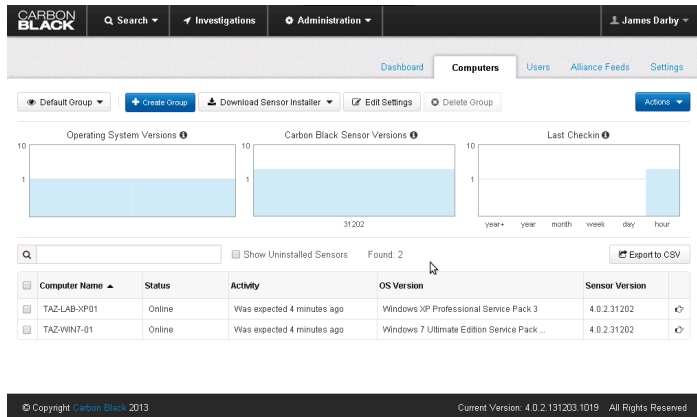


Figure 3 Download Installer

7. After the sensors are installed on the clients verify the clients appear in the UI on the master node.



**Figure 4 Master Post Sensor Install**