SKYBOX™
SECURITY

SKYBOX APPLIANCE HIGH AVAILABILITY INSTALLATION GUIDE

Version 1.5

Skybox Appliance High Availability Setup

OVERVIEW OF THE SKYBOX HIGH AVAILABILITY MODULE

What is the Skybox High Availability Module?

The Skybox High Availability Module is a feature that allows the Skybox application to minimize downtime while reliably keeping any data and tasks intact and operational.

How Does the High Availability Module Work?

The High Availability Module is based on an Active/Passive model. The Secondary server monitors the connection with the Primary server via SOAP requests. As long as the Primary server is up and operational, data will be synchronized from the Primary to the Secondary server. See Figure 1.1

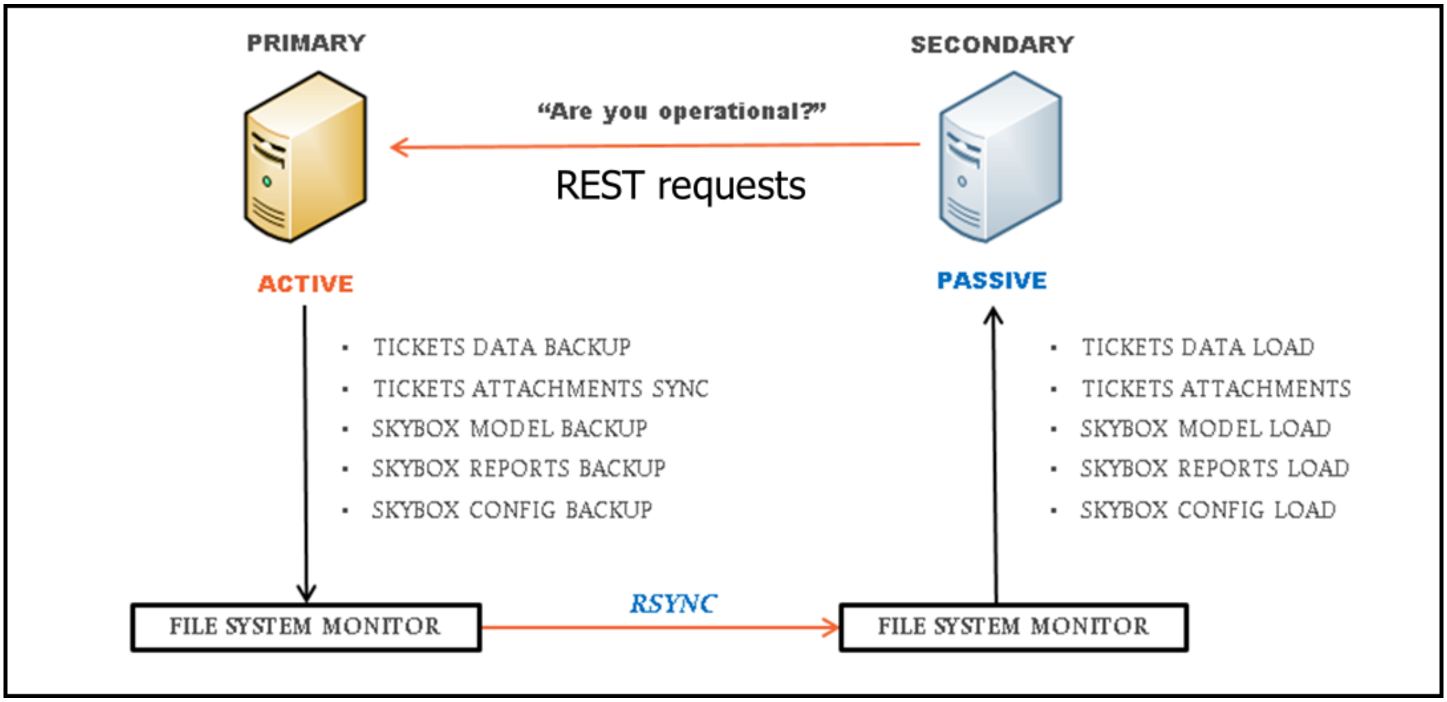


Figure 1.1 - Active/Passive Model

NAT/Port Forwarding

The Skybox High Availability set up includes port forwarding for ease of use and set up for load balancing. The Skybox High Availability Module consists of the following:

- Primary (Active) server enables a firewall that will NAT all traffic incoming to port 443 and forwards it to its internal port 8443 (listening port of Skybox)
- Secondary (Passive) server disables its firewall and performs no NAT translations. Requests on port 443 will be dropped until Secondary detects the Primary Server as unreachable

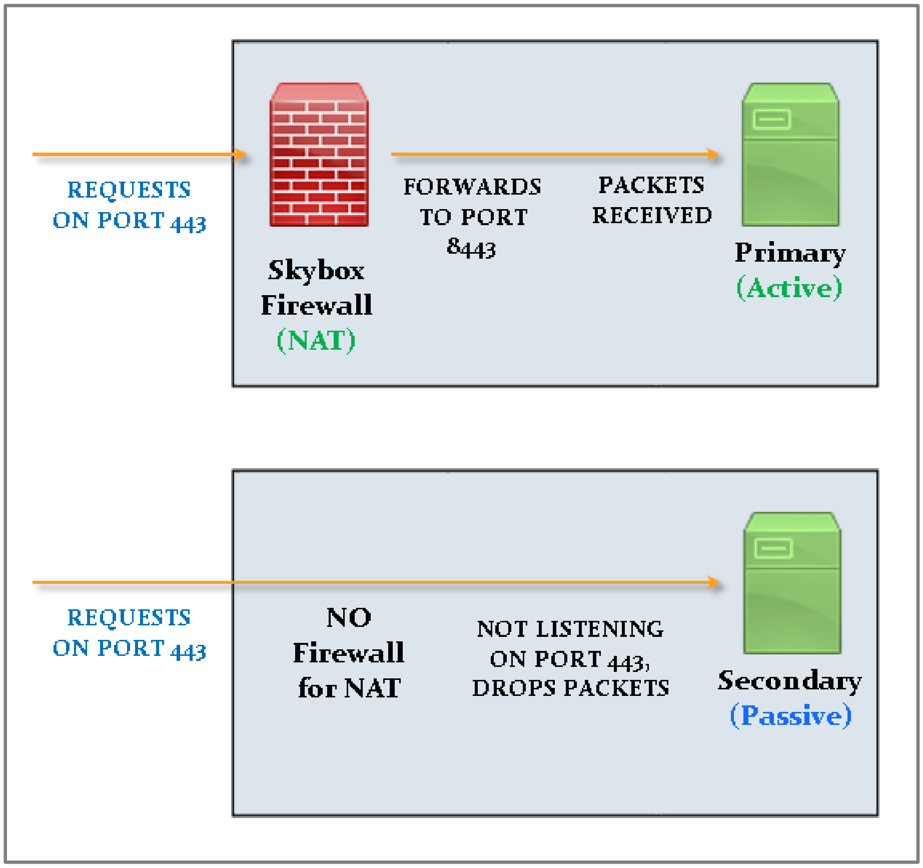


Figure 1.2 - NAT/Port translation setup on Primary and Secondary Servers

Load Balancing with Skybox High Availability

The customer can take advantage of using the NAT/Port forward to enable a load-balancing setup. In order to achieve a load-balancing setup, the customer will need an external load-balancer. For example, on F5 load balancers, the customer can create a common VIP and create a pool with the Skybox servers. Since the High Availability Module uses a NAT rule to port forward all traffic received on port 443 to its internal port 8443, but ONLY on the Active server. A health monitor can be created to check if a server is responding to HTTPS requests on port 443. The recommended health check is to use our native REST API healthcheck call: https://<server_IP>:<port>/skybox/webservice/jaxrsinternal/internal/healthcheck/ping

The load balancer will only forward to traffic to the server that is responding to the healthcheck requests on port 443. If the Primary server is unreachable, the switch over will be activated and the F5 will detect the Secondary server as active and forward traffic to its port 443. See Figure 1.3 below for an example setup.

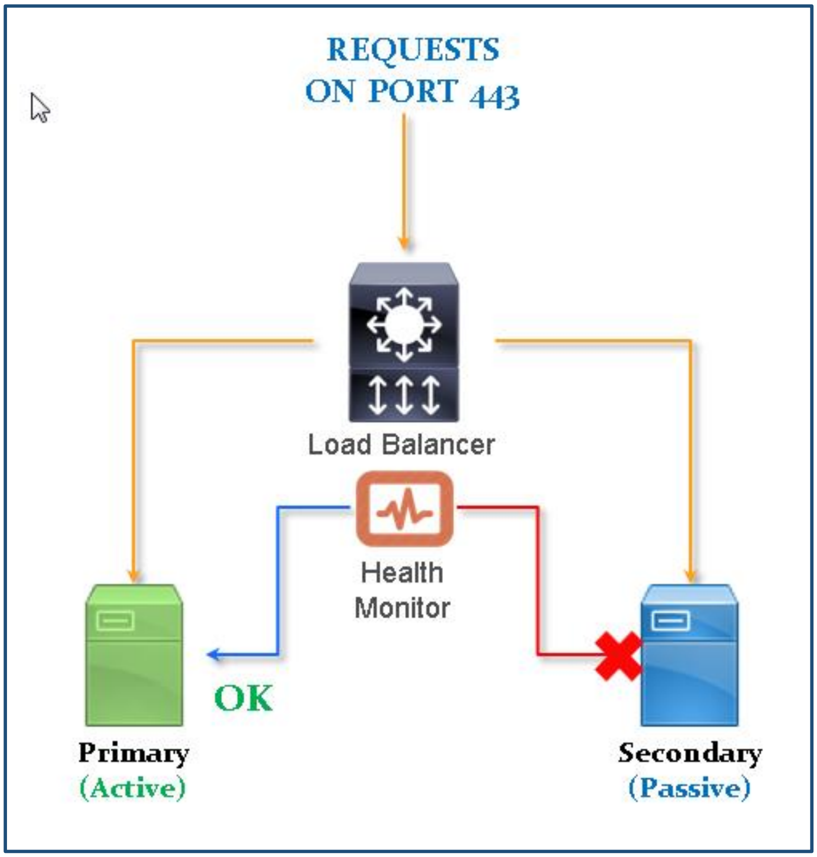


Figure 1.3 - example of a load-balancer setup

Failover Scenario Explained

When the Primary server stops responding according to the timeout thresholds configured, the Secondary server will initiate the failover and become Active (Primary). All tasks will then be operational on the Secondary server. When the Primary server comes back online, the Primary server will detect that the Secondary is active and remain as Passive. This is by design to allow customers/support personnel the time to troubleshoot and diagnose the reason for the Primary server's connection issue. Once the customer/support personnel have completed their troubleshooting, the Secondary server can then be manually set back to Passive and the Primary server back to Active.

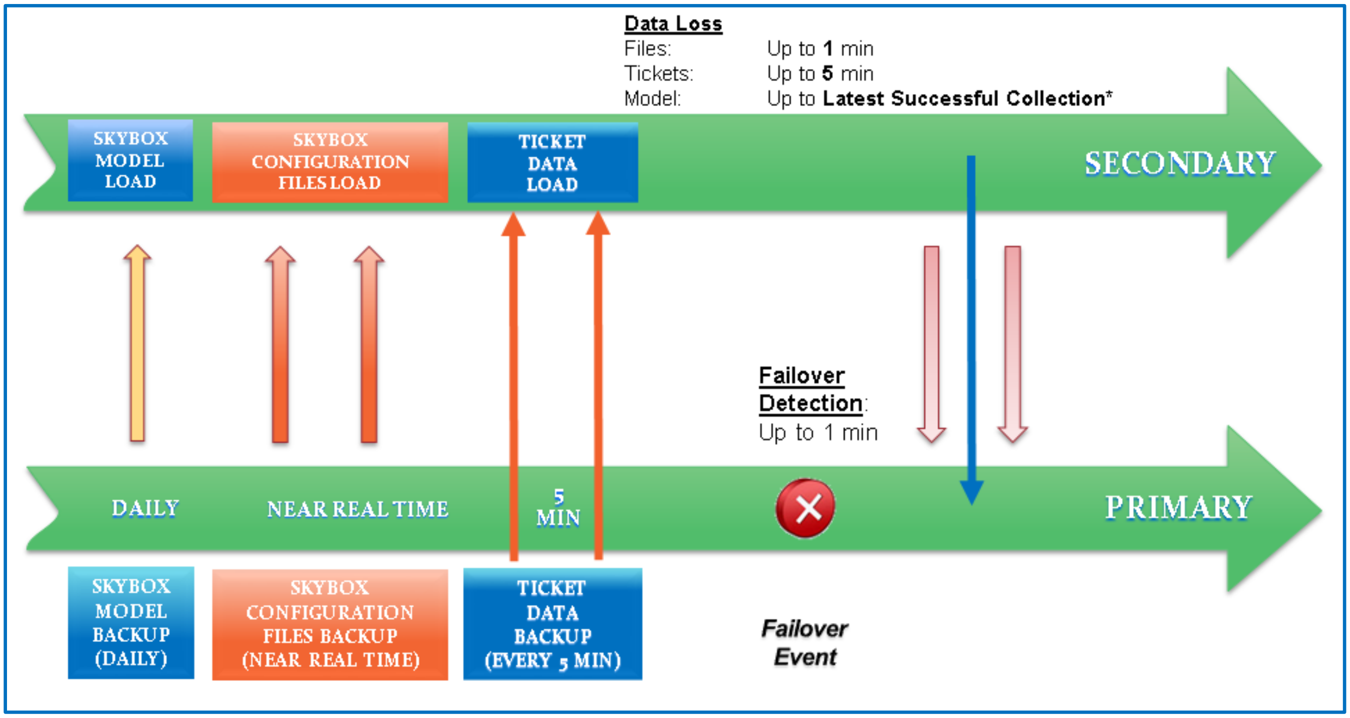
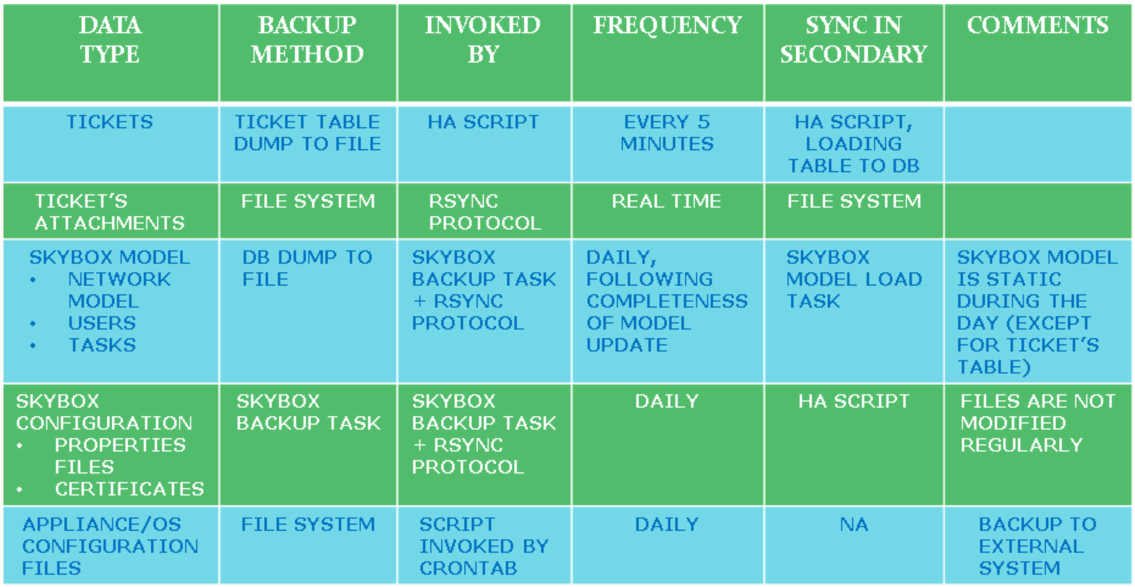

Data Loss
Files: Up to 1 min
Tickets: Up to 5 min
Model: Up to Latest Successful Collection*
SKYBOX MODEL LOAD
SKYBOX CONFIGURATION FILES LOAD
TICKET DATA LOAD
SECONDARY
Failover Detection: Up to 1 min
DAILY
NEAR REAL TIME
5 MIN
PRIMARY
SKYBOX MODEL BACKUP (DAILY)
SKYBOX CONFIGURATION FILES BACKUP (NEAR REAL TIME)
TICKET DATA BACKUP (EVERY 5 MIN)
Failover Event

Figure 1.4 - Failover scenario diagram

What Data is synchronized between the Primary and Secondary Servers?



DATA TYPE	BACKUP METHOD	INVOKED BY	FREQUENCY	SYNC IN SECONDARY	COMMENTS
TICKETS	TICKET TABLE DUMP TO FILE	HA SCRIPT	EVERY 5 MINUTES	HA SCRIPT, LOADING TABLE TO DB	
TICKET'S ATTACHMENTS	FILE SYSTEM	RSYNC PROTOCOL	REAL TIME	FILE SYSTEM	
SKYBOX MODEL • NETWORK MODEL • USERS • TASKS	DB DUMP TO FILE	SKYBOX BACKUP TASK + RSYNC PROTOCOL	DAILY, FOLLOWING COMPLETENESS OF MODEL UPDATE	SKYBOX MODEL LOAD TASK	SKYBOX MODEL IS STATIC DURING THE DAY (EXCEPT FOR TICKET'S TABLE)
SKYBOX CONFIGURATION • PROPERTIES FILES • CERTIFICATES	SKYBOX BACKUP TASK	SKYBOX BACKUP TASK + RSYNC PROTOCOL	DAILY	HA SCRIPT	FILES ARE NOT MODIFIED REGULARLY
APPLIANCE/OS CONFIGURATION FILES	FILE SYSTEM	SCRIPT INVOKED BY CRONTAB	DAILY	NA	BACKUP TO EXTERNAL SYSTEM

Figure 1.5 – table of what information is synchronized

BEFORE INSTALLATION

Before the installation of the Skybox Appliance High Availability package, certain modules and dependencies are required to be installed. ***Note**: The below settings should be applied to both primary and secondary server

Required Dependencies/Setup:

a) Ports 22/tcp and 8443/tcp are opened between appliances in both directions (firewalls)
b) Disable firewalld (if needed, please refer to this section on how to disable the Firewalld service)
c) sudo (called sudo)
d) rsync (called rsync)
e) python suds package (called python-suds)

Using Yum to Verify Installed Package

In order to check if required packages are installed on for dependencies **b-f** in the list above are installed, use the ‘yum info <package name>’ command. The return output should show an “installed” status for return Repo value. If the return value is “base”, then package isn’t installed.

For example, for the python-suds package, would enter the following command in the command shell as the ‘root’user:
yum info python-suds

```
[skyboxview@skybox-pri ~]$ yum info python-suds
Loaded plugins: fastestmirror, presto
Loading mirror speeds from cached hostfile
 * base: mirrors.usinternet.com
 * extras: mirror.team-cymru.org
 * updates: mirror.trouble-free.net
Installed Packages
Name        : python-suds
Arch        : noarch
Version     : 0.4.1
Release     : 5.el7
Size        : 939 k
Repo        : installed
From repo   : anaconda
Summary     : A python SOAP client
URL         : https://fedorahosted.org/suds
License     : LGPLv3+
Description : The suds project is a python soap web services client lib.  Suds leverages
            : python meta programming to provide an intuitive API for consuming web
            : services.  Objectification of types defined in the WSDL is provided
            : without class generation.  Programmers rarely need to read the WSDL since
            : services and WSDL based objects can be easily inspected.
[skyboxview@skybox-pri ~]$
```

Figure 2.1 - yum info command

Using Yum to Install Missing Packages

To install a missing package, use the “yum install -y <package-name>” command. For example, in figure 2.2, this host is missing the python suds package:

```
[root@skybox-sec skyboxview]# yum info python-suds
Loaded plugins: fastestmirror, presto
Loading mirror speeds from cached hostfile
 * base: centos.vwtonline.net
 * extras: mirrors.advancedhosters.com
 * updates: mirror.beyondhosting.net
Installed Packages
Name        : python-suds
Arch        : noarch
Version     : 0.4.1
Release     : 5.el7
Size        : 939 k
Repo        : installed
From repo   : anaconda
Summary     : A python SOAP client
URL         : https://fedorahosted.org/suds
License     : LGPLv3+
Description : The suds project is a python soap web services client lib.  Suds leverages
            : python meta programming to provide an intuitive API for consuming web
            : services.  Objectification of types defined in the WSDL is provided
            : without class generation.  Programmers rarely need to read the WSDL since
            : services and WSDL based objects can be easily inspected.

[root@skybox-sec skyboxview]#
```

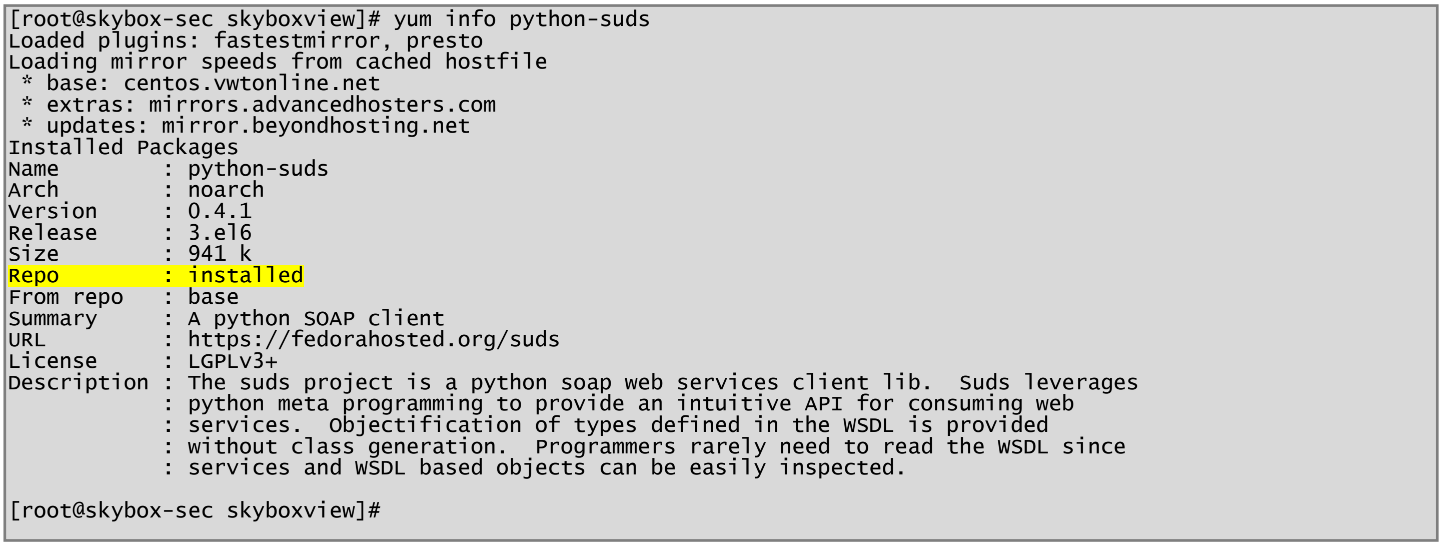
Figure 2.2 – verifying the package is installed using ‘yum info’

As the ‘root’ user, enter the following command on the command line:
yum install -y python-suds

```
[root@skybox-sec skyboxview]# yum install -y python-suds
Loaded plugins: fastestmirror, presto
Setting up Install Process
Loading mirror speeds from cached hostfile
 * base: centos.vwtonline.net
 * extras: mirrors.advancedhosters.com
 * updates: mirror.beyondhosting.net
Resolving Dependencies
--> Running transaction check
---> Package python-suds.noarch 0:0.4.1-3.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

====================================================================================================================================
 Package                           Arch                         Version                            Repository                  Size
====================================================================================================================================
Installing:
 python-suds                       noarch                       0.4.1-3.el6                        base                       218 k

Transaction Summary
====================================================================================================================================
Install       1 Package(s)

Total download size: 218 k
Installed size: 941 k
Downloading Packages:
Setting up and reading Presto delta metadata
Processing delta metadata
Package(s) data still to download: 218 k
python-suds-0.4.1-3.el6.noarch.rpm                                                                            | 218 kB     00:00
warning: rpmts_HdrFromFdno: Header V3 RSA/SHA1 Signature, key ID c105b9de: NOKEY
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
Importing GPG key 0xC105B9DE:
 Userid : CentOS-6 Key (CentOS 6 Official Signing Key) <centos-6-key@centos.org>
 Package: centos-release-6-7.el6.centos.12.3.x86_64 (@anaconda-CentOS-201112091719.x86_64/6.3)
 From   : /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : python-suds-0.4.1-3.el6.noarch                                                                                   1/1
  Verifying  : python-suds-0.4.1-3.el6.noarch                                                                                   1/1

Installed:
  python-suds.noarch 0:0.4.1-3.el6

Complete!
[root@skybox-sec skyboxview]#
```

Figure 2.3 – installing a package using yum

Use the “yum info <package-name>” command to verify the package was installed properly.See Figure 2.4 below:



```
[root@skybox-sec skyboxview]# yum info python-suds
Loaded plugins: fastestmirror, presto
Loading mirror speeds from cached hostfile
 * base: centos.vwtonline.net
 * extras: mirrors.advancedhosters.com
 * updates: mirror.beyondhosting.net
Installed Packages
Name        : python-suds
Arch        : noarch
Version     : 0.4.1
Release     : 3.el6
Size        : 941 k
Repo        : installed
From repo   : base
Summary     : A python SOAP client
URL         : https://fedorahosted.org/suds
License     : LGPLv3+
Description : The suds project is a python soap web services client lib.  Suds leverages
            : python meta programming to provide an intuitive API for consuming web
            : services.  Objectification of types defined in the WSDL is provided
            : without class generation.  Programmers rarely need to read the WSDL since
            : services and WSDL based objects can be easily inspected.

[root@skybox-sec skyboxview]#
```

Figure 2.4 – verifying the package is installed using ‘yum info’

Configuring sudo for ‘skyboxview’

The account used to log into the High Availability will be ‘skyboxview’. In order to have the correct permissions, the account needs to run the High Availability modue, the ‘skyboxview’ user will be defined as a sudoer.

As the ‘root’ user, enter the following commands on the command line:
a) Enter the command ‘visudo’ to edit the *sudoer* file. Use ONLY the ‘visudo’ command as it protects the correct format for the file:

```
[root@skybox-pri skyboxview]# visudo
```

b) Uncomment the following line by removing the ‘#’ in front of it. Move the cursor in front of the ‘#’ mark and press the ‘x’ key to delete the sing character:
Before:

```
## Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL
```

After:

```
## Same thing without a password
 %wheel ALL=(ALL)       NOPASSWD: ALL
```

c) Save and quit by entering the following command. **(*Note – this is lowercase, don’t use upper!)**:
ESC : x   (ESC refers to the Escape key)

d) Add the “skyboxview” user to the “wheel” with the following command on the command line:
usermod -aG wheel skyboxview

```
[root@skybox-pri skyboxview]# usermod -aG wheel skyboxview
```

e) Verify that the “skyboxview” user is part of the wheel group with the following command on the command line. The ‘wheel’ group should be listed. See the highlight example below:
groups skyboxview

```
[root@skybox-pri skyboxview]# groups skyboxview
skyboxview : skyboxview wheel
[root@skybox-pri skyboxview]#
```

Setting up SSH Trust Between Servers

The Primary and Secondary servers require the ability to login to each other without the prompting of a password. To facilitate this need, the servers will use Trusted SSH keys between the servers in order to connect to each other without prompting for a password every time one servers connects to the other.

a) Login as the skyboxview user. Use the following command on the command line:
su skyboxview

```
[root@skybox-pri skyboxview]# su skyboxview
[skyboxview@skybox-pri ~]$
```

b) First, create the ssh-keys by entering the “ssh-keygen -t dsa” command, leave the passphrase empty and use the default settings. This will create the public and private SSH keys used for the connection. See figure 3.1:
ssh-keygen -t dsa

```
[root@skybox-pri skyboxview]# su skyboxview
[skyboxview@skybox-pri ~]$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/skyboxview/.ssh/id_dsa):
Created directory '/home/skyboxview/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/skyboxview/.ssh/id_dsa.
Your public key has been saved in /home/skyboxview/.ssh/id_dsa.pub.
The key fingerprint is:
46:ef:13:43:da:3e:cc:4c:1a:48:92:75:53:59:11:b7 skyboxview@skybox-pri
The key's randomart image is:
+--[ DSA 1024]----+
|      . o..o+o.  |
|     o . ..  . . |
|    o . . .   E  |
|     o o =       |
|      . S *      |
|       . X o     |
|        . O      |
|           o     |
|                 |
+-----------------+
[skyboxview@skybox-pri ~]$
```

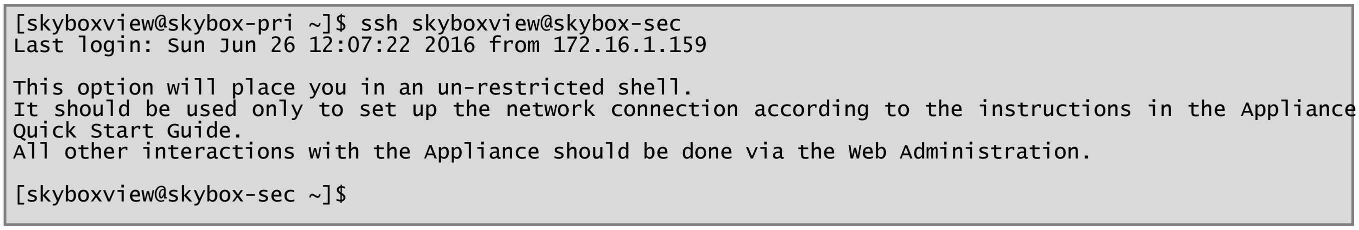
Figure 3.1 – creating SSH keys

c) Next, copy the public key to the destination server that the host will connect to by entering the ‘ssh-copy-id -i ~/.ssh/id_dsa.pub skyboxview@<other IP or hostname>’ command. Enter ‘yes’ for the RSA key fingerprint, if prompted. Enter the ‘skyboxview’password when prompted. For example, Figure 3.2 displays the output from issuing the command to connect to skybox-sec :
ssh-copy-id -i ~/.ssh/id_dsa.pub skyboxview@skybox-sec

```
[skyboxview@skybox-pri ~]$ ssh-copy-id -i ~/.ssh/id_dsa.pub skyboxview@skybox-sec
The authenticity of host 'skybox-sec (172.16.1.214)' can't be established.
RSA key fingerprint is 8e:3f:30:f4:b4:5a:69:46:f4:01:cd:9b:49:fa:e5:76.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'skybox-sec,172.16.1.214' (RSA) to the list of known hosts.
skyboxview@skybox-sec's password:
Now try logging into the machine, with "ssh 'skyboxview@skybox-sec'", and check in:

  .ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.

[skyboxview@skybox-pri ~]$
```

Figure 3.2 – copying SSH key to remote server

d) Verify that the ssh-keys were installed correctly by connecting via ssh to the destination server. You should NOT be prompted for a password if the ‘ssh-copy-id’ command was successfully installed:

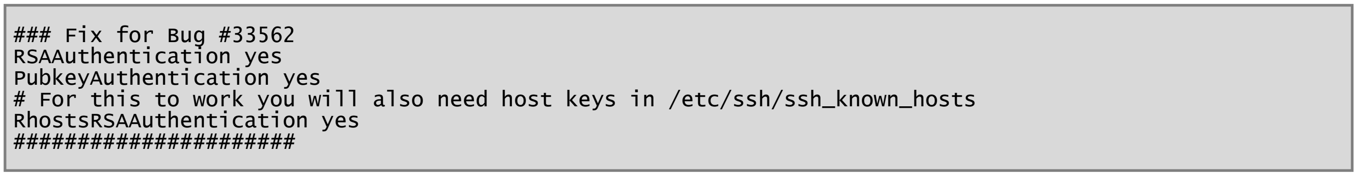


```
[skyboxview@skybox-pri ~]$ ssh skyboxview@skybox-sec
Last login: Sun Jun 26 12:07:22 2016 from 172.16.1.159

This option will place you in an un-restricted shell.
It should be used only to set up the network connection according to the instructions in the Appliance
Quick Start Guide.
All other interactions with the Appliance should be done via the Web Administration.

[skyboxview@skybox-sec ~]$
```

Figure 3.3 – verifying connection to remote server

e) Finally, the SSH server needs to configured to address an issue with Bug#33562. ***NOTE:** In the newer versions of Skybox, the fix may already be applied to the server. The fix starts on line 44.

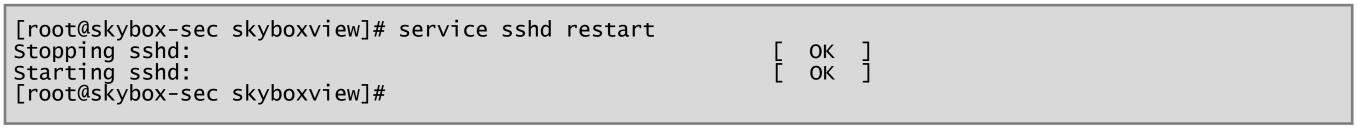
As the ‘root’ user, edit the ‘/etc/ssh/sshd_config’ file using vi. Add the following lines if they don’t exist to the bottom of the file. Enter ‘:x” to save. See figure 3.4:
### Fix for Bug #33562
RSAAuthentication yes
PubkeyAuthentication yes
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
RhostsRSAAuthentication yes
#######################



```
### Fix for Bug #33562
RSAAuthentication yes
PubkeyAuthentication yes
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
RhostsRSAAuthentication yes
######################
```

Figure 3.4 – adding fix for Bug # 33562

f) As the ‘root’ user, restart the SSH service by entering the following :
service sshd restart



```
[root@skybox-sec skyboxview]# service sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
[root@skybox-sec skyboxview]#
```

Figure 3.5 – restart the SSHD service

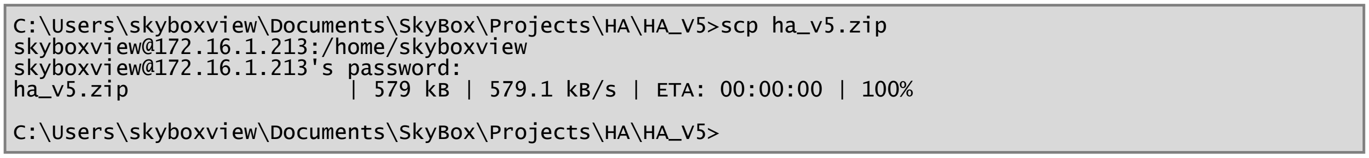
g) Perform the same setup on the remote server.

INSTALLING THE SKYBOX HA MODULE

The Skybox High Availability module will be installed on both the Primary and Secondary servers in the ‘skyboxview’ home directory: /home/skyboxview/

Extracting the Module Files

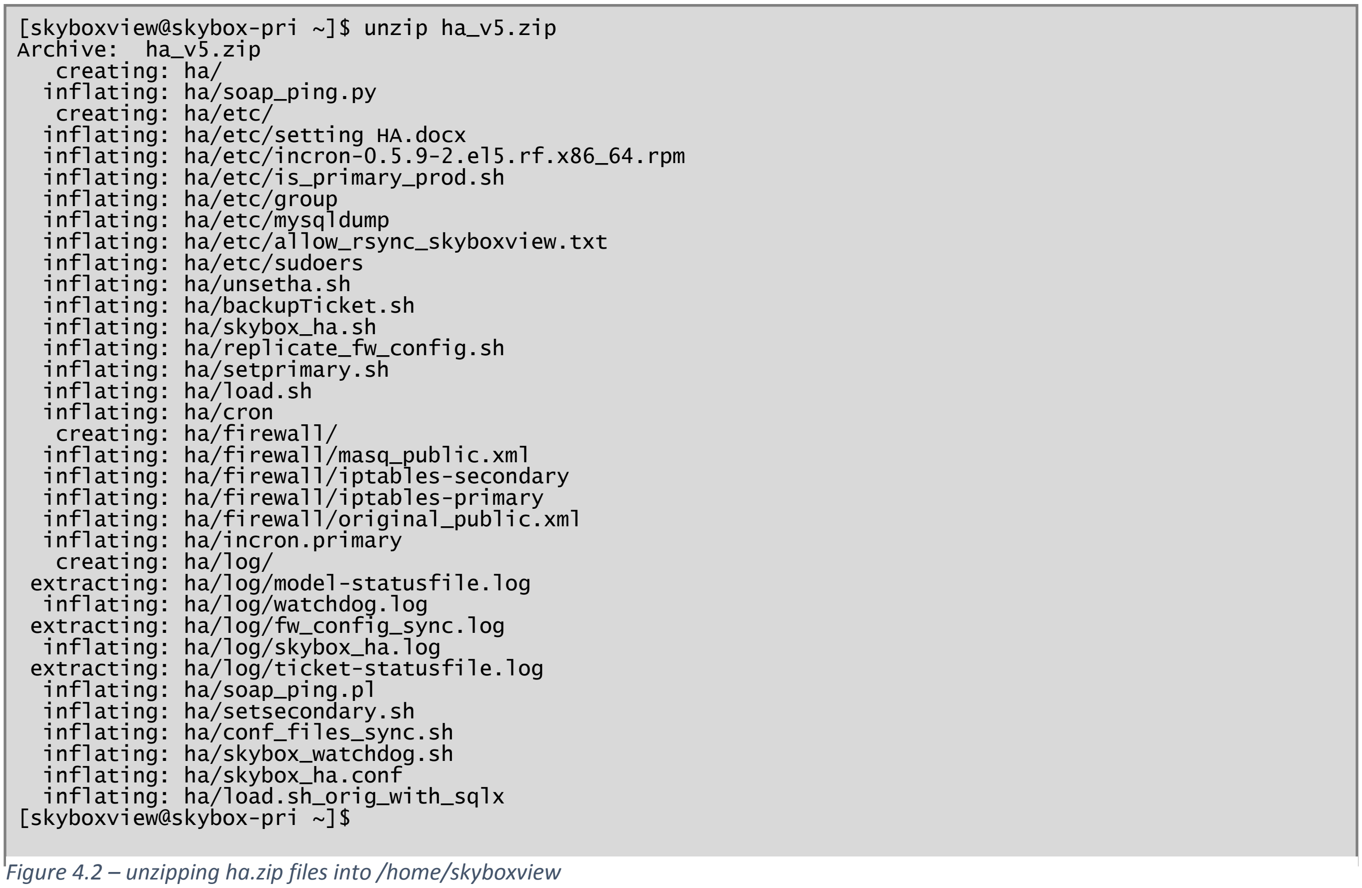
a) Copy over the ha_v*.zip file to the /home/skyboxview directory as the ‘skyboxview’ user (otherwise the file permissions will use ‘root’ instead of the ‘skyboxview’ user. Use whatever method you are comfortable with. For example, using scp:



```
C:\Users\skyboxview\Documents\SkyBox\Projects\HA\HA_V5>scp ha_v5.zip
skyboxview@172.16.1.213:/home/skyboxview
skyboxview@172.16.1.213's password:
ha_v5.zip                  | 579 kB | 579.1 kB/s | ETA: 00:00:00 | 100%

C:\Users\skyboxview\Documents\SkyBox\Projects\HA\HA_V5>
```

Figure 4.1 – copy over ha_v.zip file*

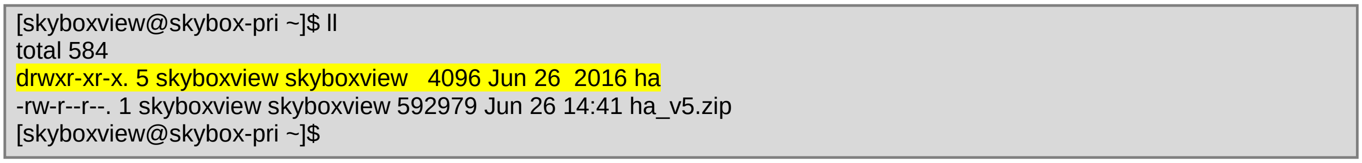
b) Unzip the ha_v*.zip file by using the following command in the command line:
unzip <filename>
Example:



```
[skyboxview@skybox-pri ~]$ unzip ha_v5.zip
Archive:  ha_v5.zip
   creating: ha/
  inflating: ha/soap_ping.py
   creating: ha/etc/
  inflating: ha/etc/setting HA.docx
  inflating: ha/etc/incron-0.5.9-2.el5.rf.x86_64.rpm
  inflating: ha/etc/is_primary_prod.sh
  inflating: ha/etc/group
  inflating: ha/etc/mysqldump
  inflating: ha/etc/allow_rsync_skyboxview.txt
  inflating: ha/etc/sudoers
  inflating: ha/unsetha.sh
  inflating: ha/backupTicket.sh
  inflating: ha/skybox_ha.sh
  inflating: ha/replicate_fw_config.sh
  inflating: ha/setprimary.sh
  inflating: ha/load.sh
  inflating: ha/cron
   creating: ha/firewall/
  inflating: ha/firewall/masq_public.xml
  inflating: ha/firewall/iptables-secondary
  inflating: ha/firewall/iptables-primary
  inflating: ha/firewall/original_public.xml
  inflating: ha/incron.primary
   creating: ha/log/
 extracting: ha/log/model-statusfile.log
  inflating: ha/log/watchdog.log
 extracting: ha/log/fw_config_sync.log
  inflating: ha/log/skybox_ha.log
 extracting: ha/log/ticket-statusfile.log
  inflating: ha/soap_ping.pl
  inflating: ha/setsecondary.sh
  inflating: ha/conf_files_sync.sh
  inflating: ha/skybox_watchdog.sh
  inflating: ha/skybox_ha.conf
  inflating: ha/load.sh_orig_with_sqlx
[skyboxview@skybox-pri ~]$
```

Figure 4.2 – unzipping ha.zip files into /home/skyboxview

c) Verify the file was unzipped by issuing the command ‘ll’ on the command line:
ll



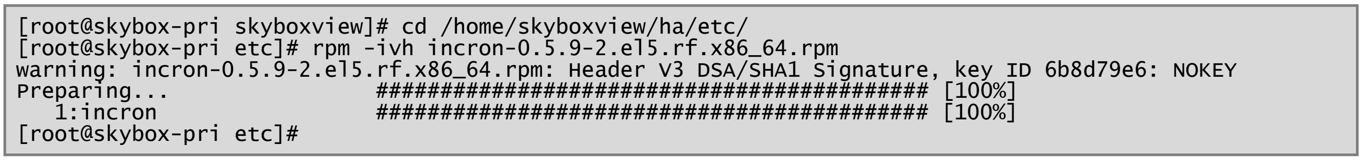
```
[skyboxview@skybox-pri ~]$ ll
total 584
drwxr-xr-x. 5 skyboxview skyboxview   4096 Jun 26  2016 ha
-rw-r--r--. 1 skyboxview skyboxview 592979 Jun 26 14:41 ha_v5.zip
[skyboxview@skybox-pri ~]$
```

Figure 4.3 – verify ha file was unzipped and directory was created

Installing and Configuring INCRON

The INCRON service monitors the directories that will be rsync’d from the Primary server to the Secondary server. The following section are the instructions to install the INCRON package

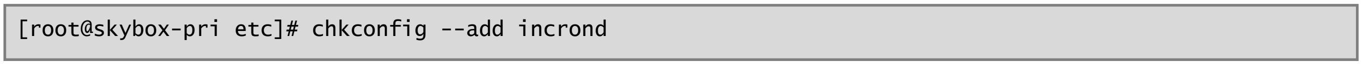
a) As the ‘root’ user, cd to /home/skyboxview/ha/etc, and run the following command to install *incron*:
rpm -ivh incron-0.5.9-2.el5.rf.x86_64.rpm



```
[root@skybox-pri skyboxview]# cd /home/skyboxview/ha/etc/
[root@skybox-pri etc]# rpm -ivh incron-0.5.9-2.el5.rf.x86_64.rpm
warning: incron-0.5.9-2.el5.rf.x86_64.rpm: Header V3 DSA/SHA1 Signature, key ID 6b8d79e6: NOKEY
Preparing...                ########################################### [100%]
   1:incron                 ########################################### [100%]
[root@skybox-pri etc]#
```

Figure 5.1 - installing the INCRON package

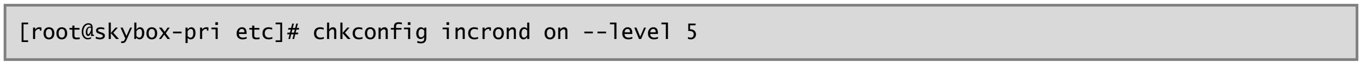
b) INCRON needs to set up to run after bootup. Add INCRON to start at bootup with the following command:
chkconfig --add incrond



```
[root@skybox-pri etc]# chkconfig --add incrond
```

Figure 5.2 - adding INCROND to startup

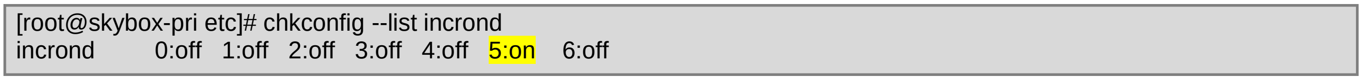
c) Change the run level for INCROND with the following command:
chkconfig incrond on --level 5



```
[root@skybox-pri etc]# chkconfig incrond on --level 5
```

Figure 5.3 - changing the run level for INCROND

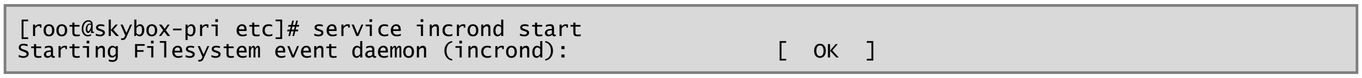
d) Verify that INCRON was added to the startup with the following command. Run level 5 should be listed as ‘on’:
chkconfig --list incrond



```
[root@skybox-pri etc]# chkconfig --list incrond
incrond         0:off   1:off   2:off   3:off   4:off   5:on    6:off
```

Figure 5.4 - verifying the run level for INCROND

e) Start the INCROND service with the following command:
service incrond start



```
[root@skybox-pri etc]# service incrond start
Starting Filesystem event daemon (incrond):                [  OK  ]
```

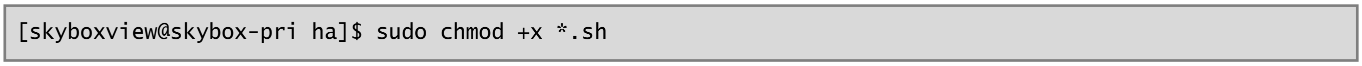
Figure 5.5 - starting the INCROND service

Server Configuration Files and Scripts

The Skybox High Availability module uses a configuration file to pull the necessary information from. The file is the skybox_ha.conf file located in ‘/home/skyboxview/ha/’: (full path is ‘/home/skyboxview/ha/skybox_ha.conf’)

The shell scripts located in the HA directory will need to have the execute bit set by the ‘skyboxview’ user to ensure that the scripts will run properly. Follow the guide below:

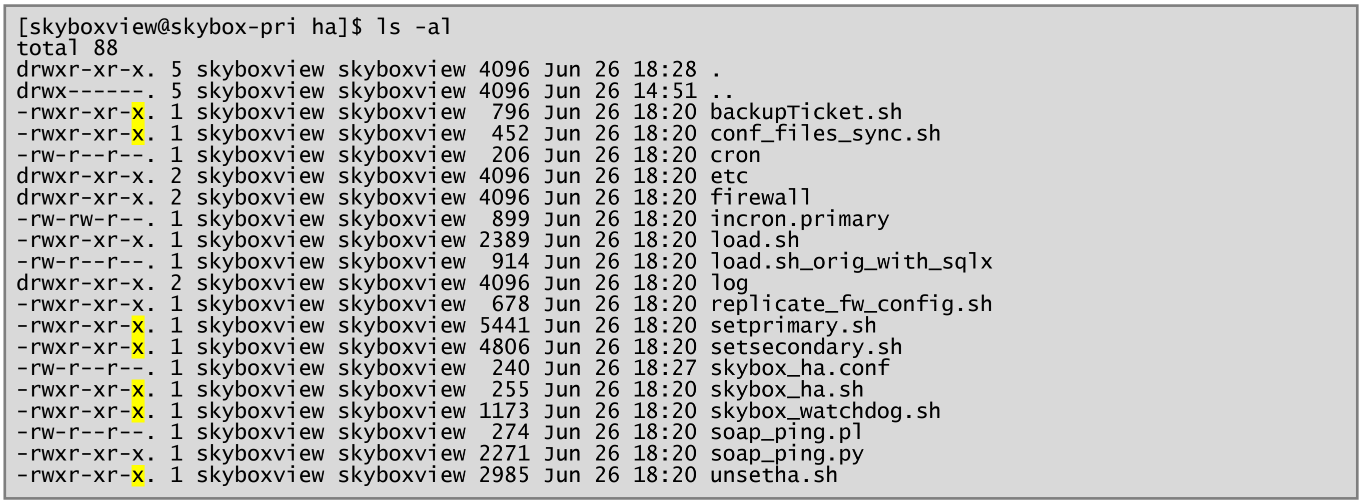
a) As the ‘skyboxview’ user, enable the execute bit for all shell scripts by running the following command:
sudo chmod +x *.sh



```
[skyboxview@skybox-pri ha]$ sudo chmod +x *.sh
```

Figure 6.1 - setting the execute bit for all scripts in HA directory

b) Verify that all the shell scripts have the execute bit set by using the following command:
ls -al

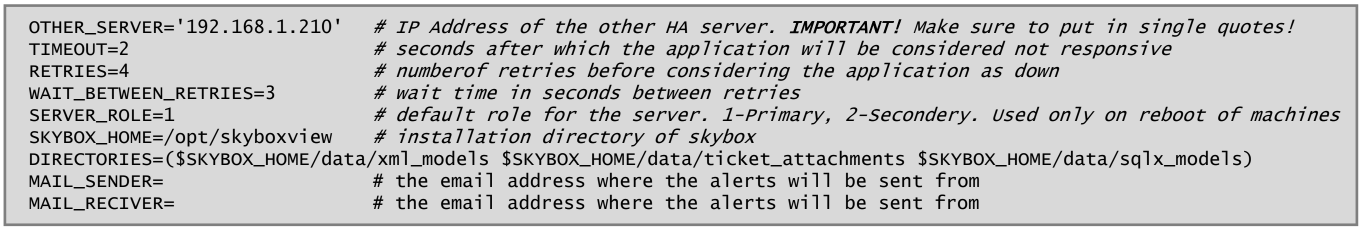


```
[skyboxview@skybox-pri ha]$ ls -al
total 88
drwxr-xr-x. 5 skyboxview skyboxview 4096 Jun 26 18:28 .
drwx------. 5 skyboxview skyboxview 4096 Jun 26 14:51 ..
-rwxr-xr-x. 1 skyboxview skyboxview  796 Jun 26 18:20 backupTicket.sh
-rwxr-xr-x. 1 skyboxview skyboxview  452 Jun 26 18:20 conf_files_sync.sh
-rw-r--r--. 1 skyboxview skyboxview  206 Jun 26 18:20 cron
drwxr-xr-x. 2 skyboxview skyboxview 4096 Jun 26 18:20 etc
drwxr-xr-x. 2 skyboxview skyboxview 4096 Jun 26 18:20 firewall
-rw-rw-r--. 1 skyboxview skyboxview  899 Jun 26 18:20 incron.primary
-rwxr-xr-x. 1 skyboxview skyboxview 2389 Jun 26 18:20 load.sh
-rw-r--r--. 1 skyboxview skyboxview  914 Jun 26 18:20 load.sh_orig_with_sqlx
drwxr-xr-x. 2 skyboxview skyboxview 4096 Jun 26 18:20 log
-rwxr-xr-x. 1 skyboxview skyboxview  678 Jun 26 18:20 replicate_fw_config.sh
-rwxr-xr-x. 1 skyboxview skyboxview 5441 Jun 26 18:20 setprimary.sh
-rwxr-xr-x. 1 skyboxview skyboxview 4806 Jun 26 18:20 setsecondary.sh
-rw-r--r--. 1 skyboxview skyboxview  240 Jun 26 18:27 skybox_ha.conf
-rwxr-xr-x. 1 skyboxview skyboxview  255 Jun 26 18:20 skybox_ha.sh
-rwxr-xr-x. 1 skyboxview skyboxview 1173 Jun 26 18:20 skybox_watchdog.sh
-rw-r--r--. 1 skyboxview skyboxview  274 Jun 26 18:20 soap_ping.pl
-rwxr-xr-x. 1 skyboxview skyboxview 2271 Jun 26 18:20 soap_ping.py
-rwxr-xr-x. 1 skyboxview skyboxview 2985 Jun 26 18:20 unsetha.sh
```

Figure 6.2 - verifying that the execute bits are set for shell scripts

c) Edit the skybox_ha.conf file located in the /home/skyboxview/ha directory using a text editor like vi. ***NOTE:** The DIRECTORIES variable should not be changed! The variable are for directories that are to be synced.
Also, the OTHER_SERVER variable MUST have single quotes, otherwise the shell script will error out due to linux interpreting the string as numerical value, not a string value!

Example Configuration



```
OTHER_SERVER='192.168.1.210'   # IP Address of the other HA server. IMPORTANT! Make sure to put in single quotes!
TIMEOUT=2                      # seconds after which the application will be considered not responsive
RETRIES=4                      # numberof retries before considering the application as down
WAIT_BETWEEN_RETRIES=3         # wait time in seconds between retries
SERVER_ROLE=1                  # default role for the server. 1-Primary, 2-Secondery. Used only on reboot of machines
SKYBOX_HOME=/opt/skyboxview    # installation directory of skybox
DIRECTORIES=($SKYBOX_HOME/data/xml_models $SKYBOX_HOME/data/ticket_attachments $SKYBOX_HOME/data/sqlx_models)
MAIL_SENDER=                   # the email address where the alerts will be sent from
MAIL_RECIVER=                  # the email address where the alerts will be sent from
```

Figure 6.3 - example configuration for a primary server

d) Before continuing to next steps, make sure BOTH Primary and Secondary servers have the skybox_ha.conf configured and the shell scripts have the execute bit set!

Executing the setprimary and setsecondary Scripts

Now it’s time to complete the set up for the Primary and Secondary server. ***IMPORTANT: These scripts must be ran as the ‘skyboxview’ user!!!***

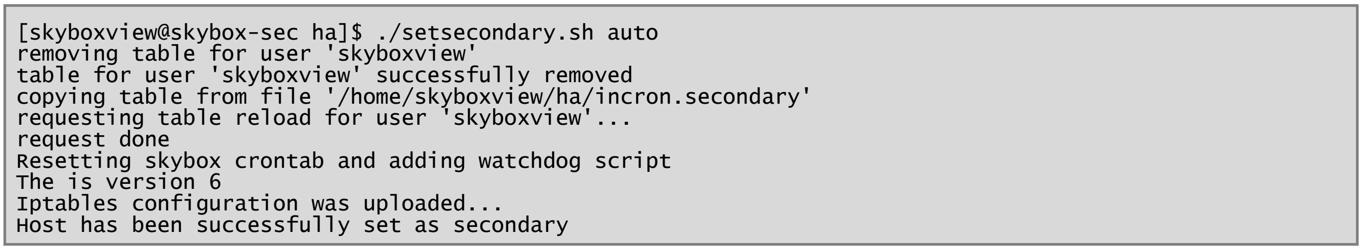
Setting up the Secondary Server First

The Secondary server must be set up first, as the Primary server will perform a ‘role’ check when its script is ran!

The script to set up the Secondary server is the ‘setsecondary.sh’ script located in the ‘/home/skyboxview/ha/’ directory. The script will do the following:

- Create the cronjobs for the ‘skyboxview’ user, which includes the skybox_watchdog.sh and the conf_files_sync.sh scripts
- Create the incronjobs for the ‘skyboxview’ user for INCROND to monitor and automatically load any model files received
- Disable any port forwarding or firewalls
- Set the sender/receiver email addresses for the alerts
- Set the task scheduling in /opt/skyboxview/server/conf/sb_server.properties to ‘false’

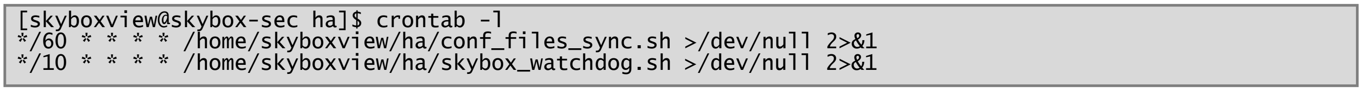
a) Log in as the ‘skyboxview’ user, cd to the /home/skyboxview/ha directory, and run the set primary script. ***NOTE: You must specify ‘auto’, otherwise the script will not configure the server properly!***
./setsecondary auto



```
[skyboxview@skybox-sec ha]$ ./setsecondary.sh auto
removing table for user 'skyboxview'
table for user 'skyboxview' successfully removed
copying table from file '/home/skyboxview/ha/incron.secondary'
requesting table reload for user 'skyboxview'...
request done
Resetting skybox crontab and adding watchdog script
The is version 6
Iptables configuration was uploaded...
Host has been successfully set as secondary
```

Figure 7.1 - running the setsecondary.sh script

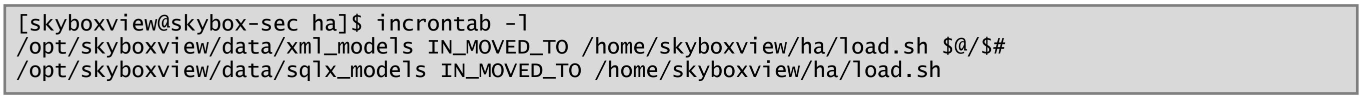
b) Verify that the cron jobs were created successfully by using the following command. The output should match up with the figure 7.2:
crontab -l



```
[skyboxview@skybox-sec ha]$ crontab -l
*/60 * * * * /home/skyboxview/ha/conf_files_sync.sh >/dev/null 2>&1
*/10 * * * * /home/skyboxview/ha/skybox_watchdog.sh >/dev/null 2>&1
```

Figure 7.2 - verifying the cron jobs were created

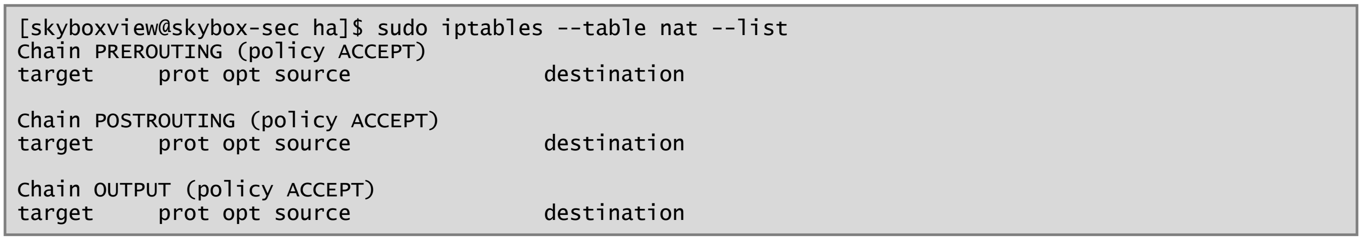
c) Verify that the incron jobs were created successfully by using the following. The output should match up with the figure 7.3:
Incrontab -l



```
[skyboxview@skybox-sec ha]$ incrontab -l
/opt/skyboxview/data/xml_models IN_MOVED_TO /home/skyboxview/ha/load.sh $@/$#
/opt/skyboxview/data/sqlx_models IN_MOVED_TO /home/skyboxview/ha/load.sh
```

Figure 7.3 - verifying the incron jobs were created

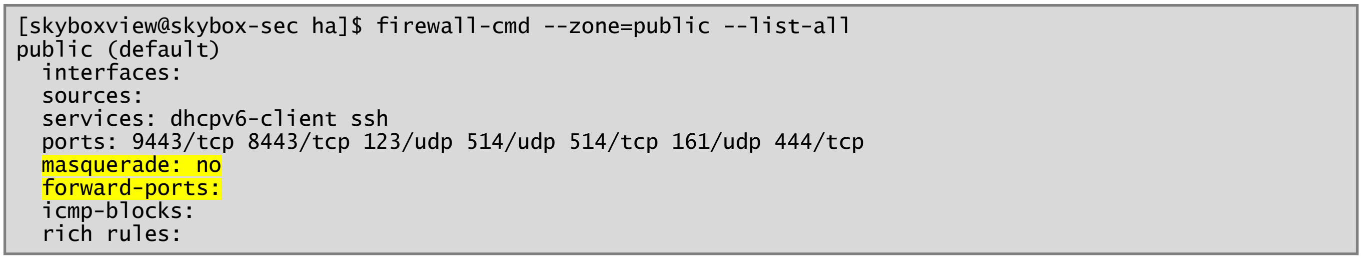
d) Verify that the firewall is disabled and port forwarding isn't enabled. Depending on the version on CentOS, the commands will vary.
For CentOS 6:
`sudo iptables --table nat --list`



```
[skyboxview@skybox-sec ha]$ sudo iptables --table nat --list
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Figure 7.4 – verifying firewall configuration with CentOS 6

For CentOS 7:
firewall-cmd –zone=public –list-all



```
[skyboxview@skybox-sec ha]$ firewall-cmd --zone=public --list-all
public (default)
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports: 9443/tcp 8443/tcp 123/udp 514/udp 514/tcp 161/udp 444/tcp
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

Figure 7.5 - verifying firewall configuration for CentOS 7

Configuring the skybox_ha.conf File

Before starting up the Primary and Secondary server, you must configure the skybox_ha.conf file. This file is used to configure the different options for the HA setup.

Review the following skybox_ha.conf file. Lines in RED will indicate the function and options for each flag:

```
# Configure the following for server role: 1 = Primary; 2 = Secondary
SERVER_ROLE=1

# Configure the following for REST ping configuration to counterpart server
OTHER_SERVER='insert_ip_address_here'              // enter the IP address of the counterpart server
TIMEOUT=10                                         // enter in seconds the timeout value
RETRIES=6                                          // enter the amount of retries before failure consideration
WAIT_BETWEEN_RETRIES=3                             // enter in seconds the amount between each ping attempt

# Configure REST ping intervals
REST_PING_INTERVAL=10                              // configure the interval between each ping check

# Configure the following value of where the Skybox installation is located - remember this is case sensitive
SKYBOX_HOME=/opt/skyboxview                        // configure the location of the Skybox installation

# Configure the following value of where the HA directory is located - remember this is case sensitive
SKYUSERHOME=/home/skyboxview/ha                    // configure the location of the HA installation

# Configure the following log locations
WATCHDOGLOG=$SKYUSERHOME/log/watchdog.log          // change for custom location for the watchdog logs
SKYBOXLOG=$SKYUSERHOME/log/skybox_ha.log           // change for custom location for the HA logs
LOADSTATUS=$SKYUSERHOME/log/ticket-statusfile.log  // change for custom location for the ticket status logs
MODELSTATUS=$SKYUSERHOME/log/model-statusfile.log  // change for custom location for the model upload logs
FWCONFSYNCLOG=$SKYUSERHOME/log/fw_config_sync.log  // change for custom location for the firewall config sync logs


# Directory locations – This should not be changed
DIRECTORIES=($SKYBOX_HOME/data/xml_models $SKYBOX_HOME/data/ticket_attachments
$SKYBOX_HOME/data/sqlx_models)

# Configure the following below to enable the mail settings
MAIL_SENDER=                                       // Enter the Sender e-mail address
MAIL_RECEIVER=                                     // Enter the Receiver e-mail address

# Enable this flag to disable the GUI port on the secondary (only configure on the secondary) - Values: true/false
DISABLE_SECONDARY_GUI=false

# Configure the following if you are using a proxy between the Primary and Secondary - Values: true/false
PROXYVAR=false                                     // enter true to enable
PROXYUSER=                                         // enter the username used to authenticate access
PROXYPASSWORD=                                     // enter the password for the username
PROXYHOST=                                         // enter the IP address of the proxy
PROXYPORT=                                         // enter the port number for the proxy
```

Setting up the Primary Server

The script to set up the Primary server is the ‘setprimary.sh’ script located in the ‘/home/skyboxview/ha/’ directory. The script will do the following:

- Create the cronjobs for the ‘skyboxview’ user, which includes the skybox_watchdog.sh, backupTicket.sh, and replicate_fw_config.sh scripts
- Create the incronjobs for the ‘skyboxview’ user to rsync the file over the Secondary server
- Create the port forwarding of port 443/tcp to forward to port 8443/tcp
- Set the sender/receiver email addresses for the alerts
- Set the task scheduling in /opt/skyboxview/server/conf/sb_server.properties to ‘true’

a) Log in as the ‘skyboxview’ user, cd to the /home/skyboxview/ha directory, and run the set primary script. ***NOTE: You must specify ‘auto’, otherwise the script will not configure the server properly!**
./setprimary auto

```
[skyboxview@skybox-pri ha]$ ./setprimary.sh auto
removing table for user 'skyboxview'
table for user 'skyboxview' does not exist
copying table from file 'incron.primary'
requesting table reload for user 'skyboxview'...
request done
Resetting skybox crontab and adding watchdog script
no crontab for skyboxview
The is version 6
Iptables configuration was uploaded...
No mail parameters supplied
Enabling Task Scheduling...
Host has been successfully set as primary
```

Figure 8.1 - running the setprimary script

b) Verify that the cron jobs were created successfully by using the following command. The output should match up with the figure 8.2:
crontab -l

```
[skyboxview@skybox-pri ha]$ crontab -l

*/10 * * * * /home/skyboxview/ha/skybox_watchdog.sh >/dev/null 2>&1
*/60 * * * * /home/skyboxview/ha/backupTicket.sh >/dev/null 2>&1
*/60 * * * * /home/skyboxview/ha/replicate_fw_config.sh >/dev/null 2>&1 [skyboxview@skybox-pri ha]$
```

Figure 8.2 - verifying the cron jobs on the Primary server

c) Verify that the incron jobs were created successfully by using the following. The output should match up with the figure 8.3:
incrontab -l

```
[skyboxview@skybox-pri ha]$ incrontab -l
/opt/skyboxview/data/xml_models/            IN_CLOSE_WRITE            /usr/bin/rsync            --log-
file=/home/skyboxview/ha/log/skybox_ha.log -av $@/$# skybox-sec:/opt/skyboxview/data/xml_models/
/opt/skyboxview/data/ticket_attachments/         IN_CLOSE_WRITE         /usr/bin/rsync          --log-
file=/home/skyboxview/ha/log/skybox_ha.log -av $@/$# skybox-sec:/opt/skyboxview/data/ticket_attachments/
/opt/skyboxview/data/sqlx_models/           IN_CLOSE_WRITE            /usr/bin/rsync            --log-
file=/home/skyboxview/ha/log/skybox_ha.log -av $@/$# skybox-sec:/opt/skyboxview/data/sqlx_models/
/opt/skyboxview/data/Live/reports/          IN_CLOSE_WRITE            /usr/bin/rsync            --log-
file=/home/skyboxview/ha/log/skybox_ha.log -av $@/$# skybox-sec:/opt/skyboxview/data/Live/reports/
/opt/skyboxview/data/collector/data_collector/ IN_CLOSE_WRITE /usr/bin/rsync --log-
file=/home/skyboxview/ha/log/skybox_ha.log -av $@/$# skybox-
sec:/opt/skyboxview/data/collector/data_collector
```

Figure 8.3 - verifying the incron jobs on the Primary server

d) Verify that the firewall is disabled and port forwarding isn’t enabled. Depending on the version on CentOS, the commands will vary.
For CentOS 6:
`sudo iptables --table nat --list`

```
[skyboxview@skybox-pri ha]$ sudo iptables --table nat --list
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
REDIRECT   tcp  --  anywhere             anywhere            tcp dpt:https redir ports 8443

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Figure 8.4 - verifying the firewall configuration on CentOS 6

For CentOS 7:
firewall-cmd –zone=public –list-all

```
[skyboxview@vm-ha-pri root]$ cd ~/ha
[skyboxview@vm-ha-pri ha]$ firewall-cmd --zone=public --list-all
public (default)
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports: 9443/tcp 8443/tcp 123/udp 514/udp 514/tcp 161/udp 444/tcp
  masquerade: yes
  forward-ports: port=443:proto=tcp:toport=8443:toaddr=
  icmp-blocks:
  rich rules:
```

Figure 8.5 - verifying the firewall configuration on CentOS 7

e) Finally, set up the ‘Model backup’ task on the Primary server using the required scheduling, as per the agreed schedule.

Failover Recovery

In a scenario where there is a failover event and the Primary Server has diagnosed and recovered, a failover recovery will need to be initiated to bring the primary back online as Active. Follow the steps below to recover the Primary as the Active server. Ensure that you start with the Secondary server first! Copy over model

a) Run the ./unset.sh command:

b) Run the setsecondary.sh script on the Secondary server with the following command:
./setsecondary.sh auto

```
[skyboxview@vm-ha-sec ha]$ ./setsecondary.sh auto
Removing incron.secondary file...
Tue Jul 12 12:29:52 EDT 2016  Incrontab.secondary file was removed successfully
removing table for user 'skyboxview'
table for user 'skyboxview' successfully removed
copying table from file '/home/skyboxview/ha/incron.secondary'
requesting table reload for user 'skyboxview'...
request done
Resetting skybox crontab and adding watchdog script
Detected IPTables, loading configuration...
Iptables configuration was uploaded...
Host has been successfully set as secondary
[skyboxview@vm-ha-sec ha]$
```

c) Run the setprimary.sh script on the Primary server with the following command:
./setprimary.sh auto

```
[skyboxview@vm-ha-pri ha]$ ./setprimary.sh auto
removing table for user 'skyboxview'
table for user 'skyboxview' successfully removed
copying table from file 'incron.primary'
requesting table reload for user 'skyboxview'...
request done
Resetting skybox crontab and adding watchdog script
Detected IPTables service, loading configuration...
Iptables configuration was uploaded...
No mail parameters supplied
Enabling Task Scheduling...
Host has been successfully set as primary
[skyboxview@vm-ha-pri ha]$
```

d) On both servers, run the following command: - remove
#crontab –e */10 * * * * /home/skyboxview/ha/skybox_watchdog.sh

```
[skyboxview@vm-ha-pri ha]$ #crontab –e */10 * * * * /home/skyboxview/ha/skybox_watchdog.sh
```

e) Verify the command did appear correctly in crontab with the following command:
crontab -l

```
[skyboxview@vm-ha-pri ha]$ crontab -l
*/10 * * * * /home/skyboxview/ha/skybox_watchdog.sh >/dev/null 2>&1
*/60 * * * * /home/skyboxview/ha/backupTicket.sh >/dev/null 2>&1
*/60 * * * * /home/skyboxview/ha/replicate_fw_config.sh >/dev/null 2>&1
```

f) Verify the Incrond service is up and running on both servers with the following command:
service incrond status

```
[skyboxview@vm-ha-pri ha]$ service incrond status
â–  incrond.service - SYSV: Filesystem event daemon, works like cron, but handles filesystem events
   Loaded: loaded (/etc/rc.d/init.d/incrond)
   Active: active (running) since Sun 2016-07-10 06:41:46 EDT; 2 days ago
     Docs: man:systemd-sysv-generator(8)
  Process: 1020 ExecStart=/etc/rc.d/init.d/incrond start (code=exited, status=0/SUCCESS)
 Main PID: 1023 (incrond)
   CGroup: /system.slice/incrond.service
           â””â”€1023 incrond
```

g) If the Incrond service is down, restart the service with the following command:
service incrond start

```
[skyboxview@vm-ha-pri ha]$ service incrond start
Starting incrond (via systemctl):  ==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to manage system services or units.
Authenticating as: skyboxview
Password:
==== AUTHENTICATION COMPLETE ===
[  OK  ]
```

h) Run the backup model task. The model file will be copied to the secondary and uploaded. Verify that the model from the primary is loaded to the secondary machine.

Updating to a new version of HA

The process to update the HA installation is a manual process. In order to update the HA, you will need to do the following:

1) Record the values from the skybox_ha.conf file for each respective server
2) Delete the HA directory
3) Copy over the zip file for the new version of HA to the directory of where the HA was installed to
4) Unzip the file using the unzip command. This will unzip the HA directory:
   unzip ha.zip
5) Change directory in the new ha folder
6) Update the skybox_ha.conf with the values from the previous configuration

Disabling Firewalld on CentOS 7

If you are installing the Skybox High Availability package on a CentOS 7 distribution, the Firewalld module will need to be disabled and replaced with IPTables if the Firewalld version is 0.3.9. This is due to a bug in the NAT forwarding in version 0.3.9. Follow the directions listed below in order to disable the Firewalld service and replace it with the Firewalld, if needed.

Run the following commands as the 'root' user:

a) Verify that version of Firewalld installed by running the following command if you are using a CentOS version 7 ISO by running the following command:
rpm -qa | grep firewalld

```
[13:07:49 skyboxview@skyboxapp ~]$ rpm -qa | grep firewalld
firewalld-filesystem-0.3.9-14.el7.noarch
firewalld-0.3.9-14.el7.noarch
```

b) First, install the iptables-services package:
yum install -y iptables-services

```
[root@vm-ha-sec ~]# yum install -y iptables-services
Loaded plugins: fastestmirror
base                                                                                                          | 3.6 kB  00:00:00
extras                                                                                                        | 3.4 kB  00:00:00
updates                                                                                                       | 3.4 kB  00:00:00
(1/2): extras/7/x86_64/primary_db                                                                             | 149 kB  00:00:00
(2/2): updates/7/x86_64/primary_db                                                                            | 5.7 MB  00:00:01
Determining fastest mirrors
 * base: mirror.isoc.org.il
 * extras: mirror.isoc.org.il
 * updates: centos.spd.co.il
Resolving Dependencies
--> Running transaction check
---> Package iptables-services.x86_64 0:1.4.21-16.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

==================================================================================================================================
 Package                             Arch                       Version                            Repository                Size
====================================================================================================================================
Installing:
 iptables-services                     x86_64                     1.4.21-16.el7                        base                      50 k

Transaction Summary
===================================================================================================================================
Install  1 Package

Total download size: 50 k
Installed size: 24 k
Downloading packages:
iptables-services-1.4.21-16.el7.x86_64.rpm                                                                    |  50 kB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Warning: RPMDB altered outside of yum.
  Installing : iptables-services-1.4.21-16.el7.x86_64                                                                           1/1
  Verifying  : iptables-services-1.4.21-16.el7.x86_64                                                                           1/1

Installed:
  iptables-services.x86_64 0:1.4.21-16.el7

Complete!
```

c) Stop the Firewalld service:
systemctl stop firewalld

```
[root@vm-ha-sec ~]# systemctl stop firewalld
```

d) Start the IPTables service:
systemctl start iptables

```
[root@vm-ha-sec ~]# systemctl start iptables
```

e) Disable the Firewalld service from starting up on boot:
systemctl disable firewalld

```
[root@vm-ha-sec ~]# systemctl disable firewalld
```

f) Masking the Firewalld service to prevent it from starting automatically or manually:
systemctl mask firewalld

```
[root@vm-ha-sec ~]# systemctl mask firewalld
Created symlink from /etc/systemd/system/firewalld.service to /dev/null.
```

g) Verify that the IPTables service is running:
service iptables status

```
[root@vm-ha-sec ~]# service iptables status
Redirecting to /bin/systemctl status  iptables.service
â–□ iptables.service - IPv4 firewall with iptables
   Loaded: loaded (/usr/lib/systemd/system/iptables.service; disabled; vendor preset: disabled)
   Active: active (exited) since Thu 2016-07-07 02:28:09 EDT; 4min 8s ago
 Main PID: 20139 (code=exited, status=0/SUCCESS)

Jul 07 02:28:09 vm-ha-sec systemd[1]: Starting IPv4 firewall with iptables...
Jul 07 02:28:09 vm-ha-sec iptables.init[20139]: iptables: Applying firewall rules: [  OK  ]
Jul 07 02:28:09 vm-ha-sec systemd[1]: Started IPv4 firewall with iptables.
```

Troubleshooting

Primary is not syncing to secondary:

a) Check connectivity and ssh settings (on both machines):
ping -c 4 <other server>

```
root@vm-ha-pri ~]# ping -c 4 vm-ha-sec
PING vm-ha-sec.il.skyboxsecurity.com (192.168.80.137) 56(84) bytes of data.
64 bytes from vm-ha-sec.il.skyboxsecurity.com (192.168.80.137): icmp_seq=1 ttl=64 time=0.615 ms
```

b) Run the REST PING API request with following command to verify the healthcheck is operational. The command should return a “HTTP/1.1 200 OK” response code at the bottom (in red). For primary, check port 8443 and 443 :
curl --insecure --connect-timeout 3 https://<server_IP>:<port>/skybox/webservice/jaxrsinternal/internal/healthcheck/ping -v

For example, running a REST PING API request on port 8443 on the local host:

```
[20:50:32 skyboxview@skyboxapp /opt/ha]$ curl --insecure --connect-timeout 3
https://127.0.0.1:8443/skybox/webservice/jaxrsinternal/internal/healthcheck/ping -v
* About to connect() to 127.0.0.1 port 8443 (#0)
*   Trying 127.0.0.1...
* Connected to 127.0.0.1 (127.0.0.1) port 8443 (#0)
* Initializing NSS with certpath: sql:/etc/pki/nssdb
* skipping SSL peer certificate verification
* NSS: client certificate not found (nickname not specified)
* SSL connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate:
*       subject: CN=www.skyboxsecurity.com,OU=Skybox Security,O=Skybox Security,L=San Jose,ST=CA,C=US
*       start date: Apr 28 11:13:30 2015 GMT
*       expire date: Apr 23 11:13:30 2035 GMT
*       common name: www.skyboxsecurity.com
*       issuer: CN=www.skyboxsecurity.com,OU=Skybox Security,O=Skybox Security,L=San Jose,ST=CA,C=US
> GET /skybox/webservice/jaxrsinternal/internal/healthcheck/ping HTTP/1.1
> User-Agent: curl/7.29.0
> Host: 127.0.0.1:8443
> Accept: */*
>
< HTTP/1.1 200 OK
< Strict-Transport-Security: max-age=31622400; includeSubDomains
< Date: Fri, 01 Feb 2019 01:50:36 GMT
< Content-Length: 0
< Server:
<
* Connection #0 to host 127.0.0.1 left intact
```

c) As the ‘skyboxview’ user, connect to the secondary server with the following command. You should NOT be prompted for a password:
ssh skyboxview@<other servier ip/hostname>

```
[skyboxview@vm-ha-pri ~]$ ssh skyboxview@vm-ha-sec
Last failed login: Tue Jul 12 12:52:47 EDT 2016 from 192.168.80.146 on ssh:notty
Last login: Tue Jul 12 12:26:40 2016

This option will place you in an un-restricted shell.
It should be used only to set up the network connection according to the instructions in the Appliance
Quick Start Guide.
All other interactions with the Appliance should be done via the Web Administration.

[skyboxview@vm-ha-sec ~]$
```

d) Make sure that incrond is running by running the below. You should be able to run the below with the user “skyboxview” which mean that the sudo is configure correctly.:
sudo service incrond status

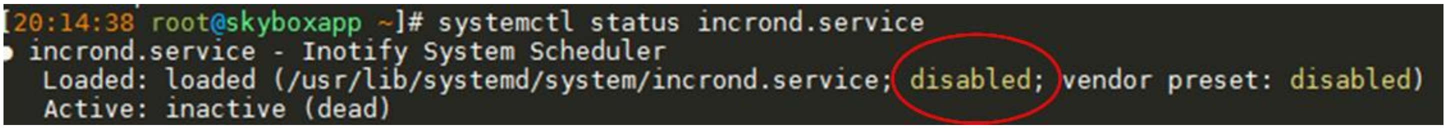
```
[sudo] password for skyboxview:
â—□ incrond.service - SYSV: Filesystem event daemon, works like cron, but handles filesystem events
   Loaded: loaded (/etc/rc.d/init.d/incrond)
   Active: active (running) since Tue 2016-07-12 12:44:49 EDT; 15min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 16989 ExecStop=/etc/rc.d/init.d/incrond stop (code=exited, status=0/SUCCESS)
  Process: 17023 ExecStart=/etc/rc.d/init.d/incrond start (code=exited, status=0/SUCCESS)
 Main PID: 17026 (incrond)
   CGroup: /system.slice/incrond.service
           â””â”€17026 incrond
```

Incron Does Not Start on Bootup:

Root Cause: In the newer versions of the Skybox ISO, like the 9.0.208 ISO, the incrond service was not enabled to run on boot.

Troubleshooting steps:
Instead of using *chkconfig*, the services uses *systemctl*. Follow the steps listed below:

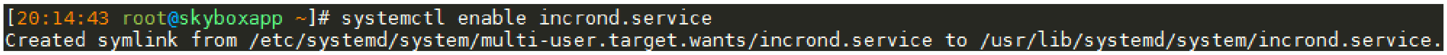
1) Run all of the following commands as the root user. Verify that the incrond service has been enabled for boot. Look for the 'disabled' keyword:
systemctl status incrond.service



```
[20:14:38 root@skyboxapp ~]# systemctl status incrond.service
● incrond.service - Inotify System Scheduler
   Loaded: loaded (/usr/lib/systemd/system/incrond.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
```

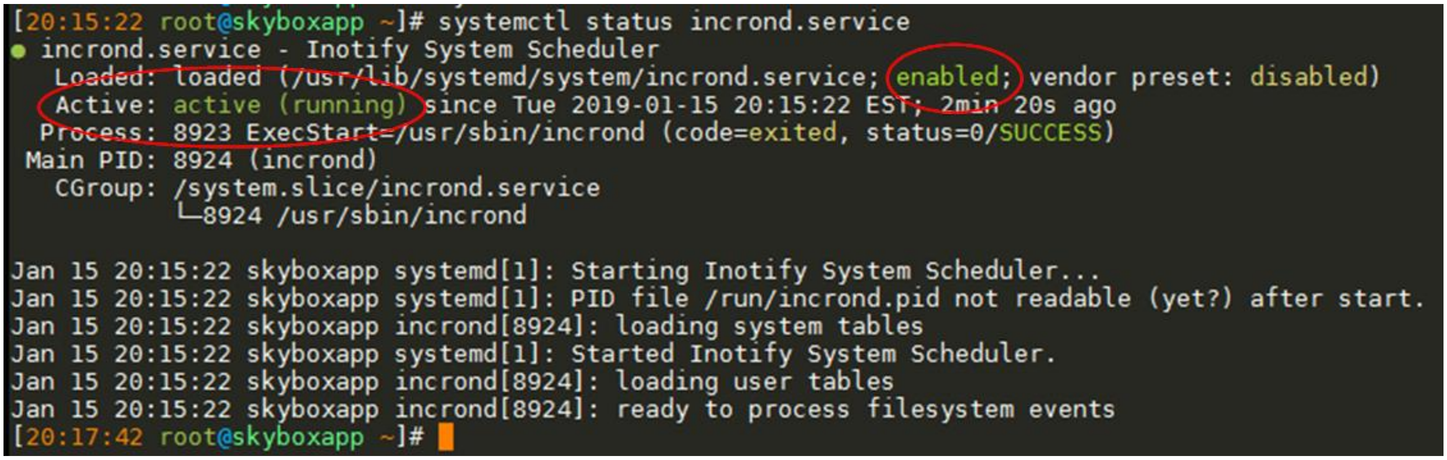
2) To start the incrond service, type of following command as root:
systemctl start incrond

3) Next step is to enable the service to run on bootup. The following command will create the symlink so that the service will start on the next boot:
systemctl enable incrond.service



```
[20:14:43 root@skyboxapp ~]# systemctl enable incrond.service
Created symlink from /etc/systemd/system/multi-user.target.wants/incrond.service to /usr/lib/systemd/system/incrond.service.
```

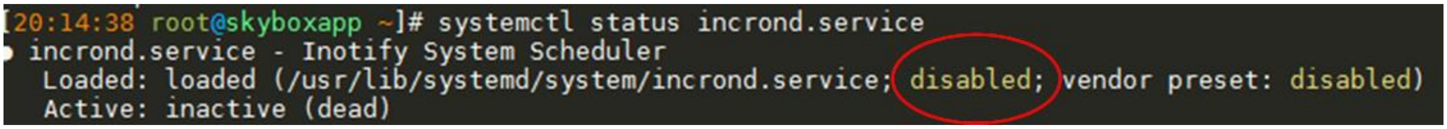
4) Afterwards, verify that the service has been started and enabled to run on boot:
systemctl status incrond.service



```
[20:15:22 root@skyboxapp ~]# systemctl status incrond.service
● incrond.service - Inotify System Scheduler
   Loaded: loaded (/usr/lib/systemd/system/incrond.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2019-01-15 20:15:22 EST; 2min 20s ago
  Process: 8923 ExecStart=/usr/sbin/incrond (code=exited, status=0/SUCCESS)
 Main PID: 8924 (incrond)
   CGroup: /system.slice/incrond.service
           └─8924 /usr/sbin/incrond

Jan 15 20:15:22 skyboxapp systemd[1]: Starting Inotify System Scheduler...
Jan 15 20:15:22 skyboxapp systemd[1]: PID file /run/incrond.pid not readable (yet?) after start.
Jan 15 20:15:22 skyboxapp incrond[8924]: loading system tables
Jan 15 20:15:22 skyboxapp systemd[1]: Started Inotify System Scheduler.
Jan 15 20:15:22 skyboxapp incrond[8924]: loading user tables
Jan 15 20:15:22 skyboxapp incrond[8924]: ready to process filesystem events
[20:17:42 root@skyboxapp ~]# 
```

Model copies to the Secondary, but doesn't load the model into Skybox

This is related to INCROND issue where the service was not enabled on. Check the status of the service using the following command:
systemctl status incrond



```
[20:14:38 root@skyboxapp ~]# systemctl status incrond.service
● incrond.service - Inotify System Scheduler
   Loaded: loaded (/usr/lib/systemd/system/incrond.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
```

REST Ping doesn't work due to a curl error (?):

The REST healthcheck API call may return a curl error.

If the curl command returns a code 56, check with the customer to verify if there is a proxy enabled between the Primary and Secondary servers.

Troubleshooting

Firewall Configurations keep resetting back to the Default Values

When the unset.sh script is ran, the Iptables and Firewalld configurations are reset. When the setprimary.sh and setsecondary.sh are ran, the configured firewall configurations replace any changes made on the server.

The work around is to make the changes permanent in relevant configuration files found in the <ha_home>/firewall directory.

FAQ – FREQUENTLY ASKED QUESTIONS

These are some of the frequently asked questions customers may ask about the Skybox High Availability Module.

Which type of cluster are you providing? (hot-stand-by, cold-stand-by)?
We are doing High Availability by using an Active/Passive in a hot-standby configuration.

Is there automatic or manual failover or no failover at all?
The Skybox High Availability package will detect a failover on the Primary and automatically switch over to the Secondary server. The High Availability will not fail back over to the Primary automatically and has to be switched over by a manual method. This is by design to allow administrators the time required to troubleshoot/diagnose any issues with the Primary server.

If there is failover, how is it triggered?
There is an automatic failover from the Primary server to the Secondary server. This is triggered by a watchdog service on the Secondary server. This service monitors the Primary server by issuing SOAP requests to verify the server is operational and the responding to requests. If the Primary service fails the status check, then the watchdog service will automatically trigger the failover and set the Secondary server as the active server.

Is the cluster active/passive or active/active?
The High Availability module is set up as an active/passive model, where the Secondary server will become active if the Primary server becomes unreachable on port 443 after the configured timeouts have been reached.

How are the IP addressing supposed to be setup? Is there a common IP? Are you using VRRP?
If the customer would like to use load-balancing with the High Availability module, they will need to use an external source, like a load balancer. For example, on F5 load balancers, the customer can create a common VIP and put the servers into a pool. The High Availability Module uses a NAT rule to port forward all traffic received on port 443 to its internal port 8443. On the F5, a health monitor can be created to check if a server is responding to HTTPS requests on port 443. If the Primary server is unreachable, the switch over will be activated and the F5 will detect the Secondary server as active and forward traffic to its port 443.

Which data is synced between cluster members?
First, the servers are not clustered. The following are synced from the Primary to the Secondary server.

- Models
- Sql models
- Reports
- Data collector
- Ticket attachments
- OS configurations

What performance impact does syncing have on the servers?
There's no performance impact on the Primary server as it's generating the backup models only. Only the Secondary server (which isn't active), will be impacted as it's uploading the model. The time is based on the size of the model.

Is it recommended to use an extra interface for syncing?

Yes, a second NIC is recommended. This will allow normal operations on the main NIC and allow the secondary NIC to be dedicated to high availability.