

Creating a cold site for use with Symantec Mobility

Note: This article assumes that the administrator has a working knowledge of Linux and a production Mobility instance. Regular backups of the following items to the cold site is required to ensure that newer devices remain managed in the event that the cold site is activated.

Contents

Register the Front-end (cold site) with the primary Mobility server.....	1
File storage.....	2
Configuration files and certificates.....	2
Restore the configuration files and storage to the new front-end.....	3
App Proxy re-registration.....	4

Register the Front-end (cold site) with the primary Mobility server

1. Follow [HOWTO101966](#) to add an additional front-end server to the production cluster.

Important: Ensure that the entire production database host is fully backed up and/or snap-shot to a working state prior to continuing with this article.

After registering the cold site FE with the production database run mysqldump to export the appstore and mdmcore databases from either the primary Mobility front-end of the database server:

2. The the following syntax, to copy the mdmcore database from the host:

```
mysqldump -u <MDM_DB_USER> -h <MDM_DB_HOST> -p --port=<MDM_DB_PORT>  
<MDM_DB_NAME> --max_allowed_packet=700M -v > ~/mdmcore.sql
```

For example:

```
mysqldump -u root -h mysql.smmglobal.net -p --port=3600 mdmcore --  
max_allowed_packet=700M -v > ~/mdmcore.sql
```

```
[root@localhost ~]# mysqldump -u root -h localhost -p --port=3600 mdmcore --max_allowed_packet=700M -v > ~/mdmcore.sql
Enter password:
-- Connecting to localhost...
-- Retrieving table structure for table AndroidDevices...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table AppConfigurations...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table AppleDevices...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table Applications...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table CMCommands...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table Certificates...
-- Sending SELECT query...
-- Retrieving rows...
```

3. Do the same as above for the appstore database.

```
-- Retrieving table structure for table xlog_logcatalog...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table xlog_logcategory...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table xlog_logdetail...
-- Sending SELECT query...
-- Retrieving rows...
-- Retrieving table structure for table xlog_logentry...
-- Sending SELECT query...
-- Retrieving rows...
-- Disconnecting from localhost...
[root@localhost ~]# █
```

Note: If blob storage is utilized the mysqldump process may take some time to complete.

File storage

Note: The following is to be completed on the primary Mobility front-end

4. If local cache storage is used, enter the following to back up the /vol1/nukona/ directory:
tar cvfP ~/vol1.tar /vol1/nukona

Configuration files and certificates

5. Make a copy of the Mobility configuration files and certificates with the following command, as root:
mkdir ~/nuk_bu
cp /usr/local/nukona/etc/settings.cfg ~/nuk_bu
cp /usr/local/nukona/certs/configurator/* ~/nuk_bu
tar cvfP ~/configs.tar ~/nuk_bu

```

[root@localhost ~]# mkdir ~/nuk_bu
[root@localhost ~]# cp /usr/local/nukona/etc/settings.cfg ~/nuk_bu
[root@localhost ~]# cp /usr/local/nukona/certs/configurator/* ~/nuk_bu
[root@localhost ~]# tar cvfP ~/configs.tar ~/nuk_bu
/root/nuk_bu/
/root/nuk_bu/sign.key
/root/nuk_bu/sign.crt
/root/nuk_bu/settings.cfg
/root/nuk_bu/gd_bundle.crt
/root/nuk_bu/certificate.pfx
/root/nuk_bu/signkey.pp
/root/nuk_bu/sign.pubkey
/root/nuk_bu/nginx.crt
[root@localhost ~]# █

```

6. Copy the configs.tar and vol1.tar to the new Mobility front-end following [HOWTO110248](#)

Restore the configuration files and storage to the new front-end

Note: The following is to be completed on the cold site front-end server.

7. Front the new front-end extract the files with the following command:

```
tar xvf <path_to_configs_tar> -C /
```

For example:

```
tar xvf ~/configs.tar -C /
```

```

[root@kms53 ~]# tar xvf ~/configs.tar -C /
tar: Removing leading `/' from member names
/tmp/nuk_bu/
/tmp/nuk_bu/settings.cfg
/tmp/nuk_bu/ad.key
/tmp/nuk_bu/sign.crt
/tmp/nuk_bu/sign.key
/tmp/nuk_bu/signkey.pp
/tmp/nuk_bu/nginx.crt
/tmp/nuk_bu/certificate.pfx
/tmp/nuk_bu/danca.crt
/tmp/nuk_bu/sign.pubkey
/tmp/nuk_bu/adkey.pp
/tmp/nuk_bu/gd_bundle.crt
[root@kms53 ~]# █

```

8. Extract the /vol1/nukona gz file with the following command, as root:
9. tar xvf <path_to_vol1_gz> -c /
For example:
10. tar xvf ~/vol1.tar -C /
11. Restore the database to the cold site DB host using the following syntax:
mysql -u <MDM_DB_USER> -h <MDM_DB_HOST> -p --port=<MDM_DB_PORT>
<MDM_DB_NAME> --max_allowed_packet=700M -v <BU_DIR>/mdmcore.sql
For example:

```
mysqld -u root -h localhost -p --port=3600 mdmcore --max_allowed_packet=700M -v <
/tmp/mdmcore.sql
```

<Image needed>

Note: The following error indicates that the database has not been created:

<Image needed>

Use create database <database_name> for example create appstore1; from the mysql shell to create the required databases.

12. Do the same for the appstore database.

13. Finally add host entry for the new database host into the machine's /etc/hosts file:

```
echo "<DBHOSTNAME> <DB_IP>" >> /etc/hosts
```

For example:

```
echo "mysql.smmglobal.net 127.0.0.1" >> /etc/hosts
```

Tip: Find the configured database hostname by entering the following: cat /usr/local/nukona/etc/settings.cfg |grep dbhost

14. Restart the Mobility services and confirm that the server is connecting to the database:

```
tail -f /var/log/nukona/appstore.log -f /var/log/nukona/celery.log
```

```
==> /var/log/nukona/appstore.log <==
timestamp=2015-12-15 06:05:01.847 +0000,logname=aclog,level=ERROR,module=authentication,function=authenticate,line=86,tenant=west,username=,sessionId=None,url=/gwapi2/inventory,msgid=,Error authenticating gateway request for gateway id=c8c97799-c9a5-4563-a585-279555136588: Instance matching query does not exist.
timestamp=2015-12-15 06:05:01.849 +0000,logname=aclog,level=DEBUG,module=middleware,function=process_response,line=809,tenant=west,username=,sessionId=None,url=/gwapi2/inventory,msgid=,timing=0.0299069881439,ms_timing=0.0222838508938, path_info=/gwapi2/inventory, msg=Request-Timing
timestamp=2015-12-15 06:05:02.50 +0000,logname=aclog,level=ERROR,module=authentication,function=authenticate,line=86,tenant=west,username=,sessionId=None,url=/gwapi2/checkin,msgid=,Error authenticating gateway request for gateway id=c8c97799-c9a5-4563-a585-279555136588: Instance matching query does not exist.
timestamp=2015-12-15 06:05:02.52 +0000,logname=aclog,level=DEBUG,module=middleware,function=process_response,line=809,tenant=west,username=,sessionId=None,url=/gwapi2/checkin,msgid=,timing=0.0469260215759,ms_timing=0.0323181152394, path_info=/gwapi2/checkin, msg=Request-Timing
==> /var/log/nukona/celery.log <==
[2015-12-15 06:05:05,763: INFO/MainProcess] msg=Received task: appstore.tasks.send_queued_log_messages[0f445701-1755-4b46-8727-68d1c5ff2fc] expires:[2015-12-15 06:07:05.714713+00:00]
[2015-12-15 06:05:05,814: INFO/MainProcess] msg=Task appstore.tasks.send_queued_log_messages[0f445701-1755-4b46-8727-68d1c5ff2fc] succeeded in 0.04661075198974s: None
[2015-12-15 06:05:06,265: INFO/MainProcess] msg=Received task: appstore.tasks.healthcheck_highpriority[1b6d2b05-3c65-4cac-9e84-78bedd92c9d] expires:[2015-12-15 06:15:06.237849+00:00]
[2015-12-15 06:05:06,274: INFO/MainProcess] msg=Task appstore.tasks.healthcheck_highpriority[1b6d2b05-3c65-4cac-9e84-78bedd92c9d] succeeded in 0.00253019388765s: None
[2015-12-15 06:05:06,278: INFO/MainProcess] msg=Received task: appstore.tasks.healthcheck_lowpriority[ca9ecd11-4af4-4727-86b3-9d851de2345] expires:[2015-12-15 06:15:06.267891+00:00]
[2015-12-15 06:05:06,281: INFO/MainProcess] msg=Task appstore.tasks.healthcheck_lowpriority[ca9ecd11-4af4-4727-86b3-9d851de2345] succeeded in 0.00197505098246s: None
[2015-12-15 06:05:06,285: INFO/MainProcess] msg=Received task: appstore.tasks.healthcheck_msa_node[0ca65d14-d155-482d-9911-b92283cc9351] expires:[2015-12-15 06:15:06.278358+00:00]
[2015-12-15 06:05:06,289: INFO/MainProcess] msg=Task appstore.tasks.healthcheck_msa_node[0ca65d14-d155-482d-9911-b92283cc9351] succeeded in 0.00235474202782s: None
```

Note: If there are entries like: Access denied for user 'root'@'localhost' (using password: YES ... confirm that the new Mobility server has permission to access the database. Consider adding new grants following: <https://dev.mysql.com/doc/refman/5.5/en/adding-users.html> Use the following command to find the DB password: cat /usr/local/nukona/etc/settings.cfg |grep dbpass

In the event that the cold site is activated any registered secure proxy will need to be re-registered with the new site. Below describes how to re-register a secure proxy with the Mobility front-end

App Proxy re-registration

1. Navigate to the FQDN of the Mobility server and to go to the Admin console > Downloads and click Download Secure Proxy:

Proxy

Secure proxy

The Secure Proxy provides access control to trusted apps from managed mobile devices to intranet servers via a proxy installed in the DMZ. The Email Proxy role provides authorized ActiveSync requests access to the Email Infrastructure. The App Proxy role relays connections to whitelisted intranet servers. **For 64-bit Redhat/CentOS 6.5**

[Download secure proxy](#)

2. Transfer this ISO file to the secure proxy front-end following: [HOWTO110248](#)
3. Create a mount point:
`mkdir /mnt/iso12`
4. Mount the secure proxy ISO to the mount point:
`mount -o loop SecureProxy_x86_64_R5.4-1.iso /mnt/iso12`
5. Change the working directory to the mount point:
`cd /mnt/iso12`
6. Execute the upgrade script with the following command, as root:
`./setup.sh --configure register`
7. Follow the prompts to complete the registration.

Note: The registration should take a moment to complete. Use a local administrative account to register the proxy.