Backup and Restore

Backups

Backups represent a snapshot of the system, including database, configuration and system applications. Backups can be created manually, scheduled automatically, or created automatically when the system is upgraded. These backups can be stored on the local file system, or to a remote network location. There is no direct requirement for Vmware snapshots. For examples of backup maintenance commands and output, refer to the topics on Scheduling and Create a Backup.

If the Cisco Unified Communications Domain Manager 10.6(1) node is not recoverable, due to for example a hardware failure, a new node can be deployed and an existing backup restored to restore the node to service.

Backup Destinations

Backups can be made to the local file system or a remote destination.

- Display available backup destinations with **backup list**.
- Add a new backup destination with **backup add <location-name> <URL>**.

Local backups are stored on a separate backup volume and the `localbackups` destination is pre-configured. If the backup volume is too small, it can be increased in size.

Note If the localbackups destination is removed or renamed, an ISO file upgrade will no longer function. Therefore, it is imperative that this destination is not removed.

Example:

backup add myserverbackup sftp://user@server/path

Backups to sftp require ssh key-based authentication to be setup.

If a common remote backup point is to be used by all nodes in the cluster, the backup destination needs to be added to each node. This can be automated by using cluster remote execution, for example:

cluster run all backup add myserverbackup sftp://user@server/path

Create Space for a Backup or Restore

If a `No space left on device` message is received during a backup or a restore, carry out the following steps:

Procedure

Step 1 In VMware, add a disk to the system:
a) Click on **VM > Edit Settings**.
b) Click **Add**.
c) Select **Hard Disk**, then **Create a new virtual disk**.
d) Set the size to be the same as the DB disk - 250GB.
e) Click **Finish**

Step 2 Log into platform account, and run **drives list**. Make note of the disk under `Unused disks:`.

Step 3 Run **drives reassign <disk from step 2> services:backups**.
Once done, all current data would have been moved to new disk and the old one can be removed from VMware. The **restore** command can now be rerun.

Adding More Space to Accommodate a Large Restore

Procedure

Step 1 Right click on the VM in the Vmware Client and click **Edit Settings**.

Step 2 On the Hardware tab, click **Add**.

Step 3 Follow the wizard to add a new hard disk to the VM with the correct size.

Step 4 If the restore size exceeds both the backup and dbroot drives size, ensure you add two hard disks to the VM. In a clustered environment, this procedure needs to be performed on all of the DB nodes.

Backup Passphrase

System backups are encrypted. The encryption key is initially set as the platform user's password as set in the installation wizard. It is recommended that this be changed after installation. This can be done by running **backup passphrase**.

The following example shows the console output:

```
platform@masternode:~$ backup passphrase
Please enter current backup passphrase
Password:
```

```
Please enter new backup passphrase
Password:
Please re-enter new backup passphrase
Password:

Backup passphrase successfully changed
```

This password needs to be kept, because restoring the backup to a new system will require this password to be the same as above.

To restore on the new system, run the above command and enter password used to create the backup

Setting up the Backup Passphrase on a New Environment

To set the backup passphrase to restore on a different environment:

Procedure

Step 1 Log into the new environment. If this is a cluster deployment, log in on the DB Primary.

Step 2 Run the **backup passphrase** command.

Step 3 Specify the current passphrase. This is normally the password of the platform user set during the deploy of the system.

Step 4 Enter the new passphrase twice.

Reassign Current Drives (Backup and DBroot)

Procedure

Step 1 Once the hard disks are added, reassign the drives using the **drives reassign <disk> <mountpoint name>** command.

Step 2 Use the **drives list** command to list the new drives added through VMware. For example, if the new drive is listed as `sdf`, use the reassign command as follows: **drives reassign sdf services:backups**.

Step 3 Similarly, to reassign the dbroot, use the reassign command as follows: **drives reassign sde mongodb:dbroot**.

Create a Backup

Backups can be created using **backup create <destination>**, for example:

backup create localbackups or **backup create myserverbackup <remote destination>**.

An example of the console output is shown below:

```
platform@myhost:~$ backup create localbackup
... collecting data
... preparing mongo data backup

... Backing up database <name>
```

```
............
... Backing up database <name>_FILES

..
... Backing up database <name>_EXPORT

.
... Not backing up database local
... Backing up database PLATFORM

..................
... Backing up database admin

.
... creating backup

 98% completed
... verifying backup
Backup was successfully created at file:///backups/a0b26a1a267a1582e2aa0258a4fa85b75d4b09bb

WARNING:  Backup maintenance of this location is not scheduled
    schedule add localbackup-maintain backup clean localbackup keep 5

platform@myhost:~$ backup list localbackup
 localbackup:
    URI: file:///backups
    Backups:
       2014-06-18 16:26
```

Backups contain all application data.

Backups can be scheduled to run automatically - refer to the **schedule** command to automated backups.

For example:

- **schedule add mybackups backup create myserverbackup**
- **schedule time mybackups 2 0**
- **schedule enable mybackups**

The cluster-wide backup can be created using the command **cluster run all backup create myserverbackup** after creating the destination with for example **cluster run all backup add myserverbackup sftp://user@server/path**. Generally, it is not recommended that a cluster wide backup be scheduled from a single node, since failure of the scheduled node could result in missing backups. Rather schedule a backup per node as above.

If a common network URI is used as backup destination across the cluster, each node's backup will be uniquely identified by its UID in the remote backup directory.

Restore the Backup

Procedure

Step 1 Copy the backup to the environment with **scp**. It will be located in the `media/` folder.

Step 2 Once the file is successfully copied, use the **backup import** command to import the backup to a location that was set up, or the default `localbackup`.

Step 3 Once the import is complete, run the **backup list** command as for example:

```
platform@Restore:~$ backup list
localbackup:
```

```
URI: file:///backups
Backups:
1 backups have been created - most recently 2015-02-26 00:22
```

Step 4 Run the **backup restore** command as for example:

```
platform@Restore:~$ backup restore localbackup 2015-02-26 00:22
```

Restore a Backup in a Clustered Environment

In a clustered environment, servers can allow for failures and can keep data intact, because when a server fails, an automatic failover occurres. If all services are kept running and data remains accessible, a backup restore would only be necessary in very specific scenarios.

Restoring a backup in a cluster would only be necessary in the following cases:

- Data Corruption (Bad Data)
- Losing the whole cluster - requiring a redeploy of new servers

Example of a Successful Restore

```
platform@Restore:~$ backup restore localbackup 2015-02-26 00:22
Services will be restarted during the restore. Do you wish to continue? y Application
voss-deviceapi processes stopped.
Stopping Application while performing database restore


--- Restore, ip=172.29.41.240, role=webproxy,application,database, loc=jhb


Application nginx processes stopped.
System restore starting from file:///backups/93d19980b574ed743d9b000a7595e42cad6a6d6b
(1424910132)
Local and Remote metadata are synchronized, no sync needed. Last full backup date: Thu Feb
 26
00:22:12 2015
Successfully restored to /backups/appdata/restore_temp_1427441507, moving to /backups/appdata
Removing temporary files in /backups/appdata/restore_temp_1427441507 local\|admin
Dropping database PLATFORM before restoring MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/PLATFORM [object Object]
Repairing database PLATFORM before restoring MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/PLATFORM [object Object]
Dropping database VOSS_FILES before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/VOSS_FILES
[object Object]
Repairing database VOSS_FILES before restoring MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/VOSS_FILES [object Object]
Dropping database VOSS before restoring MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/VOSS [object Object]
Repairing database VOSS before restoring MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/VOSS [object Object]
Trying with oplogReplay restore successfull
{'172.29.41.240': (200, '\n')}
Starting Application after performing database restore
--- Restore, ip=172.29.41.240, role=webproxy,application,database, loc=jhb
Application services:firewall processes stopped. Application nginx processes started.
Restarting services
Application processes stopped. Application processes started.
System settings have changed, please reboot using 'system reboot'
Restored successfully
You have new mail in /var/mail/platform
```

Maintaining Backups

A complete list of backups on a location can be displayed using **backup list <location>**.

Backups can be deleted using the following commands:

- **backup clean <location> keep <N>** will delete older copies so that only N copies are kept
- **backup clean <location> before <yyyy-mm-dd [HH:MM]>** will delete copies older than the specified date.

By default, there is no regular maintenance of backups, and a scheduled job should be created to perform this maintenance, for example:

- **schedule add backuprotate backup clean localbackups keep 5**
- **schedule time backuprotate rotate 3 0**
- **schedule enable backuprotate**

Exporting Backups

The backups are encrypted and may comprise of multiple files on the backup destination.

If a backup is to be exported to another system, it must be exported with the command:

backup export <location> <destination-URI> <yyyy-mm-dd [HH:MM]>

For example:

```
backup export local backup destination-location 2014-04-30 11:16
```

In turn, the backup can be imported on the remote server using **backup import <source-URI>**.

Scheduling

Any CLI command can be scheduled to run automatically, including but not restricted to backups and security upgrades.

By default there is no backup maintenance scheduled. Backup maintenance can be scheduled with the number of copies to be kept.

The automated job schedule format is as follows:

- **schedule add <job-name> <user-command>**
- **schedule time <job-name> <hour> <minute>**
- **schedule time <job-name> every <N> hours**
- Alternatively the job can be scheduled to run every week on Monday with **schedule time <job-name> weekly 1**; where 0 is Sunday, 1 is Monday, 2 is Tuesday, 3 is Wednesday, 4 is Thursday, 5 is Friday and 6 is Saturday
- **schedule enable <job-name>**

Example:

schedule add mybackups backup create localbackups

schedule time mybackups 2 0

schedule time mybackups weekly 0

schedule enable mybackups

Among the tasks that can be scheduled are:

- Backup creation, e.g. **schedule add backupme backup create localbackup**
- Backup maintenance, e.g. **schedule add backupclean backup clean localbackup keep 5**
- Health reports, e.g. **schedule add reports diag report**

DR Failover

The Cisco Unified Communications Domain Manager 10.6(1) system makes use of database replication facilities during normal operation. During a failover, if 50% or more of the service resources are lost, the system will no longer function without manual intervention. In this case, the following process should be followed.

Procedure

Step 1 Display the current cluster topology using **cluster status**.

Step 2 Remove the dead nodes using **cluster del <ip>**.

Step 3 Once the cluster topology is adjusted, the cluster must be reprovisioned using **cluster provision**.

Step 4 Afterward, the cluster status can be rechecked with **cluster status**.

DR Failover and Recovery

DR Failover and Recovery Scenarios

A number of failover scenarios and recovery steps are shown. In each case, a node topology is assumed, a node failure scenario is indicated and a set of recovery steps are provided.

Cisco Unified Communications Domain Manager System Recovery is supported from the following failover scenarios:

- Loss of a non-primary node in the Primary site
- Loss of a non-primary server in the DR site
- Loss of the Primary Database Server
- Loss of a Primary Site
- Loss of a DR Site

For the scenarios below, the following procedures and definitions apply:

- In the event of a network failure or a temporary network outage affecting a single node, the node will be inaccessible and the cluster will respond in the same way as if the node had failed. If network connectivity is then restored, no action is required, because the node will again start communicating with the other nodes in the cluster, provided no changes were made to that node during the outage window.
- In a clustered deployment, the datacentre would typically be two different datacentres, for example "Virginia" and "Seattle". These can be thought of as a primary site and a DR (Disaster Recovery) site in case of a failure in the primary site. These two datacentres can exist on the same physical hardware, so the separation of the cluster is into two sets of three nodes.

  When datacentres are defined during installation, the nodes of a cluster may or may not be in the same physical location. The cluster is designed to communicate across all nodes, regardless of their physical location.

Scenario: Loss of a Non-primary Node in the Primary Site

- The administrator deployed the cluster into a Primary and DR site.
- The cluster is deployed according to the *Cisco Unified Communications Domain Manager Planning and Install Guide*.
- The example is a typical cluster deployment: 6 nodes, where 4 nodes are database servers and 2 nodes are proxy servers. The design is preferably split over 2 physical data centers.

```
Data Centre: jhb
             application : AS01[172.29.42.100]
                           AS02[172.29.42.101]

             webproxy :    PS01[172.29.42.102]
                           AS01[172.29.42.100]
                           AS02[172.29.42.101]

             database :    AS01[172.29.42.100]
                           AS02[172.29.42.101]

Data Centre: cpt
             application : AS03[172.29.21.100]
                           AS04[172.29.21.101]

             webproxy :   PS02[172.29.21.102]
                          AS03[172.29.21.100]
                          AS04[172.29.21.101]

             database :   AS03[172.29.21.100]
                          AS04[172.29.21.101]
```

Node Failure

- Normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a loss of a primary node is experienced. In this scenario `AS02[172.29.42.101]` failed while transactions were running.
- Examine the cluster status running **cluster status** to determine the failed state:

```
platform@AS01:~$ cluster status


        Data Centre: unknown
        application : unknown_172.29.42.101[172.29.42.101] (not responding)
```

```
webproxy : unknown_172.29.42.101[172.29.42.101] (not responding)

database : unknown_172.29.42.101[172.29.42.101] (not responding)


Data Centre: jhb
application : AS01[172.29.42.100]

webproxy : PS01[172.29.42.102]
           AS01[172.29.42.100]

database : AS01[172.29.42.100]


Data Centre: cpt
application : AS03[172.29.21.100]
              AS04[172.29.21.101]

webproxy : PS02[172.29.21.102]
           AS03[172.29.21.100]
           AS04[172.29.21.101]

database : AS03[172.29.21.100]
           AS04[172.29.21.101]
```

- At this point, *all* transactions that are currently in flight are lost and will not recover.
- The lost transactions have to be rerun.
- With the database server `AS02[172.29.42.101]` still down, replaying the failed transactions are successful.

Recovery Steps if the server that is lost, is unrecoverable:

1. A new unified node needs to be deployed. Ensure the server name, IP information and datacentre name is the same as on the server that was lost.
2. Run **cluster del 172.29.42.101**, because this server no longer exists.
3. Delete all database weights (**database weight del <ip>**), for example **database weight del 172.29.42.101** .
4. Run **cluster provision** before the new server is added.
5. Switch on the newly installed server.
6. If the node will be a unified, application or web proxy node, run **cluster prepnode** on it.
7. Run **cluster add <ip>** of the new unified server to add it to the existing cluster.
8. Add database weights so that the are weights distributed throughout the cluster (**database weight add <ip> <weight>**).
9. Run **cluster provision** to join the new unified node to the cluster communications.

Scenario: Loss of a Non-primary Server in the DR Site

- The administrator deployed the cluster into a Primary and DR site.
- The cluster is deployed according to the *Cisco Unified Communications Domain Manager Planning and Install Guide*.

- The example is a typical cluster deployment: 6 nodes, where 4 nodes are database servers and 2 nodes are proxy servers. The design is preferably split over 2 physical data centers.

Node Failure

- Normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a loss of a primary node is experienced. In this scenario `AS02[172.29.42.101]` failed while transactions were running.
- Examine the cluster status running **cluster status** to determine the failed state:

```
Data Centre: unknown
          application : unknown_172.29.42.101[172.29.42.101] (not responding)

          webproxy : unknown_172.29.42.101[172.29.42.101] (not responding)

          database : unknown_172.29.42.101[172.29.42.101] (not responding)


Data Centre: jhb
          Application : AS01[172.29.42.100]
                        AS02[172.29.42.101]

          webproxy :    PS01[172.29.42.102]
                        AS01[172.29.42.100]
                        AS02[172.29.42.101]

          database :    AS01[172.29.42.100]
                        AS02[172.29.42.101]

Data Centre: cpt
          application : AS03[172.29.21.100]

          webproxy :    PS02[172.29.21.102]
                        AS03[172.29.21.100]

          database :    AS03[172.29.21.100]
```

- At this point, *all* transactions that are currently in flight are lost and will not recover.
- The lost transactions have to be rerun.
- With the database server `AS02[172.29.42.101]` still down, replaying the failed transactions are successful.

Recovery Steps if the server that is lost, is unrecoverable:

1. A new unified node needs to be deployed. Ensure the server name, IP information and datacentre name is the same as on the server that was lost.
2. Run **cluster del 172.29.42.101**, because this server no longer exists.
3. Delete all database weights (**database weight del <ip>**), for example **database weight del 172.29.42.101**
4. Run **cluster add <ip>** before the new server is added.
5. Switch on the newly installed server.
6. If the node will be a unified, application or web proxy node, run **cluster prepnode** on it.
7. Run **cluster add <ip>** of the new unified server to add it to the existing cluster.

8 Add database weights so that the weights are distributed throughout the cluster (**database weight add <ip> <weight>**).

9 Run **cluster provision** to join the new unified node to the cluster communications.

Scenario: Loss of the Primary Database Server

- The administrator deployed the cluster into a Primary and DR site.
- The cluster is deployed according to the *Cisco Unified Communications Domain Manager Planning and Install Guide*.
- The example is a typical cluster deployment: 6 nodes, where 4 nodes are database servers and 2 nodes are proxy servers. The design is preferably split over 2 physical data centers.

Node Failure

- Normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a loss of a primary database server is experienced. In this scenario `AS01[172.29.42.100]` failed while transactions were running.
- Examine the cluster status running **cluster status** to determine the failed state:

```
Data Centre: unknown
        application : unknown_172.29.42.100[172.29.42.100] (not responding)

        webproxy : unknown_172.29.42.100[172.29.42.100] (not responding)

        database : unknown_172.29.42.100[172.29.42.100] (not responding)


Data Centre: jhb
        application : AS02[172.29.42.101]

        webproxy :    PS01[172.29.42.102]
                      AS02[172.29.42.101]

        database :    AS02[172.29.42.101]

Data Centre: cpt
        application : AS03[172.29.21.100]
                      AS04[172.29.21.101]

         webproxy :   PS02[172.29.21.102]
                      AS03[172.29.21.100]
                      AS04[172.29.21.101]

         database :   AS03[172.29.21.100]
                      AS04[172.29.21.101]
```

- The loss of the Primary database server will cause an election and the node with the highest weighting still running will become primary. The election itself may take 10-30 seconds.
- Check the weights set in the cluster configuration: **database weight list**

```
platform@AS01:~$ database weight list
    172.29.21.100:
        weight: 10
    172.29.21.101:
        weight: 20
    172.29.42.100:
        weight: 50
```

```
    172.29.42.101:
        weight: 40
```

- The primary node 172.29.42.100 failed and therefore node 172.29.42.101 will become the primary node after election.
- To find the primary database, run **database primary**.

```
platform@AS02:~$ database primary
172.29.42.101
```

- At this point *all* transactions that are currently in flight are lost and will not recover.
- The lost transactions have to be rerun.
- With the database server `AS01[172.29.42.100]` still down, replaying the failed transactions is successful.

Recovery Steps if the server that is lost, is unrecoverable:

1. A new unified node needs to be deployed. Ensure the server name, IP information and datacentre name is the same as on the server that was lost.
2. Run **cluster del 172.29.42.100**, because this server no longer exists.
3. Delete all database weights (**database weight del <ip>**), for example **database weight del 172.29.42.101**
4. Run **cluster provision** before the new server is added.
5. Switch on the newly installed server.
6. If the node will be a unified, application or web proxy node, run **cluster prepnode** on it.
7. Run **cluster add <ip>** of the new unified server to add it to the existing cluster.
8. Add database weights so that the weights are distributed throughout the cluster (**database weight add <ip> <weight>**).
9. Run **cluster provision** to join the new unified node to the cluster communications.

Scenario: Loss of a Primary Site

- The administrator deployed the cluster into a Primary and DR site.
- The cluster is deployed according to the *Cisco Unified Communications Domain Manager Planning and Install Guide*.
- The example is a typical cluster deployment: 6 nodes, where 4 nodes are database servers and 2 nodes are proxy servers.The design is preferably split over 2 physical data centers.
- The cluster might also be in two geographically dispersed areas. The cluster has to be installed in two different site names or data center names. In this scenario, a portion of the cluster is in Johannesburg and the other is in Cape Town, South Africa:

```
Data Centre: jhb
        application : AS02[172.29.42.101]

        webproxy :    PS01[172.29.42.102]
                      AS02[172.29.42.101]

        database :    AS02[172.29.42.101]
```

```
Data Centre: cpt
        application : AS03[172.29.21.100]
                      AS04[172.29.21.101]

         webproxy :   PS02[172.29.21.102]
                      AS03[172.29.21.100]
                      AS04[172.29.21.101]

         database :   AS03[172.29.21.100]
                      AS04[172.29.21.101]
```

Primary site failure

- Normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a loss of a Primary site is experienced. In this scenario, `AS01[172.29.42.100]`,`AS01[172.29.42.101]` and `AS01[172.29.42.100]` failed while transactions were running.
- At this point, *all* transactions that are currently in flight are lost and will not recover.
- Examine the cluster status by running **cluster status** to determine the failed state:

```
Data Centre: unknown
        application : unknown_172.29.42.100[172.29.42.100] (not responding)
                              unknown_172.29.42.101[172.29.42.101] (not responding)

        webproxy : unknown_172.29.42.100[172.29.42.100] (not responding)
                   unknown_172.29.42.101[172.29.42.101] (not responding)
                   unknown_172.29.42.102[172.29.42.102] (not responding)

        database : unknown_172.29.42.100[172.29.42.100] (not responding)
                   unknown_172.29.42.101[172.29.42.101] (not responding)


Data Centre: jhb
             application :


             webproxy :


             database :


Data Centre: cpt
           application :   AS03[172.29.21.100]
                           AS04[172.29.21.101]

             webproxy :   PS02[172.29.21.102]
                          AS03[172.29.21.100]
                          AS04[172.29.21.101]

             database :    AS03[172.29.21.100]
                           AS04[172.29.21.101]
```

- The cluster will be not be operational and manual intervention is needed to recover if a continued flow of transactions is required with a minimum of downtime.
- To recover the lost nodes and if they are unrecoverable, carry out the following recovery steps.

Recovery Steps:

1. Run **cluster del <ip>** on the failed nodes from the existing half of the cluster.
2. Remove all database weights from the cluster: **database weight del <ip>**

3. At this point you do have the option to provision half the cluster for a faster uptime of your DR site. Only the DR site will be operational after the provision.
4. If you choose to bring the full cluster back up, you need to redeploy the primary site nodes if the nodes are unrecoverable without a doubt.
5. Run **cluster provision** before the new servers are added.
6. Deploy 3 nodes: 2 as unified nodes and 1 as a proxy node.
7. If the node will be a unified, application or web proxy node, run **cluster prepnode** on it.
8. After the redeployment, at this stage run **cluster add <ip>** for the nodes to become part of the cluster.
9. Add the database weights back, using **database weight add <ip> <weight>**
10. Run **cluster provision primary** to ensure that a primary is selected for the provisioning stage.

Scenario: Loss of a DR Site

- The administrator deployed the cluster into a Primary and DR site.
- The cluster is deployed according to the *Cisco Unified Communications Domain Manager Planning and Install Guide*.
- The example is a typical cluster deployment: 6 nodes, where 4 nodes are database servers and 2 nodes are proxy servers. The design is preferably split over 2 physical data centers.
- The cluster might also be in two geographically dispersed areas. The cluster has to be installed in two different site names or data center names. In this scenario, a portion of the cluster is in Johannesburg and the other is in Cape Town, South Africa:

```
Data Centre: jhb
        application : AS02[172.29.42.101]

        webproxy :    PS01[172.29.42.102]
                      AS02[172.29.42.101]

        database :    AS02[172.29.42.101]

Data Centre: cpt
        application : AS03[172.29.21.100]
                      AS04[172.29.21.101]

         webproxy :   PS02[172.29.21.102]
                      AS03[172.29.21.100]
                      AS04[172.29.21.101]

         database :   AS03[172.29.21.100]
                      AS04[172.29.21.101]
```

DR site failure

- Normal operations continue where the cluster is processing requests and transactions are committed successfully up to the point where a loss of a DR site is experienced. In this scenario, `AS03[172.29.21.100]`,`AS04[172.29.21.101]` and `PS02[172.29.21.100]` failed while transactions were running.
- At this point, *all* transactions that are currently in flight are lost and will not recover. The lost transactions have to be rerun.
- With the DR site still down, replaying the failed transactions is successful

- Examine the cluster status by running **cluster status** to determine the failed state:

```
Data Centre: unknown
      application : unknown_172.29.21.100[172.29.21.100] (not responding)
                        unknown_172.29.21.101[172.29.21.101] (not responding)

       webproxy : unknown_172.29.21.100[172.29.21.100] (not responding)
                  unknown_172.29.21.101[172.29.21.101] (not responding)
                  unknown_172.29.21.102[172.29.21.102] (not responding)

       database : unknown_172.29.21.100[172.29.21.100] (not responding)
                  unknown_172.29.21.101[172.29.21.101] (not responding)


Data Centre: jhb
             application : AS01[172.29.42.100]
                           AS02[172.29.42.101]

             webproxy :    PS01[172.29.42.102]
                           AS01[172.29.42.100]
                           AS02[172.29.42.101]

             database :    AS01[172.29.42.100]
                           AS02[172.29.42.101]

Data Centre: cpt
             application :


             webproxy :


             database :
```

- The cluster will be operational, but only on the Primary Site.
- You need to recover the lost nodes and if they are unrecoverable. Follow the recovery steps below.

Recovery Steps

1. Run **cluster del <ip>** on the failed nodes from the existing half of the cluster.
2. Remove all database weights from the cluster: **database weight del <ip>**
3. Run **cluster provision** before a new server is added.
4. If you choose to bring the full cluster back up, you need to redeploy the DR site nodes if the nodes are unrecoverable.
5. Deploy 3 nodes: 2 as unified nodes and 1 as a proxy node.
6. If a node will be a unified, application or web proxy node, run **cluster prepnode** on it.
7. After the redeployment, at this stage run **cluster add <ip>** for the nodes to become part of the cluster.
8. Add the database weights back, using **database weight add <ip> <weight>**
9. Run **cluster provision primary** to ensure that a primary is selected for the provisioning stage.

High Availability Disaster Recovery

High Availability Overview

High Availability (HA) is an approach to IT system design and configuration that ensures Cisco Unified Communications Domain Manager is operational and accessible during a specified time frame. This is achieved using redundant hardware and resources. If there is a failure, an automatic failover will occur to the secondary database node.

Default High Availability Disaster Recovery Scenario

Cisco Unified Communications Domain Manager 10.6(1) supports using off-the-shelf VMware tools.

High Availability is implemented using VMware HA clusters, with data accessed via a central storage facility (SAN). VMware monitors the primary server, and should it fail, another instance of the VM is automatically started on a different hardware instance. Since data is shared on the SAN, the new HA instance will have access to the full dataset.

Disaster Recovery is implemented by streaming data updates to a separate DR instance that remains powered on. If the primary server fails, the DR instance can take over operation. The switch-over to DR instance is scripted, but must be invoked manually.

During a HA failover, the HA instance assumes the primary IP address, and no reconfiguration of other UC elements is required. However, in the case of a DR failover, interaction with other UC elements should be considered:

- DNS can be used effectively to provide hostname abstraction of underlying IP addresses. In such a case, a DNS update will allow existing UC elements to seamlessly interact with the new DR instance.
- If DNS is not available, and the UC elements cannot be configured with the IP address of the DR instance, it is necessary for the DR instance to assume the primary IP address. In such a case, the DR and the primary IP addresses can be swapped using the CLI interface. Standard networking practices should be employed to ensure that the IP address is correctly routed, e.g. Stretched layer-2 vLAN, and ensuring that the Primary and DR instances are not operated with the same IP address.

The following failure points should be considered:

- Since the HA instance is started automatically if the primary instance fails, a slight interruption in service is expected, including VMware polling latency in determining that the primary server has failed, and the startup delay of the HA instance. This delay is around 3 minutes.
- If data is corrupted on the SAN, the HA instance will start with the same corrupt code and data instances.
- Since VMware is checking only for VM liveness, it is not able to check that the primary instance is functionally active.
- Data updates are transported to the DR instance. If data updates cannot be shipped by the primary instance, SNMP traps are generated informing administration of the problem. However, if this is not fixed in a timely manner, it is possible for the DR instance to become out of sync. These delays could result in data loss between the primary and DR instances. Database updates are scheduled every 3 minutes and/or 16Mb.

HA and DR Scenario with Cisco VMDC Geo-Redundancy Architecture

High Availability and Disaster Recovery instances can be geo-relocated at will within the capabilities of the underlying network architecture.

For example, it is feasible to extend a VMware High Availability cluster geographically using high speed data links and layer-2 stretched vLANs.

Disaster Recovery as implemented by the Cisco Unified Communications Domain Manager 10.6(1) system lends itself to geographical separation with streaming data replication to a second powered-on instance.

Interaction with other UC elements must be considered within the capabilities of the network, using either DNS for seamless transition, or IP reconfiguration either within the UC elements or the Cisco Unified Communications Domain Manager 10.6(1) system.

Configuring a HA System Platform on VMWare

This is an optional step, however, for production servers it is highly recommended that they are run in a HA deployment configuration. This can be done by the client, but should be checked by a system representative

Procedure

Step 1 Log into VMware VSphere, then select **File > New >Cluster**.

Step 2 Enter the Name, and select the Turn on VMware HA checkbox.

Step 3 Make sure that the Enable Host Monitoring checkbox and Enable: Do not power on VMs that violate availability constraints radio buttons are selected.

Step 4 Select the required default restart priority.

Step 5 Select the VM Monitoring Only option from the VM Monitoring drop-down list, and set the Default Cluster Settings/Monitoring sensitivity to High.

Step 6 Select the Disable EVC radio button, unless you know the exact version of CPU technologies that are enabled on your system.

Step 7 Select the Store the swapfile in the same directory as the virtual machine (recommended) radio button.

Step 8 Ensure the settings are all correct and click the **Finish** button.

Step 9 Drag all of the machines that will be used into the newly created cluster.

Step 10 Once done, they will be listed below the new cluster, with any VM's that were moved into the root of the cluster.

Step 11 Select each of the Machines in the cluster then select the Configuration tab.

Step 12 If Time Configuration is displayed in red, select Properties, then click the **Options** button.

Step 13 Select NTP Settings, and then click the **Add** button.

Step 14 Select the Restart NTP service to apply changes checkbox, and then click the **OK** button.

Step 15 Select the relevant Cluster, and then select the Summary tab. There should be no configuration issues listed.

Troubleshooting

'No Space Left on Device' Error

You receive the following error message while backing up or restoring Cisco Unified Communications Domain Manager 10.6(1) on a virtual machine: 'No Space Left on Device.' You can create a new virtual disk on the node with the primary database and then reassign the Cisco Unified Communications Domain Manager 10.6(1) data to the new disk. The new disk has enough space for you to perform the backup or restore operation.

Procedure

Step 1 Turn off the virtual machine that contains the primary database.

Step 2 In VMware, add a disk on the node that contains the primary database:

a) From the **VM** menu, click **Edit Settings**.
b) Click **Add**. The Add Hardware Wizard opens.
c) Select **Hard Disk** and then click **Next**.
d) Select **Create a new virtual disk** and then click **Next**.
e) Set the capacity to be the same as the database disk: 250 GB.
f) Accept the default file name and location, or click **Browse** to select a different location.
g) Click **Finish**.

Step 3 Turn on the virtual machine. Your guest operating system recognizes the new virtual disk as a new, blank hard disk.

Step 4 Log in to the platform account on the virtual machine and run the **drives list** command.

Step 5 In the command output, note the following information, which you will use in step 6:

- The name of the new disk in the 'Unused disks' section
- The identifier of the current disk, 'services:backups,' in the 'Used disks and mountpoints' section

Step 6 Run the following command: **drives reassign <new disk name> services:backups**
All current data is moved to the new disk. You can continue with your backup or restore operation.

Loss of the whole cluster and redeploying new servers

The high level redeploy and backup restore steps are as follows:

- Redeploy the cluster.
- Store the backup you want to restore in a different location.
- Recreate the remote backups on the primary node using **backup create <loc-name> <URI>**.
- Copy the saved backup under the new UID folder on the remote backup server.
- Do a **backup list**.

For example:

```
pxetest:
     URI: sftp://sftpusr:********@172.29.42.249/AS03
     Backups:
          1 backups have been created - most recently 2014-08-21 10:24
```

A **backup restore** can now be run on the primary.

The example console output below shows the steps and process:

Identifying the database primary:

```
platform@AS01:~$ database primary
172.29.42.100
```

Listing the backups:

```
platform@AS01:~$ backup list
 localbackup:
     URI: file:///backups
     Backups:
        2 backups have been created - most recently 2014-08-21 17:59
 pxetest:
     URI: sftp://sftpusr:********@172.29.42.249/AS01
     Backups:
        2 backups have been created - most recently 2014-08-21 12:54

You have new mail in /var/mail/platform
```

Restoring the backup:

```
platform@AS01:~$ backup restore pxetest 2014-08-21 12:54
Services will be restarted during the restore. Do you wish to continue? y
Application <name>-deviceapi processes stopped.
Stopping Application while performing database restore


---------- AS02, ip=172.29.42.101, role=webproxy,application,database, loc=cpt


Stopping nginx:proxy

---------- AS01, ip=172.29.42.100, role=webproxy,application,database, loc=cpt


Application nginx processes stopped.


---------- AS02, ip=172.29.42.101, role=webproxy,application,database, loc=cpt


Application nginx processes stopped.


---------- AS04, ip=172.29.21.191, role=webproxy,application,database, loc=jhb


Application nginx processes stopped.


---------- AS03, ip=172.29.21.190, role=webproxy,application,database, loc=jhb


Application nginx processes stopped.
System restore starting from
sftp://sftpusr:sftpusr@172.29.42.249/AS01/ba1e37deff1309edcc2595bf46c6bfc2a99ca164
Local and Remote metadata are synchronized, no sync needed.
Last full backup date: Thu Aug 21 12:54:25 2014
Successfully restored to /backups/appdata/restore_temp_1408699183, moving to /backups/appdata
Removing temporary files in /backups/appdata/restore_temp_1408699183
local
Dropping database <name>_FILES before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/<name>_FILES
[object Object]
```

```
Repairing database <name>_FILES before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/<name>_FILES
[object Object]
Dropping database PLATFORM before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/PLATFORM
[object Object]
Repairing database PLATFORM before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/PLATFORM
[object Object]
Dropping database <name> before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/<name>
[object Object]
Repairing database <name> before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/<name>
[object Object]
Dropping database <name>_LOCKING before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/<name>_LOCKING
[object Object]
Repairing database <name>_LOCKING before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/<name>_LOCKING
[object Object]
Dropping database admin before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/admin
[object Object]
Repairing database admin before restoring
MongoDB shell version: 2.6.1
connecting to: 127.0.0.1:27020/admin
[object Object]
Trying with oplogReplay
Trying without oplogReplay
restore successfull
Restarting services

Application processes stopped.

Application processes started.

System settings have changed, please reboot using 'system reboot'
```