PERFORCE

Version everything.

Perforce Software, Inc.
510.864.7400
consulting@perforce.com
www.perforce.com

Disaster Recovery Fail-over Plan

Table of Contents

1 Overview

The DR server is being replicated using the Perforce built in server to server replication via the "p4 pull" command and a rsync script to replicate checkpoints. This document provides the steps to fail over from the main server to the DR server.

NOTE: This is a generic document that needs to be customized for the specific set up in your environment.

2 Fail Over Procedure

2.1 Check replica status

This is just a verification that replication is up to date before we start the whole process.

On the replica machine, run `p4 pull -lj` to check the status of the replication. The command should return two matching journal sequence numbers.

2.2 Stop access to master server – done on master server

Change the `/p4/1/bin/p4broker.cfg` file to block all user access to the server. There is an entry in the config file to do this vi search for words "block all access", just uncomment that entry in the config file, and save the file. The broker does not have to be restarted to pick up the change. It will take effect immediately.

If you are not running a broker, just validate the replica is up to date and then shut down the master, then the replica.

2.3 Check replica status

On the replica machine, run `p4 pull -lj` to check the status of the replication. You want to wait for the replica to fully catch up before shutting down the master and replica instances. Once you have verified the replication is up to date:

stop the master by running `/p4/1/bin/p4d_1_init stop` on the master server

stop the replica by running
`/p4/common/bin/p4master_run 1 /p4/1/bin/p4_1 cluster end-journal`
`/p4/1/bin/p4d_1_init stop` on the replica server
Now look in /p4/1/checkpoints.rep for the highest numbered p4_1.jnl.XXX file in the folder and move that file to /p4/1/logs/journal replacing the empty one that is there.

Next, run:
/p4/1/bin/p4d_1 -r /p4/1/root -J /p4/1/logs/journal -jj /p4/1/checkpoints/p4_1

2.4 Change the replica to become the master

Change the DNS entry or move the VIP from the master to the new master if you are using a VIP address.

On the replica server, edit `/p4/<INSTANCE>/root/server.id` and change the `name to the name of the MASTER server id` and save the file. Now you can restart the server as the master by running `/p4/1/bin/p4d_1_init start` .

After this is done, the former replica is now the master.

Next run:

```
rm /p4/1/root/state
rm /p4/1/root/statejcopy
rm /p4/1/root/rdb.lbr
```

2.5 Update the crontab (Run all commands as perforce user)

On the new master server, originally the replica, run `crontab /p4/p4.crontab` to reset the crontab to run the master server crontab settings.

On the original master server, now the replica, run `crontab /p4/p4.crontab.replica` to change it to run the replica scripts.

It is a good idea to use crontab -l > p4.crontab on the master and crontab -l > p4.crontab.replica on the replica before you make these changes on each machine just to be sure that you have the current crontab settings saved. You can copy these files to the other machine to make sure you load the current settings on the other machine as well.

2.6 Change the original master to become the replica

On the original master, edit `/p4/1/root/server.id` and change the name to the name of the REPLICA server id and save the file.
Move `/p4/1/logs/journal`, e.g. to `/p4/1/logs/journal.orig.master`
Delete the following files if they exist:

```
/p4/1/root/state
/p4/1/root/statejcopy
/p4/1/root/rdb.lbr
```

You also need to login into the master server as the service user on the replica so that replication can run.

Run:
```
/p4/common/bin/p4master_run <instance> p4 -p ssl:master:port -u service trust (Only if your site
is using SSL)
/p4/common/bin/p4master_run <instance> p4 -p ssl:master:port login <
/p4/common/bin/adminpass
/p4/common/bin/p4master_run <instance> p4 -p ssl:master:port login service
```

Now, start the server as a replica by running `/p4/1/bin/p4d_1_init start`.
Edit `/p4/1/bin/p4broker.cfg` and comment out the lines that were blocking all user access (search for "block all access") and save the file if you are using a broker.

Check replication by running:

```
p4 login  < /p4/common/bin/adminpass
p4 pull -lj
```

on the master server. You should see two matching journal sequence numbers. The `p4 pull -lj` will only succeed when run on a replica pulling from a master. So this proves that the roles have been switched.

Finally, run these final two steps:

On the new replica server, run the `sync_replica.sh` command that cron normally runs in order to make sure it is working properly.

```
/p4/common/bin/p4master_run 1 /p4/common/bin/sync_replica.sh
```

NOTE 1: *If the master server has been down for more than 7 days*, you will have to reset the replica as follows before starting it. The reason for the 7 days value is that the number of old checkpoints and journals to keep is set to 7 in the `p4_vars` file. The replication uses the old journals in order to catch up from where it last stopped, so if the old ones have rotated off, you have to reset the replica with a new checkpoint and updated versioned files.

To reset the replica, from the replica run:

If your servers are set up to rsync without a password, you can reset the replica by running:

```
/p4/common/bin/p4master_run 1 /p4/common/bin/weekly_sync_replica.sh
```

`weekly_sync_replica.sh` performs the following steps:

```
rsync -avz --delete perforce@master_server:/p4/1/checkpoints/
     /p4/1/checkpoints
rm -r -f /p4/1/root/db.*
rm -r -f /p4/1/offline_db/db.*
rm -r -f /p4/1/logs/journal
ls -lah /p4/1/checkpoints
```
Make note of the highest numbered p4_1.ckp.#.gz file in that directory. Assume it is 10 for this example:

```
/p4/1/bin/p4d_1 -r /p4/1/root -jr -z /p4/1/checkpoints/p4_1.ckp.10.gz
/p4/1/bin/p4_1 -p master_server:port -u service login
```

```
/p4/1/bin/p4d_1_init start
/p4/1/bin/p4d_1 -r /p4/1/offline_db -jr –z
      /p4/1/checkpoints/p4_1.ckp.10.gz
```

Optionally, you may want to run:
```
rsync -avz --delete perforce@master_server:/p4/1/depots/ /p4/1/depots
```

or

```
/p4/common/bin/p4master_run 1 /p4/common/bin/p4verify.sh
```

to make sure the depot files are up to date.

NOTE 2: If the master server is inoperable, follow these steps:

   (1) On the replica server, run `/p4/1/bin/p4d_1_init stop`
   (2) Do the steps in Section 2.4 above