



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

Commit/Edge Failover Plan

Table of Contents

1	Overview.....	3
2	Fail Over Procedure.....	3
2.1	Check replica status.....	3
2.2	Stop access to master server – done on master server.....	3
2.3	Rotate the journal.....	3
2.4	Check replica status.....	3
2.5	Shutdown the replica and master instance(s).....	4
2.6	Copy the master journal.....	4
2.7	Change the replica to become the master.....	4
2.8	Update the crontab (Run all commands as perforce user).....	4
2.9	Change the original master to become the replica.....	5

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.

Also, this process depends on you using `sync_shared_replica.sh` on the commit server cluster that is using shared depotdata storage.

If this is an unplanned failover, shut down replica instances and start at step 2.7. You will have to reset the Edge servers as well by stopping them, removing the state and journal files and starting them back up after you bring the new master on-line. You also won't be able to do step 2.9 until the original master is working again.

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

Add

```
list user * * -//...
```

to bottom of the table, but make sure to leave the p4admin and service user entries below that line so they are not blocked like this:

```
list user * * -//...
```

```
super user service * //...
```

```
super user p4admin * //...
```

2.3 Rotate the journal

Rotate the journal on the master server instance(s) in using:
`/p4/common/bin/p4master_run <instance> p4 admin journal`

2.4 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.

2.5 Shutdown the replica and master instance(s)

Once you have verified the replication is up to date:

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

2.6 Copy the master journal

On the master server, for each instance, run:

```
cp /p4/<instance>/logs/journal /depotdata/p4/<instance>/journal
```

On the replica server, for each instance run:

```
mv /depotdata/p4/<instance>/journal /p4/<instance>/logs/journal
```

(The reason for copying the journals over from the original master is to keep them byte identical, and avoid having to reset all of the Edge servers.)

2.7 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/rdb.lbr
```

Run `p4 protect` and remove the lines you added to block access.

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

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

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

It is a good idea to use `crontab -l > /depotdata/p4/p4.crontab` on the master and `crontab -l > /depotdata/p4/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.

2.9 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.

```
rm /p4/1/logs/journal
```

Delete the following files if they exist: `/p4/1/root/state` or `/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`.

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_shared_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_shared_replica.sh
```