

HA(高可用性)環境構築

説明

Perforce Helix Coreのフェイルオーバー機能を使用したHA環境の構築、および構築後の構築後のテスト運用手順を説明しています



- ・ マスターサーバとスタンバイサーバのタイムゾーンが一致している必要があります。「[タイムゾーンの設定](#)」をご覧ください
- ・ p4dのバージョンが一致している必要があります。

PDF版マニュアルダウンロード

[HA構築マニュアル.pdf](#)

本構築作業の条件は以下の通りです

	マスター	スタンバイ	Perforce super user
サーバ名	master	standby	super
ServerID	master	standby	レプリケーション用ユーザ名
Option	-	mandatory ※nomandatoryで構築後に変更	service
server-service	commit-server	standby	
IP	192.168.56.114	192.168.56.115	
Port	1666	1666	
P4ROOT	/master	/standby	

構築手順

1. スタンバイサーバのサーバ仕様作成【マスターサーバで作業します】

- ① p4コマンドで"standby"というスタンバイサーバのサーバ仕様を作成します

実行コマンド

```
p4 -u super -p master-server:port server standby-serverID  
例 : p4 -u super -p 192.168.56.114:1666 server standby
```

- ② 設定を修正・追加します

赤字部分を修正・追記

```
ServerID: standby
Type: server
Services: standby ●standard(default)を上書き
Options: nomandatory ●変更しない※新規サーバはmandatoryにできません。後で変更します
ReplicatingFrom: master ●一行追記する(マスターサーバのserver ID)
Description:
Created by super.
DistributedConfig:
any#P4JOURNAL=../journals/journal
any#P4LOG=../logs/log

any#unicode=1
P4LOG=log
P4TARGET=192.168.56.114:1666
P4TICKETS=/standby/.p4tickets
db.replication=readonly
lbr.replication=readonly
monitor=2
rpl.journalcopy.location=1
serviceUser=service
startup.1=journalcopy -i 1
startup.2=pull -L -i 1
startup.3=pull -u -i 1
```



解説

startup.1=journalcopy -i 1 : 起動時にjournalcopy(マスターサーバのjournalをスタンバイに移送)を実行
startup.2=pull -L -i 1 : 移送したローカルのjournalから情報を読み取る
startup.3=pull -u -i 1 : アーカイブファイルを取得する
rpl.journalcopy.location=1 : journalcopyスレッドの書き込みを設定する構成可能変数
※より詳細な情報は『[Helix Core P4コマンドリファレンス \(2019.1\)](#)』をご参照ください

2. サービスユーザの作成と設定【マスターサーバで作業します】

① p4コマンドで"service"というレプリケーション用サービスユーザを作成します

実行コマンド

```
p4 -u super -p master-server:port user -f service-user
例 : p4 -u super -p 192.168.56.114:1666 user -f service
```

② 設定を追加します

赤字部分を追記

```
User: service
Type: service ●一行追記する
Email: service@master
FullName: service
```

③ サービスユーザ "service" のパスワードを設定します

実行コマンド

```
p4 -u super -p master-server:port passwd service-user
例 : p4 -u super -p 192.168.56.114:1666 passwd service
```

パスワード設定

```
Enter new password: (任意のパスワードを入力します)
Re-enter new password: (確認のため同じパスワードを入力します)
```

- ④ サービスユーザ "service" のアクセスレベルの設定を追加します

実行コマンド

```
p4 protect
```

表示例：赤文字部分を追記

```
write user ** //...
list user ** -//spec/...
super user super * //...
super user service * //... ●一行追加する
```

3. チェックポイントの作成と移動【マスターサーバで作業します】

- ① スタンバイサーバ用のチェックポイントファイルを作成します
作成したチェックポイントファイル名は "checkpoint" としています
チェックポイントファイル(ファイル名: checkpoint.ckp.1)が作成されます

実行コマンド

```
p4 -u super -p master-server:port admin checkpoint filename
例：p4 -u super -p 192.168.56.114:1666 admin checkpoint checkpoint
```

- ② チェックポイントファイルを転送します

FTPツール(任意)を使用して、スタンバイサーバへチェックポイントファイルを転送します。
※ここではスタンバイサーバの /tmp に格納しています

4. スタンバイサーバの設定【スタンバイサーバでの作業です】

 ①～④の作業については、サーバ構築方法によって対応が異なります。実環境をご確認の上、必要に応じて作業してください。

- ① P4ROOT用のディレクトリ "standby" を作成します
※既にルートフォルダを作成済みであれば、作業不要です

実行コマンド例

```
mkdir /$P4ROOT
例：mkdir /standby
```

- ② 作成したディレクトリのownerを"root"から"perforce"に変更します
※ownerがperforceであれば作業不要です

実行コマンド

```
chown perforce:perforce -R /$P4ROOT/
例：chown perforce:perforce -R /standby/
```

- ③ FTP転送したチェックポイントファイル(checkpoint.ckp.1)をSP4ROOT(ここでは /standby)へ移動します
※SP4ROOTへ転送済みであれば作業不要です

実行コマンド

```
mv /転送場所/filename /$P4ROOT/
例：mv /tmp/checkpoint.ckp.1 /standby/
```

- ④ チェックポイントファイルのownerを"root"から"perforce"に変更します
※ownerがperforceであれば、作業不要です

実行コマンド

```
chown perforce:perforce filename  
例 : chown perforce:perforce checkpoint.ckp.1
```

⑤ Perforceサービスを停止します

実行コマンド

```
p4dctl stop server-name  
例 : p4dctl stop standby
```

⑥ 不要なファイルを削除します 構築したスタンバイサーバに存在するデータベース等を削除します

\$P4ROOTに移動し、以下のファイルを削除します

- ・ファイル名が[db.]で始まるファイル(データベース)すべて
- ・server.locksフォルダ：フォルダごと削除

実行コマンド例

```
rm -f db.*  
rm -fr server.locks/
```

⑦ ServerIDを指定します サーバ"standby"のserverIDを"standby"に指定します ※既に指定されている場合は作業不要です

実行コマンド

```
p4d -r /$P4ROOT -xD standby-serverID  
例 : p4d -r /standby -xD standby
```

⑧ チェックポイントファイルを展開します。※ownerが"perforce"になるように展開してください。 コマンド実行後、db.ファイルが作成されたことを確認します

実行コマンド

```
p4d -r /$P4ROOT/ -jr filename  
または  
sudo -u perforce p4d -r /$P4ROOT -jr filename
```

⑨ Perforceサービスを開始します

実行コマンド

```
p4dctl start server-name  
例 : p4dctl start standby
```

5. レプリケーションの設定と確認【スタンバイサーバでの作業です】

① ユーザ"service"が使用するレプリケーション用チケットを作成します ※"/standby/.p4tickets"はサーバ仕様作成時(手順1-②)に指定した場所を入力します。

実行コマンド

```
p4 -p master_server:port -u service -E "P4TICKETS=/standby/.p4tickets" login  
例 : p4 -p 192.168.56.114:1666 -u service -E "P4TICKETS=/standby/.p4tickets" login
```

② レプリケーション用チケットの確認をします

実行コマンド

```
ls -la /standby/.p4tickets
```

実行結果：例

```
-r-----. 1 performe performe 53 10月 10 16:48 /standby/.p4tickets
```

③ レプリケーションの確認をします

実行コマンド

```
p4 -u super -p standby_server:port pull -lj  
例：p4 -u super -p 192.168.56.115:1666 pull -lj
```

表示されるメッセージ例

```
Current replica journal state is: Journal 1, Sequence 8403.  
Current master journal state is: Journal 1, Sequence 8403.  
The statefile was last modified at: 2019/10/18 14:01:11.  
The replica server time is currently: 2019/10/18 14:01:42 +0900 JST
```

※大規模なサイトの場合、同期に時間がかかる場合がありますので、時間を置いて再度コマンドを実行してください。
"Sequence"が同じ数値になれば同期完了です。
※エラーが表示される場合は、作業を停止し、弊社サポートまでご連絡ください。

解説

pull -ljコマンド：レプリカとマスターにおけるカレントジャーナルの状態を表示するコマンドです
以下の内容が含まれます
・カレントジャーナルの状態 ・ステートファイルの最終更新時刻
・サーバのローカル時刻 ・タイムゾーン

6. mandatoryへ変更する【マスターサーバで作業します】

① マスターサーバとスタンバイサーバのジャーナルの整合性を確認します

実行コマンド

```
p4 servers -j
```

実行結果

```
master '2019/10/18 14:07:44' commit-server 1/8773 1/8773 wadL/1 1  
standby '2019/10/18 14:07:43' standby 1/8575 1/8575 wAdl/4 1
```

解説

p4 servers -j：ジャーナルコピーの進行状況を確認するコマンドです。
マスターサーバとスタンバイサーバのジャーナルの数値(緑文字)が異なる場合はまだ進行中であり、一致していないこととなります。再度実行し、マスターサーバとスタンバイサーバが一致することを確認します。

② serverID "standby" の設定を表示します。

実行コマンド

```
p4 -u super -p master_server:port server standby-serverID  
例：p4 -u super -p 192.168.56.114:1666 server standby
```

③ serverID "standby" のOptionsを変更します

赤文字が変更箇所です

Services: standby
Options: **mandatory** ●nomandatoryからの書き換え
ReplicatingFrom: master

④ mandatoryに変更されたことを確認します

実行コマンド

p4 servers -J

実行結果：※末尾に「mandatory」と表示されます

master '2019/10/18 14:09:47' commit-server 1/9179 1/9179 wadL/1 1
standby '2019/10/18 14:10:40' standby 1/9179 1/9179 wAdl/4 1 **mandatory**

■ フェイルオーバー：テスト運用一マスターサーバが起動している場合【スタンバイサーバで作業します】

マスターサーバが起動できており、ネットワークに接続できている状況を想定した運用手順です。

① フェイルオーバーの確認

実行コマンド

p4 failover

表示されるメッセージ

Checking if failover might be possible ...
Checking for archive file content not transferred ...
Verifying content of recently updated archive files ...
After addressing any reported issues that might prevent failover, use --yes or -y to execute the failover.
※異なるメッセージ・エラーが表示される場合は、テストを中止し、弊社サポートまでご連絡ください。

② フェイルオーバーの実行

自動的にPerforceサーバが再起動され、マスターサーバへと切り替わります。

実行コマンド

p4 failover -y

表示されるメッセージ

Starting failover process ...
Refusing new commands on server from which failover is occurring ...
Giving commands already running time to complete ...
Stalling commands on server from which failover is occurring ...
Waiting for 'journalcopy' to complete its work ...
Waiting for 'pull -L' to complete its work ...
Waiting for 'pull -u' to complete its work ...
Checking for archive file content not transferred ...
Verifying content of recently updated archive files ...
Stopping server from which failover is occurring ...
Moving latest journalcopy'd journal into place as the active journal ...
Updating configuration of the failed-over server ...
Restarting this server ...
Perforce Server re-starting...

■ フェイルオーバー：テスト運用—マスターサーバが起動していない場合【スタンバイサーバで作業します】

マスターサーバが起動できない・ネットワークに接続できない状況を想定した運用手順です。

① フェイルオーバーの確認

実行コマンド

```
p4 failover -s master-server  
例：p4 failover -s master
```

表示されるメッセージ

Checking if failover might be possible ...
Checking for archive file content not transferred ...
Verifying content of recently updated archive files ...
After addressing any reported issues that might prevent failover, use --yes or -y to execute the failover.
※異なるメッセージ・エラーが表示される場合は、テストを中止し、弊社サポートまでご連絡ください。

解説

-s serverID……切り替えるサーバ（今回はマスターサーバ）のserverIDを指定します
"-s"オプションが無い場合は以下のメッセージが表示されます
Server ID must be specified in the '-s' or '--serverid' argument for a failover without the participation of the server from which failover is occurring.
Empty identifier not allowed.

② フェイルオーバーの実行

自動的にPerforceサーバが再起動され、マスターサーバへと切り替わります。

実行コマンド

```
p4 failover -s master-server -y  
例：p4 failover -s master -y
```

表示されるメッセージ

```
Starting failover process ...
Refusing new commands on server from which failover is occurring ...
Giving commands already running time to complete ...
Stalling commands on server from which failover is occurring ...
Waiting for 'journalcopy' to complete its work ...
Waiting for 'pull -L' to complete its work ...
Waiting for 'pull -u' to complete its work ...
Checking for archive file content not transferred ...
Verifying content of recently updated archive files ...
Stopping server from which failover is occurring ...
Moving latest journalcopy'd journal into place as the active journal ...
Updating configuration of the failed-over server ...
Restarting this server ...
Perforce Server re-starting...
```

■ フェイルオーバー：テスト運用一参照サーバの変更

この項目では、フェイルオーバーを実行後、旧マスターサーバ(ここではmaster)を参照していたサーバでの対応についてのご案内です。ここではエッジサーバを例にしています。
※マスターサーバとスタンバイサーバに対するDNSのエイリアス設定がされている場合は対応は不要です。

① 状況を確認する【エッジサーバで作業します】

実行コマンド

```
p4 pull -lj
```

i このメッセージが表示される場合は、DNSのエイリアス設定がされていません

```
Replica access to P4TARGET server failed.
Remote server refused request. Please verify that service user is correctly logged in to remote server, then retry.
TCP connect to master-server:port failed.
connect: master-server:port: Connection refused
```

② “P4TARGET” を修正します

実行コマンド

```
sudo -u perforce p4d -r /edge1 "-cset edge1#P4TARGET=新master-server:port"
例 : sudo -u perforce p4d -r /edge1 "-cset edge1#P4TARGET=192.168.56.115:18800"
```

③ 変更を確認します

実行コマンド

```
sudo -u perforce p4d -r /edge1 "-cshow"
```

実行結果

```
edge1: P4TARGET = 新master-server:port
```

④ 同期の確認

実行コマンド

```
p4 pull -lj
```

コマンド結果

Current replica journal state is: Journal 4, Sequence 6065.
Current master journal state is: Journal 4, Sequence 6065.
The statefile was last modified at: 2019/10/21 02:38:44.
The replica server time is currently: 2019/10/21 02:38:51 +0900 JST
"Sequence"が同じ値になれば同期完了です。

構築、テストは以上です。