

Perforce Helix Core on AWS

Primary Server Advanced Configuration Guide

Zenta Hori – Solutions Architect, AWS Game Tech Japan

Kevin Ashman – Senior Partner Solution Architect, Game Tech

December 2020



Introduction

This guide steps you through testing, configuring, and connecting to a deployed Perforce Helix Core server on AWS following the steps from the [AWS Perforce GitHub Sample](#) and the Perforce on [AWS Getting Started Blog](#).

Advanced Configuration of Primary Perforce Server

1. First, we need to log in to the server we built via SSH. A login user named **p4admin** was created, which we will use. The standard `ec2-user` user still exists if needed. Open a terminal and run the following command.
 - Replace `[your-ssh-key.pem]` with your ssh key and `[54.248.42.141]` with your EIP as shown in **PerforceMasterEIP** from your CloudFormation Stack Outputs tab.

```
$ ssh -i ~/.ssh/[your-ssh-key.pem] p4admin@54.248.42.141
```

2. Once you successfully logged in to the server, you can run the Perforce client using `p4`. If you see the following response from Perforce after running `p4 -p localhost:1666 info`, it indicates that the server is built successfully.

```
$ p4 -p localhost:1666 info
User name: p4admin
Client name: ip-10-0-0-63
Client host: ip-10-0-0-63.ap-northeast-1.compute.internal
Client unknown.
Current directory: /home/p4admin
Peer address: 127.0.0.1:38606
Client address: 127.0.0.1
Server date: 2020/04/23 10:19:02 +0000 UTC
Server version: P4D/LINUX26X86_64/2019.2/1942501 (2020/04/02)
ServerID: master.1
Server services: standard
Server license: none
Case Handling: insensitive
```

3. If you are use Unreal Engine 4 (UE4) for development, you will want to configure the following setting (optional). Run the following command for typemap configuration.

- More information can be found [here](#) in the Perforce Documentation.

```
$ p4 -p localhost:1666 -u perforce typemap
```

4. The command above starts vi editor, so modify the text as follows.
 - Note: Use command “i” to enter text insertion mode and edit text. Copy and paste the following texts to the location starting with “TypeMap:”. After that, press [esc] to return to command mode and save with “:wq”.

```
# Add the following texts to the content of the file opened with vi.
```

```
TypeMap:
```

```
binary+w //depot/....exe
binary+w //depot/....dll
binary+w //depot/....lib
binary+w //depot/....app
binary+w //depot/....dylib
binary+w //depot/....stub
binary+w //depot/....ipa
binary //depot/....bmp
text //depot/....ini
text //depot/....config
text //depot/....cpp
text //depot/....h
text //depot/....c
text //depot/....cs
text //depot/....m
text //depot/....mm
text //depot/....py
binary+l //depot/....uasset
binary+l //depot/....umap
binary+l //depot/....upk
binary+l //depot/....udk
binary+l //depot/....ubulk
```

Testing Primary Server

Now that we succeed in connection to our primary server via SSH, we can test it by committing a file to Perforce from the terminal.

1. First set an environment variable to avoid specifying the same address and the same port number every time.

```
$ echo 'export P4PORT=localhost:1666' >> ~/.bash_profile
$ source ~/.bash_profile
```

2. Create a p4admin user on the primary server. When vi editor launches, confirm the content and type “:wq” to save it.

```
$ p4 user p4admin
User p4admin saved.
```

3. Create a depot on the primary server. You can name the depot as you like. In the below example we will name it “repository”. When vi editor launches, confirm the content and type “:wq” to save it.

```
$ p4 depot repository
Depot repository saved.
```

4. Next, we will create a workspace for client. Change the current directory to **/home/p4admin** and run the following command to launch vi editor and display the content of the config. Check **Root** and **View** items. Make sure that **Root** is **/home/p4admin** as below. If it's not, correct it as follows. Also ensure that **View** item has at least “**///repository/... //p4admin/repository/...**” as follows. There is no problem with it having additional items you might find. Press [esc] to return to command mode. Enter “:wq” and exit vi unless you see any problem. This creates a workspace for the client in Depot.

```
$ cd /home/p4admin
$ p4 client p4admin
```

```
Client p4admin saved.
```

```
# Confirm the following configuration in the content of the file opened
with vi.
# If this is not set correctly, p4 commands and commits will not work
properly.
Root: /home/p4admin
View:
  //spec/... //p4admin/spec/...
  //repository/... //p4admin/repository/...
  //depot/... //p4admin/depot/...
```

5. Now, we will work at `/home/p4admin/repository`. Create a file named `~/repository/test.config` and test committing the file in the server.

```
$ mkdir -p ~/repository
$ cd repository/
$ echo test > test.config
```

6. Add the file to a changelist with the following command.

```
$ p4 -c p4admin add -t text test.config
//repository/test.config#1 - opened for add
```

7. Submit the change to the depot with the following command: The command will open vi editor so you can describe the reason for the change in the “**Description:**” section. If it shows change submitted, you finished testing the commit process.

```
$ p4 -c p4admin submit
Change 1 created with 1 open file(s).
Submitting change 1.
Locking 1 files ...
add //repository/test.config#1
Change 1 submitted.
```

```
# When vi launches, describe the change. Save the change with “:wq” to
close vi.
# Describe the change
Description:
  add new file
Files:
  //repository/test.config # add
```

[Optional] Installing P4 client tool on your local Mac or Windows PC

1. Please follow the instruction on how to install P4 client tool corresponding to your OS in [this](#) page.

2. Let's verify that you can connect to servers by running p4 command from your local PC. Replace the IP address below with your primary server's IP address.

- When you access the Perforce Server from your office, please ensure that your office network allows communication via Port:1666.
- If you do not use VPN or AWS Direct Connect for the connection to the AWS cloud, the connection is established over the internet and the communication is not encrypted. Please ensure that committed data don't include any confidential information in files.

```
$ p4 -p 54.248.42.141:1666 info
User name: john
Client name: xxxxxxxxxx
Client host: xxxxxxxxxx.sample.com
Client unknown.
Current directory: /Users/john/work
Peer address: xxx.xxx.xx.xx:63559
Client address: xxx.xxx.xx.xx
Server date: 2020/03/03 08:47:02 +0000 UTC
Server version: P4D/LINUX26X86_64/2019.2/1918134 (2020/02/12)
ServerID: master.1
```

```
Server services: standard
Server license: none
Case Handling: insensitive
```

Summary

At this point you have verified the Perforce server setup and added the admin account. You have also tested the server by locally committing a file and optionally connecting from your local machine. You can now proceed with either testing this server further or using it for production.