Foreword

Greetings, and welcome to the ultimate Perforce Starter's Guide. This guide has three main sections, presented in the following order:

1. Setting Up a Perforce Client
2. Getting Unity Hooked Up to Perforce
3. Setting Up a Perforce Server with AWS

Note: this guide is tailor-made for those who are both a) on a Windows machine (specifically Windows 10) and b) using Unity for their project (specifically a version from 2020). I have not been able to try it out on another device or in an engine other than Unity 2020.

Note: Anytime the instructions state to type something, ignore the start and end quotation marks. (e.g. type "scringy bingo" → you type just scringy bingo)

Anywho, without further ado, let's get started on setting some stuff up!

Setting Up a Perforce Client

This section is going to guide you through all the necessary steps it takes to set up a Perforce client on your machine.

If you are responsible for making the server in your development team, skip this section and go to "Getting Unity Hooked Up to Perforce" after getting it set up.

Necessary Setup

1. Download Helix Visual Client (P4V)
    a. https://www.perforce.com/downloads/helix-visual-client-p4v
    b. Download the client for whatever family and platform your machine currently runs.
    c. Run the installer.
    d. Keep all of the applications selected. Note the install directory, as that is where your Unity collaborations will eventually go. Click next.
    e. Leave all fields as default. Perforce will whine that you didn't change the server. Click OK and click next again; you'll be able to change the server later once you get the address from the person who set up the server – **from here on out, they will be referred to as "your admin".**
    f. Click Install. Once installed, exit out of the installer.

Creating a Perforce User Account

1. Go to your windows search bar and type "p4v" This should bring up the Perforce client. Open it.
2. It will ask you for the server and user. Paste the address you get from your admin and add ":1666" to the end of the address if it's not there already.
3. Create a new user by clicking "New" next to the user field.
    a. Specify an easy to remember username, your name, and email address. A password is not necessarily, and Perforce won't try to confirm your email.

4. Set up a new workspace by clicking the “New” button beside the field.
    a. Perforce will attempt to create a new folder inside the Perforce directory on your machine. This is where all Unity projects you create and all projects you get from Perforce will be stored.
    b. Name the Workspace something easy to remember, like “Workspace”
    c. Leave everything else as default and click “OK”
5. Click “OK” and a new window should pop up. This window is the repository that will contain all project files.
6. **Congratulations!** Your user account is all set up and ready to go!

Getting Unity Hooked Up to Perforce

This section is going to guide you through all the necessary steps it takes to get Unity working with Perforce. All the work it takes to set up a Perforce user and server means nothing if you can’t get it working with Unity!

Necessary Setup

1. Make sure you have Unity installed, preferably a 2020 version.
2. That’s it.

Hooking Them Up

1. Create a Unity project in your “Workspace” folder which you set up. It should be located inside of your Perforce folder.
2. Go to Edit → Project Settings
    a. In the window that opens up, click on the “Version Control” tab.
    b. In the new screen, select “Perforce” from the dropdown menu.
    c. Type in the following:
        i. Your username you made for your Perforce user account.
        ii. Your password (if you made one for the account)
        iii. The name of your workspace (no need to specify a file path, just type the plain name of your workspace)
        iv. The address of the server, ending with “:1666”
    d. Click “Connect”. The message beneath “Log Level” will update to show that you’re connected or if something failed.
3. If you connected successfully, Perforce has successfully integrated with Unity.
    a. You will see a variety of icons next to files inside of Unity now. This is a great sign! You can see a key of these symbols in the Version Control tab.

Setting Up .p4ignore

1. **This is a very important step!** It will prevent you from uploading redundant files to your Perforce server, saving you precious space.
2. Following this document is a separate document named “.p4ignore”. Be sure to download this.
    a. Place it inside of your project directory in, which should be: C:\Users\<User>\Perforce\<Workspace Name>
    b. The provided .p4ignore file comes courtesy of the TA from my Game Programming class, Sloan. Thanks Sloan!

3. Open your Command Prompt
    a. Navigate to your project directory using the “cd” command
    b. Once there, type “p4 set P4IGNORE=.p4ignore” and press ENTER
    c. This sets the ignore file.
    d. If you get an error, most likely the issue is you decided not to download the Perforce command terminal. Run the Perforce client installer again and select that option, which should fix the issue.
    e. You can verify this worked by typing “p4 ignores”. If you see a couple lines that say “p4root”, you’re golden!
4. **Congratulations!** Your Unity is all set up with Perforce and ready to go!

Setting Up a Perforce Server with AWS

This section is going to guide you through all the necessary steps it takes to set up the Perforce server. For this, we’ll be using Amazon Web Services, which is a nice, free hosting option for projects under 30 gigabytes. **This is by far the longest process out of the three, so strap in!**

Only one person in your development team will need to take these steps to setup a server, as this is where all the development files will be stored for every team member to access. If you just need to set up a Perforce client, scroll down to “Setting Up a Perforce Client”

All download links and other necessary addresses are provided in their relevant sections.

Necessary Setup

2. Download PuTTY
    a. https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
    b. It’s used to help interface with the AWS server.
    c. Download the correct installer for your system and run it.
    d. Leave all settings as default.
    e. Ensure that PuTTY is installed to this file path:

        C:\Program Files\PuTTY

    f. Copy that file path, as you’ll need it soon.
3. Setup PuTTY Environment Variable
    a. Since PuTTY will be used via command line, we need to set it up in Windows path.
    b. Type “edit the system environment variables” in Windows search bar and click the result of the same name.
    c. Click “Environment Variables”
    d. Click the Path variable and click edit
    e. In the new screen, find this variable:

        %USERPROFILE%\AppData\Local\Microsoft\WindowsApp

    f. Select that variable and click edit.

g. Add a semicolon to the end, then paste the file path for PuTTY you got previously to the end of it.
h. Press the ENTER key and click OK until you exit all Environment Variable screens.
i. Open your Command Prompt and type “putty”. If a PuTTY window appears, then you’re all set.

4. Download 7-Zip of Other Archiver
    a. https://www.7-zip.org/download.html
    b. You will need an archiver program to access one of the files you’re about to download. 7-Zip is a free and easy-to-use option.
5. Download Helix Core (P4D) Server
    a. https://www.perforce.com/downloads/helix-core-p4d
    b. Select the following options:
        i. Family: Linux
        ii. Platform: Linux (x64)
    c. Regardless of your system, you need to choose these options since the AWS server we’ll be using runs on Linux.
    d. Click download. You may need to enter some information to get the download, though no money is required. You should get a file called:

        helix-core-server.tgz

    e. Place this file in a folder someplace, as this next step will generate quite a few files.
    f. If you have 7-Zip:
        i. Right-click **helix-core-server.tgz** →7-zip → Extract Here
        ii. You’ll get a new file called **helix-core-server.tar**
        iii. Right-click **helix-core-server.tar** → 7-zip → Extract Here
        iv. A whole mess of files should pop out. Delete all files, including the .tgz and .tar files **EXCEPT** the **p4d** file.
    g. If you have a different archiver, the principle is the same. Extract until you get the **p4d** file.
    h. Place this **p4d** file in a folder that’s easily accessible.
6. Download Helix Visual Client (P4V)
    a. https://www.perforce.com/downloads/helix-visual-client-p4v
    b. Download the client for whatever family and platform your machine currently runs.
    c. Run the installer.
    d. Keep all of the applications selected. Note the install directory, as that is where your Unity collaborations will eventually go. Click next.
    e. Leave all fields as default. Perforce will whine that you didn’t change the server. Click OK and click next again; you’ll be able to change the server later once it’s set up.
    f. Click Install. Once installed, exit out of the installer.
7. Set Up an Amazon Web Services (AWS) Account
    a. https://aws.amazon.com/
    b. If you don’t already have an account, go there, and set up a new one.
    c. You will need:

i. An email address.
ii. A mobile phone number.
iii. A physical address.
iv. A valid credit card. - ***Don't worry about payments!***

d. Even though you'll be selecting the free Basic Plan, Amazon still requires a card on file in case you start branching into other services.
e. Once you follow all registration steps, the final step in the registration process is to select the free "Basic Plan".
f. It may take a little over 24 hours to set up your account, though it was fairly quick in my experience.

Setting Up the AWS Server

1. Sign Into AWS
    a. Go to the search bar and type "ec2"
    b. Click "EC2: Virtual Servers in the Cloud"
2. Launching an EC2 Instance
    a. Click the "Launch Instance" button and click the "Launch instance" option that drops down from it.
    b. **Step 1:** Select "Amazon Linux 2 AMI (HVM), SSD Volume Type"
        i. Leave the default 64-bit (x86) option on the left selected.
        ii. Click the "Select" button.
    c. **Step 2:** The default "t2 micro" type should already be selected. If not, select it and click "Next: Configure Instance Details"
    d. **Step 3:** Skip, all default values are good. Click "Next: Add Storage"
    e. **Step 4:** Under "Size (GiB)" set the field to 30. This is the maximum amount of space allowed by the free tier. Click "Next: Add Tags"
    f. **Step 5:** Skip, no tags needed. Click "Next: Configure Security Group"
    g. **Step 6:** This is probably the most involved step. We'll go field-by-field.
        i. Click "Add Rule"
        ii. Type: "Custom TCP Rule"
        iii. Port Range: 1666
        iv. Source: Anywhere
        v. At the top, change the "security group name" and "description" to something easy to remember.
        vi. Click "Review and Launch"
    h. **Step 7:** AWS will lecture you on cybersecurity, saying that your instance could be broken into easier than a new pair of sandals.
        i. Ignore it, what does Amazon know about security?
        ii. Really though, it's fine. It'd be much more of a hassle to specify all IP addresses that are allowed to access it and you're not going to be storing nuclear launch codes on this server.
        iii. Click "Launch"
    i. A window titled "Select an existing key pair…" should pop up
        i. Select "create a new key pair" in the first drop down
        ii. Choose a good name (without spaces!) for the key pair name. **Remember this name!**
        iii. Click "Download Key Pair" and you should get a file called

[key name].pem

iv. Move this .pem file to the same directory as the p4d file you extracted earlier.
v. Click “Launch Instances”

3. Launch Status
    a. You should see a new screen saying that your instance is launching and a tip talking about charge notifications.
        i. It may be a good idea to set your email preferences now so that you get notified of any usage that may result in a charge.
    b. Click “View Instances”
    c. It may take a hot minute for your new instance to be fully set up. Once the “Instance State” says “running”, you’re good to go.
    d. Make sure to leave this screen open in your browser! You’re going to need it again soon!

Key Setup

1. **Only proceed once the Instance State changes to “running” on AWS!**
2. Open your Command Prompt and type “puttygen”
    a. The “PuTTY Key Generator” screen should open.
    b. Click “Load” and navigate to the folder containing the .pem and p4d files.
    c. Change the file type from .ppk to “All Files” in the file explorer window that opens.
    d. Select the .pem file you downloaded and click “Open”
    e. Click OK on the PuTTYgen Notice; we know it’s a foreign key.
    f. Make sure that the type of key is set to “RSA”, then click “save private key” and click “Yes”
    g. Save it in the same location as the .pem and p4d files and give it the same name as the .pem file.
3. Close the PuTTY Key Generator

EC2 Server Setup

1. Open your Command Prompt
    a. Navigate with the command “cd” to the file path where you saved the .ppk and p4d files
    b. Example: “cd Desktop/Setup” (I saved the two files in a folder called “Setup” on my Desktop)
    c. Press enter, and your console is now operating inside that directory.
2. Open up the list of your AWS instances in your browser again
    a. Select the instance you just created. **If it’s not running by now, then you ignored my previous warning and I’ll be just a bit miffed.**
    b. Select the instance and click the “Connect” button.
    c. A new screen should appear showing a section called “SSH client”
3. Set Up with PuTTY
    a. Copy the example string at the very bottom of the new screen in your browser. You can do this quickly by clicking the new little squares.

b. Go back to your Command Prompt and hit CTRL+V to paste. **But don't you dare hit ENTER.** We ain't done yet.
c. There are a few things about the pasted line we have to change. They are:
    i. Change "ssh" to "putty"
    ii. Change ".pem" to ".ppk" in the quotes of the string.
    iii. The file name in the quotes of the string should be the same as the one you gave the provided and generated security key (the .pem and .ppk files)
    iv. Press ENTER.
d. Your final string should look something like this, though I censored the address. Note my key name in quotes may be different from yours.

e. PuTTY's gonna show up again and immediately complain.
    i. click "yes" on the security warning; we trust the host we're connecting to (after all, we just made it)
    ii. Remember the username you use to connect
        1. (it should be "ec2-user")
    iii. Close the window and click OK on the dialog box that appears.

4. Uploading the p4d File
    a. Go back to your Command Prompt and hit CTLR+V again to paste the string you copied previously. **I swear, if you press ENTER before I tell you to…**
    b. We gotta edit it again! Here's the changes you gotta make:
        i. Change "ssh" to "pscp"
        ii. Between "pscp" and the "-i" type "-P 22" to specify the port.
        iii. Change ".pem" to ".ppk" in the quotes of the string once again.
        iv. Between the quotes and address, type "p4d"
        v. Add a colon (:) at the very end of the line.
        vi. Press ENTER.
    c. Your final string should look something like this, though I censored the address. Note my key name in quotes may be different from yours.

    d. **If you get an error**, namely with failing to assign a requested address, try removing the "22 -i", though this is meant to fix that issue.
    e. It may freeze for a second, and then show progress being made on the file transfer. Once it completes, the p4d file should be on your AWS instance.
5. Setting Up the Perforce Server on Your AWS Instance
    a. Still in the Command Prompt, press the up arrow on your keyboard until you see the line starting with "putty" again. Press ENTER.
    b. The ec2-user window will appear again. If it requires you to specify a user, type "ec2-user"
    c. Type the following commands in order:

i. “ls” (that’s a lowercase L not an uppercase I) This will show all files in the current directory. If you only see “p4d” after pressing enter you’re good to go.
ii. Make a new directory called “perforce” by typing “mkdir perforce”
iii. Type “ls” again to make sure the new directory was made.
iv. Move the p4d file into the new directory by typing “mv p4d perforce/p4d”
v. Move into the perforce directory by typing “cd perforce/”
vi. Inside the directory, type “chmod +x p4d” to turn the p4d file into an executable.
vii. Finally, type “./p4d”. If you see “Perforce Server starting…” you’re sitting pretty! You just got a Perforce server all set up on AWS.

d. Close out of the ec2-user window.

P4Admin Setup

1. Type “p4admin” in your Windows search bar and click on the “P4Admin” program.
    a. You should see a screen with fields for a server address and user.
2. Go back to your EC2 instance on AWS and navigate back to the “Connect to instance” screen by selecting your instance and clicking the “Connect” button.
    a. Copy the field under the 4th point, the “Public DNS”
    b. This is the address you and any collaborators will use to access your Perforce server via the Perforce client
3. Go back to P4Admin
    a. Paste the copied DNS Address in the server field.
    b. Add “:1666” at the end of the address.
    c. Create a new user, giving it an easy-to-remember username.
    d. Fill out every field except the password, as it’s not needed. Once again, no real need for security for these projects. Perforce won’t send you an email verification for this, either.
    e. Once all of this is set up, click “OK” and then two new windows should pop up.
    f. Click “Yes” on the Helix Admin window; it’s letting you know you’re the admin of the server.

Final Stretch – Perforce Client Repository Setup

1. Go to your windows search bar and type “p4v” This should bring up the Perforce client. Open it.
2. It will ask you for the server and user. Paste the Public DNS address in the server field again, ensuring the address ends with “:1666”
3. Your user should already be specified, but if not, browse for the one you made in the previous section.
4. Set up a new workspace by clicking the “New” button beside the field.
    a. Perforce will attempt to create a new folder inside the Perforce directory on your machine. This is where all Unity projects you create and all projects you get from Perforce will be stored.
    b. Name the Workspace something easy to remember, like “Workspace”

    c. Leave everything else as default and click “OK”
5. Click “OK” and a new window should pop up that will get you started on setting up the repository.
    a. You should be given an option between choosing “streams” or “classic”. I choose classic because it’s simpler. Steams seem more fit for larger teams, though you may want to do your own research in this regard and pick what you think is best for you.
    b. Pick some folder on your system that you want to put on the Perforce. You can delete it later if it’s not your Unity project.
    c. Click “OK” to everything to get your dummy folder uploaded to the repository. Once it’s done, you’ll have access to the repository.
    d. There should be two other windows asking if you want to keep Perforce updated and if you’re okay with them collecting your data. Answer these however you wish.
6. **Congratulations**! You just set up fully featured Perforce server! Take a breather!