Server Deployment Package (SDP) for Perforce Helix

SDP Upgrade Guide (for UNIX/Linux)

Perforce Professional Services

Version v2020.1, 2020-09-22

Table of Contents

1. Introduction . . . 1
2. Upgrade Order: SDP first, then Helix P4D . . . 2
   2.1. Upgrading P4D . . . 2
3. SDP and P4D Version Compatibility . . . 3
4. SDP Installation Methods . . . 4
5. Custom HMS Managed Installations . . . 5
6. Upgrade from SDP 2020.1+ . . . 6
7. Upgrade from pre-20.1 SDP . . . 7
   7.1. Mount Point Names . . . 7
   7.2. Operating System User . . . 7
   7.3. Upgrade Procedure for Pre-SDP 2020.1 SDP. . . . 7
      7.3.1. Preparation . . . 8
         PREP STEP 1: Acquire Downloads . . . 8
         PREP STEP 2: Deploy new SDP **commmon** files . . . 8
         PREP STEP 3: Modify new SDP **config** files . . . 8
         PREP STEP 4: Configure new SDP **instance bin** files and symlinks. . . . 8
         PREP STEP 5: Determine Your Metadata Symlink Type (Fixed or Variable) . . . 8
         PREP STEP 6: Account for any customizations. . . . 8
         PREP STEP 6: Account for customization (if any) . . . 9
      7.3.2. Execution . . . 9
         EXEC STEP 1. Stop Services . . . 9
         EXEC STEP 2. Move old SDP Aside . . . 9
         EXEC STEP 3. Upgrade Physical Structure . . . 10
         STEP 3: Replace Instance Symlink with Directory . . . 10
         STEP 4: Convert Fixed to Variable Metadata Symlinks . . . 11
         STEP 5: Replace Instance Symlink with Directory . . . 11
         EXEC STEP 4. Put new SDP **common** files in place. . . . 12
         EXEC STEP 5. Put new SDP **instance bin** files in place. . . . 12
         POSTOP STEP N: Cleanup - Deferred . . . 12

Chapter 1. Introduction

This document defines the procedure for ugprading the Perforce Helix Server Deployment Package (SDP) on UNIX/Linux systems.

Chapter 2. Upgrade Order: SDP first, then Helix P4D

Starting with the r20.1 release of SDP, the released versions of SDP match released versions of P4D. So SDP r20.1 will be guaranteed to work with P4D r20.1.

The SDP should be upgraded prior to the upgrade of Helix Core (P4D). If you are upgrading P4D to or beyond P4D 2019.1 from a prior version of P4D, you *must* upgrade the SDP first. If you run multiple instances of P4D on a given machine (potentially each running different versions of P4D), upgrade the SDP first before upgrading any of the instances.

2.1. Upgrading P4D

If P4D is already r2019.1 or newer, after the SDP upgrade, Helix Core can be upgraded using the SDP `upgrade.sh` script. Commonly, the SDP and P4D upgrades are done in a single maintenance window.

If run on a global topology, the `upgrade.sh` script must be run on outermost replica servers of the topology first (e.g. replicas of edges), and then their P4TARGET servers, and finally on the master/commit server after all replicas have been upgraded. Run `/p4/common/bin/upgrade.sh -man` to see the documentation on staging binaries prior to running `upgrade.sh` on any given machine.

If P4D is older than r2019.1, contact us for upgrade details.

Chapter 3. SDP and P4D Version Compatibility

The SDP is often forward- and backward-compatible with P4D versions, but for best results they should be kept in sync by upgrading SDP before P4D. This is partly becuase the SDP contains logic that helps upgrade P4D, which can change as P4D evolves.

The SDP is aware of the P4D version, and has backward-compatibility logic to support older versions of P4D. This is guraanteed for supported versions of P4D (generally current and two prior releases). Backward compatiblity of SDP with older versions of P4D is likely to extend farther back, though sans the "offically supported" guarantee.

Chapter 4. SDP Installation Methods

There are 4 SDP installation methods, each described in detail later.

- **In-Place, Automated**: Automated in-place upgrades can be done if your current SDP version is r20.1 or later, and you are upgrading SDP to r20.2 or later.
- **In-Place, Manual**: Manual upgrades of the SDP must be peformed to upgrade from versions older than r20.1.
- **Migration-Style**: A migration-style upgrade is one in which the existing server machines (virtual or physical) are left in place, and brand new "green field" machines are installed fresh using the Helix Installer (which installs the latest SDP on a "green field" baseline operating system). Then the Helix Core data is migrated from the existing hardware to the new hardware. This approach is especially appealing when upgrading other aspects of the infrastructure, such as the hardware and/or operating system.
- **Custom with HMS**: The Helix Management System is used by some customers.

Chapter 5. Custom HMS Managed Installations

If the Helix Management System (HMS) is used to manage this installation, you should have custom site-specific documentation for upgrading the SDP that supercedes this documentation. If the file `/p4/common/bin/hms` exists at your site, you have an HMS-managed site. Conact Perforce Consulting for more information.

Note that HMS solutions are inherently custom and not officially supported, but can be fully automated for global Helix Core topologies.

Chapter 6. Upgrade from SDP 2020.1+

If your current version of the SDP is 2020.1 or later, you will be able to upgrade the SDP on any given machine to 2020.2 or later using the `upgrade_sdp.sh` script. Starting with the SDP 2020.2 release, an `upgrade_sdp.sh` will be included in the SDP which will SDP in place on any given machine from any prior version, so long as the version is 2020.1 or later.

To confirm that your current version of the SDP is 2020.1 or later, check the contents of the file: `/p4/common/Version`

Chapter 7. Upgrade from pre-20.1 SDP

The following procedure accounts for details for upgrading from various older implementations of the SDP, going back as 2007.

7.1. Mount Point Names

You will need to be aware of your moint point names. While referred to as mount point names in this document, in any given installation any or all of the three SDP "mount points" may be simple directories on the root volume or some other volume. In some installations, creative liberties were taken to use fewer than three volumes, and in some cases the operating system root volume was used as one of the volumes. Investiage and be aware of how your installation was configured.

In the examples below, the modern SDP mount point names are used:

- `/hxdepots` - Volume for versioned files and rotated/numbered metadata journals.
- `/hxmetadata`- Volume for active and offline metadata
- `/hxlogs` - Volume for active journal and various logs.

Depending on the version of the SDP, the above values may be used, or earlier defaults such as, `/depotdata`, `/metadata`, and `logs`. However, often customer sites changed from the defaults to custom values, such as `/depots`, `/p4db`, `/p4journal`, and others.

In the sample steps in this document, adapt the steps to use your local values for mount point names to the new values.

If your site uses two volumes for metadata, `/hxmetadata1` and `/hxmetadata2`, continue using the same names below.

7.2. Operating System User

You will need to be aware of your operating system user that `p4d` runs as in your environment.

The sample steps below assume that Perforce runs as the `peforce` operating system user, which is typical. Adapt if your user is something else, such as `p4admin`.

In modern installations, the default home directory is `/home/perforce`, though in some installations the home directory is `/p4`. In either case, this does not need to be changed during the upgrade process.

7.3. Upgrade Procedure for Pre-SDP 2020.1 SDP.

The procedure is broken into 3 phases, Preparation, Execution, and PostOp. Preparation steps can be done in a non-disruptive manner on the production server ahead of the Execution. Execution steps are generally performed in a scheduled maintenance window. PostOp steps are done some time after (perhaps days or weeks after) the upgrade is complete.

7.3.1. Preparation

Preparation steps are:

1. Acquire Downloads.
2. Deploy new SDP **common** files.
3. Modify new SDP **config** files.
4. Configure new SDP **instance bin** files and symlinks.
5. Determine Metadata Symlink Type (Fixed or Variable)
6. Account for customization (if any)

PREP STEP 1: Acquire Downloads

Download the latest SDP tarball release from this link: https://swarm.workshop.perforce.com/projects/perforce-software-sdp/download/downloads/sdp.Unix.tgz.

Copy the downloaded tarball to the machine and put it in `/hxdepots/sdp.Unix.tgz`. (If a file with the same name exists from a previous upgrade, move it aside first.)

Next, get the latest `p4`, `p4d`, `p4broker`, and `p4p` executables from the Perforce FTP server: https://ftp.perforce.com/perforce/r20.1/bin.linx26x86_64, and get them copied to `/tmp`.

The following is typically performed in a scheduled maintenance window.

PREP STEP 2: Deploy new SDP commmon files

EDITME

PREP STEP 3: Modify new SDP config files

EDITME

PREP STEP 4: Configure new SDP instance bin files and symlinks.

EDITME

PREP STEP 5: Determine Your Metadata Symlink Type (Fixed or Variable)

EDITME

PREP STEP 6: Account for any customizations.

Login as the `perforce` operating system user, and run this command:

```
ls -l /p4/1/root /p4/1/offline_db
```

The `root` and `offline_db` will be symlinks.

Depending on how old the SDP is, the structure will either be *fixed* or *variable* metadata symlinks. Determine which you have.

Variable Metadata Symlink References

If one of the symlinks points to a directory ending in `db1`, and the other in `db2` (it doesn't matter which is pointing to which), you have **variable metadata symlinks**.

Fixed Metadata Symlink References

If the target of the `root` and `offline_db` symlinks points to directories ending in the same names, i.e. `root` and `offline_db`, then you have **fixed metadata symlinks.**

PREP STEP 6: Account for customization (if any)

7.3.2. Execution

Execution steps are:

1. Stop Services
2. Move Old SDP aside.
3. Upgrade Physical Structure
4. Put new SDP **common** files in place.
5. Put new SDP **instance bin** files in place.

EXEC STEP 1. Stop Services

Stop the `p4d` service for all instances on this machine. Also stop all p4broker services running on this machine (if any).

For this short maintenance, the broker cannot be left running (e.g. to broadcast a "Down For Maintenance (DFM)" message) because the structure change cannot be started until *all* processes launched from the SDP directory structure have stopped.

Sample commands:

```
p4d_1_init status
p4d_1_init stop
p4d_1_init status

p4broker_1_init status
p4broker_1_init stop
p4broker_1_init status
```

The extra `status` are for situational awareness; and are not strictly necessary.

EXEC STEP 2. Move old SDP Aside

EXEC STEP 3. Upgrade Physical Structure

In this step, the physical structure of the upgrade is done for pre-2019.1 SDP.

The structure of the SDP changed in the 2019.1 release, to increase performance and reduce complexit in post-failover operations. The following notes describe how to do an in-place conversion to the new structure.

First, become familar with the Pre-2019.1 and 2019.1+ structures.

SDP Pre-2019.1 Structure:

- `/p4` is a directory on the operating system root voume, `/`.
- `/p4/N` is a symlink to a directory is typically the mount point for a storage volume (`/hxdepots` by default).
- `/p4/N` contains symlinks for `/hxdepots`, `/hxmetadata`, and `hxlogs`, as well as tickets and trust files.

SDP 2019.1+ Structure:

- `/p4` is a directory on the operating system root volume, `/`, (same as Pre-2019.1 Structure).
- `/p4/N` is local directory on the operating system root volume,
- `/p4/N` contains symlinks for `/hxdepots`, `/hxmetadata`, and `hxlogs`, as well as tickets and trust files (same as the Pre-2019.1 structure)
- `/p4/N/bin` is local directory on the operating system root volume. The `bin` directory is the only actual directory in `/p4/N`; other items are files or symlinks to directories.

The `verify_sdp.sh` script (included in the SDP starting with SDP 2019.1) give errors if the 2019.1+ SDP structure is not in place.

Converting the SDP structure in-place to the new style requires downtime on the edge/replica of interest. While the downtime can be brief if only the SDP structure is changed, commonly the P4D is upgraded in the same maintenance window. If the P4D is pre-2019.1, a longer maintenance window will be required, depending on duration of checkpoints.

Following is the procedure to upgrade the structure in-place on a machine.

In the following sample procedure, the default SDP instance name of `1` is used, and default mount point names are used. Adapt this to your environment by applying this procedure to each instance on any given machine. If you have multiple instances, apply this procedure for each instance, one at a time.

STEP 3: Replace Instance Symlink with Directory

Move the instance symlink aside, and replace it with a regular directory. Then copy the `.p4*` files (e.g. `.p4tickets` and `.p4trust`) into the new directory. Sample commands:

```
cd /p4
mv 1 1.old_symlink
mkdir 1
cd 1
cp -p /p4/1.old_symlink/.p4t* .
```

STEP 4: Convert Fixed to Variable Metadata Symlinks

If you have Fixed Metadata Symlinks, first convert them to to Variable Metadata Symlinks. If you already have Varialbe Metadata Symlinks, proceed to STEP 5.

In this step, move the underlying directories that will be pointed to by the `root` and `offline_db` sylmink names, and move them to their `db1` and `db2` names.

```
mv /hxmetadata/p4/1/root /hxmetadata/p4/1/db1
mv /hxmetadata/p4/1/offline_db /hxmetadata/p4/1/db2
```

STEP 5: Replace Instance Symlink with Directory

Next, recreate the same symlinks you see reported by the `ls` command:

```
ls -l /p4/1.old_symlink/*
cd /p4/1

ln -s /hxmetadata/p4/1/db1 root
ln -s /hxmetadata/p4/1/db2 offline_db
```

Do not just copy the sample commands above. Pay close attention to the `ls` output, and make sure the `root` points to whatever it was pointing to before, either a directory ending in `db1` or `db2` (unless you just converted from Fixed Metadata Symlinks in STEP 4). Also confirm that `offline_db` and `root` aren't both pointing to the same directory; one should be pointing to `db1` and the other to `db2`.

Then, create additional symlinks akin to whatever else is in `/p4/1.old_symlink`

That should look something like this:

```
cd /p4/1
ln -s /hxdepots/p4/1/depots
ln -s /hxdepots/p4/1/checkpoints
ln -s /hxdepots/p4/1/checkpoints.YourEdgeServerID
ln -s /hxlogs/p4/1/logs
ln -s /hxlogs/p4/1/tmp

ls -l
```


Next, create the `bin` directory, as a local directory and copy files to it:

```
mkdir bin
cd bin
cp /p4/1.old_symlink/bin/p4d_1_init .
cp /p4/1.old_symlink/bin/p4broker_1_init .
ln -s /p4/common/bin/p4broker_1_bin p4broker_1
ln -s /p4/common/bin/p4_bin p4_1
```

Last, take a look at `/p4/1.old_symlink/bin/p4d_1` - that `p4d_1` will be either a tiny script or a symlink (depending on whether your p4d is case sensitive or not). If you server is case sensitive, it will be a symlink. If your server is case-insensitive, it will be a tiny script.

If your server is case sensitive, create the symlink like this:

```
ln -s /p4/common/bin/p4d_1_bin p4d_1
```

OR, if your server is case-sensitive, that p4d_1 will be a tiny script, so just copy it:

```
cp /p4/1.old_symlink/bin/p4d_1 .
```

Then, start your server again, and run the verify_sdp.sh script and confirm that it's happy now. If so, you should be good to run the refresh_P4ROOT_from_offline_db.sh.

EXEC STEP 4. Put new SDP common files in place.

EDITME

EXEC STEP 5. Put new SDP instance bin files in place.

EDITME

POSTOP STEP N: Cleanup - Deferred