WHITE PAPER

Best Practices for Deploying Helix Core on AWS

Introduction

A diverse range of organizations can significantly enhance their product development lifecycle and time to market by taking advantage of the near infinite compute, storage, and network resources available on AWS. This technical guide provides recommendations for implementing and optimizing development workflows on AWS by describing architectural components. This guide is based on the knowledge and experience of senior consultants at both Perforce Software and Amazon Web Services (AWS), as well as Perforce customers operating on AWS infrastructure. Please send feedback to consulting@perforce.com.

Contents

This document is intended to provide guidance for a standard deployment of Helix Core servers in the AWS cloud environment. A sample topology is illustrated, which is suitable for the demanding needs of large-scale software development. The concepts and details of mechanics are generally applicable to a wide range of Perforce workloads on AWS.

This guide presumes some familiarity with Perforce terminology ("edge servers," "replicas," etc.), and AWS terminology ("EC2 instances," "Virtual Private Clouds," etc.). References are also made to the Perforce Server Deployment Package (SDP), which is introduced in this document.

AWS Deployment Topologies

Both hybrid and exclusive AWS topologies are explored. The Version Everything mindset takes full advantage of the power of Helix Core. It entails versioning more than source code. When versioning build results, sharing modular code architectures allows for massive scale, which enables better module reuse and more fully automated continuous delivery processes. The large binary files and continuous movement of them to support Continuous Integration and Deployment processes make a hybrid infrastructure worthy of consideration. However, that must be considered against the simplicity of an exclusive AWS topology.

In both scenarios, having the master server in AWS is the best practice. This allows for the best possible data durability and availability.

Another key goal in all topologies is to leverage the AWS world-class WAN infrastructure for movement of data around the globe, including AWS Global Accelerator. Use the public internet only when it is necessary to connect from AWS data centers to your own offices or to end users.We recommend visiting instances.vantage.sh to easily review AWS instances.

Summary of a Hybrid Topology

A hybrid deployment topology takes advantage of both AWS and on-premises infrastructure to deliver a solution meeting availability, performance, and systems management needs.

The hybrid topology illustrated in this document is most appropriate for organizations that have significant investment in existing on-premises infrastructure to leverage. It helps optimize for minimization of AWS data egress charges by keeping the vast majority of routine read-only traffic, more than 98% of total traffic, within the corporate LAN environments.

The key concept of a hybrid topology is that **syncs are local**, and **submits go to AWS**. This is achieved by deploying Perforce edge servers in on-premises data centers around the globe, all connected to a master server in AWS. In reality, this is a "cloud native" master Perforce server.

Edge servers are best suited for the job, as they provide the best performance. However, edge servers require site-local backup and Perforce checkpoint operations, in addition to checkpoints that will occur in AWS for the master. It is also recommended that, like the master, each edge server be deployed with a site-local HA replica server machine to enable fast recovery in case of the loss of the edge server.

As an alternative to edge servers, using basic forwarding replicas or even more basic Perforce proxies can simplify the topology. Replicas and proxies can reduce or eliminate the need for site-local backups that are required for edge servers. However, edge servers provide better performance for a wider range of user operations. Therefore, they are the preferred approach.

There are no Perforce software license cost implications for using edge servers, forwarding replicas, or proxies as the on-premises element of a hybrid topology.

Connecting AWS to On-Premises

The easiest way to connect your infrastructure on AWS to your on-premises topology is by using a VPN. This connects your Amazon VPC (Virtual Private Cloud) to your on-premises network or data center. It is connected by a Virtual Private Gateway on the Amazon side, and by a Customer Gateway on your side. The Customer Gateway is either a physical device or software appliance. AWS has tested devices from vendors like Checkpoint, Cisco, Dell, Fortinet, Juniper, and Palo Alto. The Customer Gateway can also run on some Windows Server machines.

If the bandwidth and potential latency of an internet connection doesn't meet your throughput needs, you can use another AWS service called Direct Connect, which uses dedicated network connections from your premises to AWS.

The AWS Exclusive Topology

Environments that are considering upgrading infrastructure to meet growing software development needs, or do not want to maintain on-premises servers, should consider an AWS exclusive "all-in" topology.

Essentially, this has users connect to the AWS master server directly, with no on-premises infrastructure other than user equipment (desktops, laptops, and workstations).

An AWS exclusive infrastructure optimizes for ease of administration. It is important to note that even in cases where on-premises infrastructure is an option, it is worthwhile to do benchmarking of the cloud-only configuration. When you compare the performance of on-premises infrastructure with AWS exclusive, the results may surprise you (e.g., AWS may perform better than on-premises).

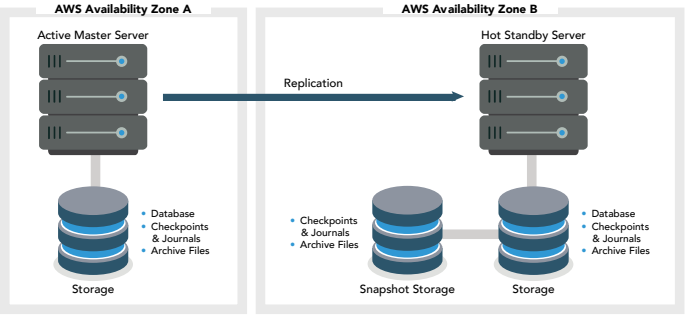

Figure 1: The AWS exclusive topology

An AWS exclusive topology may also use edge servers, standby servers, forwarding replicas, and proxies, just as a global on-premises topology would.

In addition to administration and performance benefits, an AWS exclusive topology brings additional benefits in areas such as security, disaster recovery, integration with AWS platform services (such as DNS and directory services), and in supporting other workloads such as CI/CD. A simpler caching strategy can yield savings on data egress charges, while having the simplicity of an AWS exclusive operation. These tools are easier to deploy with AWS/Perforce and you're paying only for what you use.

Build Servers

Build servers are a great example of additional workloads that are ideal for AWS configurations. You can configure a build server next to the master in an AWS cluster placement group, which provides low network latency, and/or high network throughput between them. This facilitates high velocity CI/CD applications.

If you have cyclical or unpredictable demand for CI/CD resources, you can automatically scale either using EC2 instances or containers to bring up additional build runner machines. Jenkins, for example, has a plugin that lets you automate the process of launching instances and sending traffic to them, based on build server load. You can automatically terminate the instances as the traffic goes back down.

Security Benefits of AWS Exclusive

Beyond performance and ease of administration, an AWS exclusive infrastructure offers security benefits through the use of AWS IAM (Identity and Access Management), security groups, and NACLs (network access control lists). Combined with Helix Core's built-in protection architecture, you can build out a strong defense in depth strategy, that covers physical, network, system, application, and data layers. This includes:

- Fine-grained access controls.
- Multi-factor authentication.
- HSM-based key storage.
- Server-side encryption.

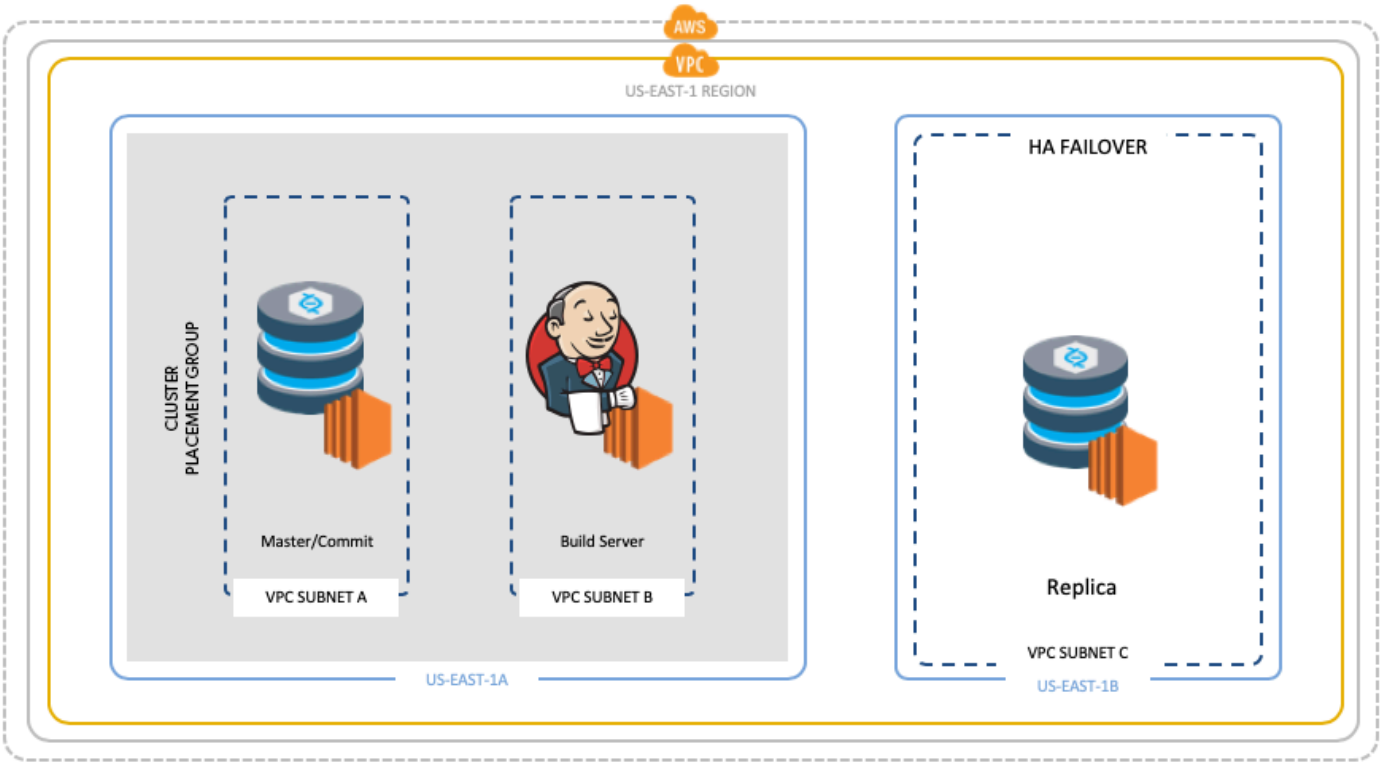

Figure 2: Build server in placement group

As with many AWS services, implementing these capabilities is generally less expensive, and requires less effort on the part of systems administrators, than similar on-premises systems delivering such functionality.

Monitoring the Topology

AWS exclusive topology brings you built-in services that can be leveraged for monitoring. AWS CloudWatch gives you directly actionable insights into the global health of your topology. Logs, metrics, and events give you a view of resources, applications, and services that you are using on AWS. It can also be used to monitor on-premises servers. CloudWatch has integration to another AWS service called SNS (Simple Notification Service), which can be used to send alerts (text, email) to your team based on metrics out of defined range. CloudWatch can even be used to fire off remediation cycles, such as rebooting, or even initiating server failover processes, in the event it is discovered that a particular instance is unresponsive.

Hybrid vs. AWS Exclusive

In many ways, hybrid should be looked at as AWS exclusive with the addition of intelligent local caching on-premises delivered by Perforce Federated Architecture. As a general statement, it is certain that the hybrid topology will greatly reduce data egress from AWS as compared to an exclusive topology. However, benchmarks testing performance benefits for end users might not show significant difference in performance for end users when leveraging on-premises hardware, perhaps counter-intuitively. Results will vary based on many factors, including greatly varying quality of on-premises hardware as compared with consistent and continuously evolving AWS infrastructure.

AWS Backup Only

Some customers with existing on-premises infrastructure and limited AWS experience can start with AWS in a very non-intrusive manner. They first create a disaster recovery (DR) Helix Core replica in AWS without changing any on-premises infrastructure. This can be useful in gaining valuable experience for your IT staff in AWS before making a larger move. It is completely transparent to users and has no impact on them.

The effectiveness of a DR replica is enhanced by using EBS snapshots with the built-in Multi-AZ durability of S3. You can extend this with cross-region S3 replication. Both of these strategies help to further ensure your assets are protected.

With S3 lifecycle policies, you can affordably keep your backups and log files "forever" at a low cost with tiered storage options driven by rules with specific SLAs and costs based on the frequency and speed of retrieval you need. If you don't want to keep them forever, you can control when they will be deleted through a policy, too.

Testing and Experimenting

Another great way to get started with AWS is the deployment of additional replicas as test environments for new infrastructure tools and workflows, for external code testing — or in-game development for new releases of the game engine.

While this approach has benefits, it also ultimately defers some of the biggest benefits that are only fully realized further into the cloud. Helix Core cloud deployments have been successful to the extent the approach allows. Hiring a guide like Perforce Consulting can help you go more fully and boldly into AWS.

Introduction of Edge Servers

For customers already familiar with Helix Core edge servers, the following is nothing new. Customers considering a cloud migration involving a hybrid topology might find their first exposure to Helix Core edge servers.

One consideration for deploying new edge servers in general is the initial introduction of an edge server into the topology. It is not transparent to users like a forwarding replica or proxy can be. This is because users must do a one-time transition from their existing workspaces, to create new workspaces that are bound to the edge server. This generally involves dealing with all opened files in the to-be-abandoned workspaces, shelving, submitting, or reverting opened files. Shelved files can be reopened in the new workspaces on the edge server. As this transition involves coordination with end users, communication of user responsibilities is key to project success.

The Rest of the Ecosystem

This guide does not discuss the rest of the Helix Core ecosystem, which can include identity management systems (such as Active Directory or LDAP), workflow management, defect tracking systems, or Continuous Integration and Deployment solutions, etc.

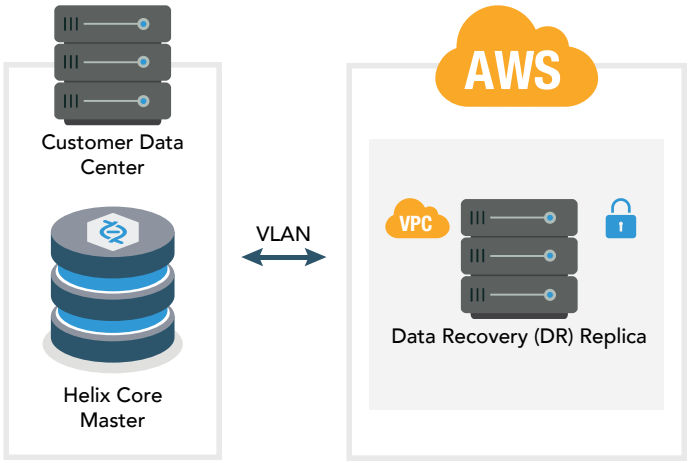

Figure 3: AWS backup only

The impact on each such system should be accounted for when doing a cloud migration. Having done many cloud migrations, the impact on such integrations is usually minimal. It typically involves opening narrow network firewall rules to, for example, allow the AWS master server to authenticate from an on-premises AD server. AWS has a service called AWS Directory Service, which is based on Microsoft Active Directory and compatible with features you may have implemented on-premises, and can validate users if the link to the on-premises server goes down.

Sample Hybrid Cloud Topology

A sample hybrid cloud topology will include a master server residing in AWS in a primary region, typically nearest the largest concentration of users. A Helix Core standby replica will be configured in a separate availability zone in the same AWS region, enabling high availability capability and real-time data protection.

Users will access Helix Core via edge servers deployed on-premises in a corporate data center, in cases where a local data center is available. Each edge server will have a site-local standby replica server to support potential failover of that edge server, enabling high availability capability at that site.

When development teams are globally distributed, additional forwarding replicas should be deployed within AWS in the region nearest major concentrations of users. The aforementioned Global Accelerator service from AWS can make this simpler to configure from a networking standpoint. The on-premises edge servers will then connect to the closest regional forwarding replica, or the master if it is closer. This approach is intended to manage AWS infrastructure when moving data across trans-continental and transoceanic distances, improving performance and reducing costs as compared to using public internet pipes.

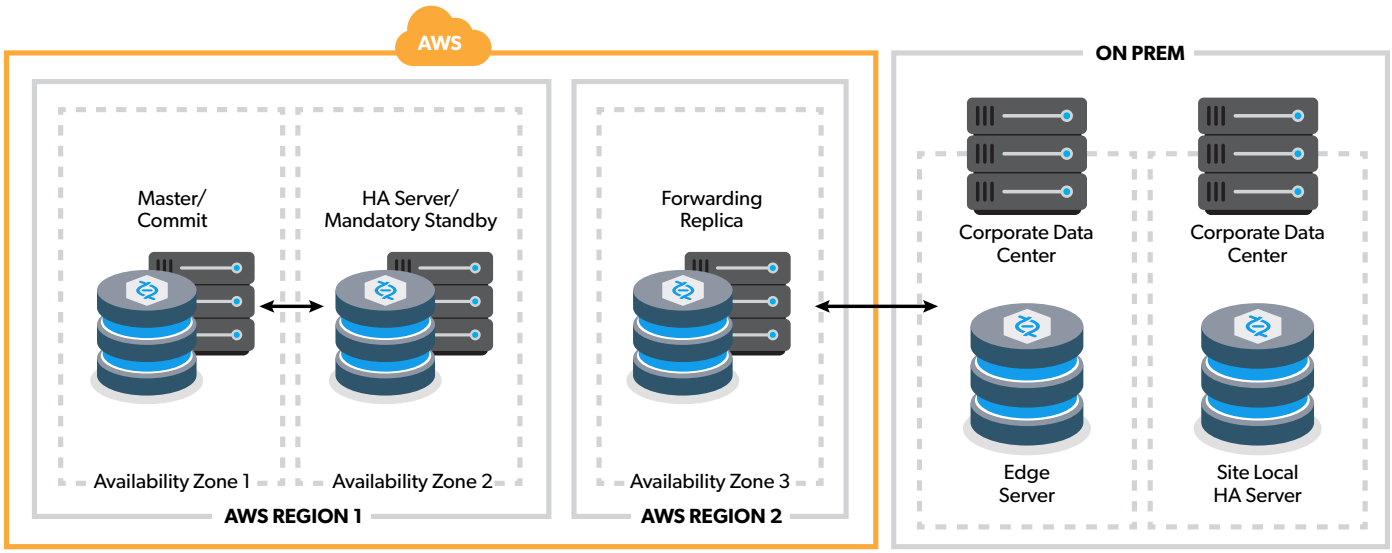

Figure 4: Example hybrid topology

For purposes of illustration, a sample topology will be described. This sample topology uses a hybrid cloud approach, achieving the highest possible data safety and availability for the master server, while leveraging on-premises infrastructure and minimizing AWS data egress charges.

The following are host names and Perforce ServerID values:

Notes:

- The host naming convention indicates that the host is dedicated to the Helix Core infrastructure (hence the p4- prefix), but does not indicate the specific role. This can change, and possibly even be reversed, in a failover situation. Host names have numeric suffixes.
- The Helix Core ServerID defines the current role of the Helix Core service on the given machine (for the given data set; this sample illustrates only a single data set).
- The ServerID values are a standard, codified in the SDP mkrep.sh script. This standard should not be altered.

- The host naming convention is illustrative of a sample best practice, but can be adapted to use an organization's host naming convention. However, it is generally deemed best that the host name not be tied to a specific role, to avoid confusion in failover scenarios.
- End users will not need to know the server host names, but will instead use a host alias that is the same as the server hostname, but with the numeric suffix and 'dash' characters removed, e.g. p4awsnva or p4syd.

Helix Core HA Replication Details

The Helix Core master/commit servers will reside on a host named p4-awsnva-01, with the -01 suffix being an arbitrary integer. This may change from time-to-time as the hardware is updated to a new operating system, a new instance type, or otherwise replaced.

EC2 Hostname	ServerID	Description
p4-awsnva-01	master.1	Master/commit server. The .1 is the SDP instance name, a data set identifier.
p4-awsnva-02	p4d_ha_awsnva	High availability (HA) server, configured as a mandatory standby replica targeting the master.
p4-awssyd-01	p4d_fr_awssyd	Forwarding replica in Sydney, Australia, targeting the master.
p4-syd-01	p4d_edge_syd	Edge server on-premises in Sydney, Australia, targeting the forwarding replica in AWS.
p4-syd-02	p4d_fs_edge_syd	Site-local HA server, configured as a standby server targeting the Sydney edge server.

The master/commit server will have a designated high availability (HA) server. For purposes of this guide, an HA replica is one that:

- Is located in the same AWS region as its master server.
- Is in a different availability zone within the same AWS region as the master server it targets.
- Is configured with a **server spec** named according to the server spec naming convention. For example, p4d_ha_awsnva for an HA server in the us-east-1 region in Northern Virginia, USA.
- Has a Services: field of the server spec set to a value of standby.
- Has an Options: field of the server spec set to a value of mandatory.
- Is running Helix Core (P4D) 2018.2 or later, as this version includes key features that our replication strategy relies on for using standby and forwarding-standby types of replica, and supports the p4 failover command.
- Has a db.replication setting of readonly.
- Has an lbr.replication setting of readonly, indicating a full replica of metadata and archive files.
- Has an rpl.journalcopy.location setting of 1, optimizing journal storage.
- Is not filtered in any way, with no use of '-T' flag in the replica's configured "pull" startup commands, and no use of various *DataFilter values in the server spec.

Helix Core DR Replication Details

In addition to an HA solution, a disaster recovery solution should be provided. In this sample topology, the forwarding replica in the AWS Sydney data center provides DR capability and routes long-distance WAN traffic on the optimal route using AWS infrastructure.

Server Deployment Package

The Perforce Server Deployment Package (SDP) is open source software available from Perforce. It is used to manage a variety of Helix Core topology components, including those deployed on EC2 instances in AWS as well as on-premises.

The SDP evolves with Helix Core, providing an ongoing reference implementation of many best practices of various kinds, from routine maintenance (zero downtime checkpoints) to performance optimization and much more.

For purposes of this document, the key things to understand about the SDP are its optimal storage layout. The SDP maintains three storage volumes for Helix Core (in addition to the OS root volume):

- /hxdepots — Contains contents of archive files as well as rotated journals and checkpoints, with potentially massive amounts of data. This is typically accessed with long-stream reads and writes. It must be backed up and should also be encrypted.
- /hxmetadata — Contains metadata databases only, frequently accessed with random I/O patterns. It must not be backed up directly, and it is constantly being written.
- /hxlogs — Contains active journal, active server log, and various other logs. This does not need to be backed up.

EBS Storage Configuration

Before configuring the EC2 instance, create and configure EBS2 storage volumes as noted to the right. This is the set of storage volumes needed for both the master and HA replica EC2 instances.

Here gp2 indicates AWS General Purpose SSD storage, and st1 indicates AWS Throughput Optimized HDD storage. The objectives:

- Spend for the performance of gp2 where it will deliver the most value, e.g., meeting the low-latency demands of the metadata volume.
- Use gp2 on the logs volume, the files in which are rotated frequently and thus should not grow extremely large.
- Utilize cheaper st1 volumes where they will perform well, i.e. for long-stream read and write operations that commonly occur on the depots volume.

The base sizes are for a production development server. Actual sizes will vary and may be larger, especially for the /hxdepots volume, which can easily be multiple terabytes (or more!) based on the scale of software to be developed with this server.

Storage Configuration for On-Premises Edge Servers

For configuring storage for the on-premises edge server machines (an edge server and its site-local HA server), use the best equivalents of the Amazon EBS settings listed above.

EC2 Instance Configuration

The following information is useful for launching EC2 instances for Helix Core servers. The headings roughly match those seen when you select “Launch Instance” from the EC2 section of the AWS Console.

After the first instance is created, subsequent instances can be more easily configured by selecting an existing instance from the AWS console to use as a template (e.g., the p4-awsnva-01 host), and selecting, “Launch more like this”.

Note that the VPC and security groups should be created before launching the instance.

EBS Name Tag	Mount	Base Size	EBS Storage Notes
`p4-awsnva-01-hxdepots`	/hxdepots	1TB	gp3, *Encrypted*
`p4-awsnva-01-hxlogs`	/hxlogs	24GB	gp3, Unencrypted
`p4-awsnva-01-hxmetadata`	/hxmetadata	64GB	gp3, Unencrypted
`p4-awsnva-02-hxdepots`	/hxdepots	1TB	gp3, *Encrypted*
`p4-awsnva-02-hxlogs`	/hxlogs	24GB	gp3, Unencrypted
`p4-awsnva-02d-hxmetadata`	/hxmetadata	64GB	gp3, Unencrypted

Choose AMI: Select an Amazon Linux AMI

Select the latest available Amazon Linux AMI to use. Amazon Linux AMIs are tuned to take optimal advantage of the AWS infrastructure, allow the widest selection of Amazon EC2 instance types, and are easy to maintain with standard RHEL/CentOS administration practices. We recommend visiting instances.vantage.sh to help you easily compare EC2 Instance features.

Choose Instance Type: Go with c6

Consider the following when selecting the Amazon EC2 instance type:

- Select an EC2 instance type in the **compute-optimized** family of instances, specifically **c6** rather than c5 instance types, unless not available in a particular region. Memory-optimized instances may also perform well. However, on balance, compute-optimized is expected to deliver the best overall performance for Helix Core development. Faster processors help avoid bottlenecks with the large number of compression/decompression operations typical when handling large digital assets.
- Select only an instance type with **EBS-Optimized** available.
- Select only an instance type with "Network Performance" indication of "**Up to 10 Gigabit**" or better.
- Select an instance with sufficient RAM. For example, for a customer developing software with total assets on the order of 2TB, we went with **c6** instance type, which has 32G of RAM available.

Configure Instance options

When configurating instance options, associate it with the Helix Core VPC. Select the appropriate availability zone (subnet), being sure the master and HA servers are in the same region but different subnets.

Ignore the following settings:

- Placement groups.
- Capacity reservation.

Check the following options:

- Protect against accidental termination.
- Enable CloudWatch detailed monitoring.

The tenancy should be shared. We do not recommend using dedicated, as this is intended to apply to data workloads involving strict regulatory data isolation compliance requirements. The term dedicated in the context of AWS tenancy does not imply superior performance. This is corroborated by detailed benchmarks done by an AWS customer. Note the following:

- There is a slight, but almost negligible, difference in CPU load. It's not much to begin with, and a preference of compute optimized instance types makes it so even that small difference is completely absorbed.
- There is no impact on network performance or latency.
- There is no impact on memory access.

Lastly, the elastic network interface should not be used with Helix Core.

Configure Storage options

Only the instance root volume needed to hold the OS is created at instance creation time. The EBS storage volumes for the different Helix Core-related /hx* volumes is created separately, as noted above.

Add Tags

Add a name tag and have that name match the intended hostname of the machine, e.g., p4-awsnval-01.

Reserved Instances

When you launch EC2 instances, the standard option is On-Demand, which means you can start, stop, and terminate them at will. Once you have solidified your configuration, and you know you have specific servers that will be long-lived, you should replace those On-Demand Instances with Reserved Instances. With Reserved Instances, you pay a certain amount upfront and commit to a term contract. This can result in a steep discount compared to the On-Demand price of the same configuration.

Availability Zones

An Availability Zone is an isolated location inside a region. Each region is made up of several Availability Zones. Each Availability Zone belongs to a single region. AZs (as AWS experts call them) are isolated from one another, but are connected through low-latency links.

Ensure that the master and standby server are in different EC2 availability zones, for optimal redundancy.

Virtual Private Cloud (VPC) and Perforce License Files

Define a Virtual Private Cloud specifically with perforce in the VPC tag name, dedicated for usage by Perforce components.

The private IP addresses of AWS instances should be the ones given to Perforce sales (sales@perforce.com) when requesting Perforce license files.

Server Deployment Package (SDP)

Server Storage and Depot Spec Standards

Perforce customers already using the SDP will not need to make adjustments to follow storage standards, as those are baked into the SDP. When moving to AWS, adopting the following standards is necessary to ensure all critical data is on an EBS volume, backed up, encrypted correctly, and on the cost-optimized storage solution.

The standards are:

- The server.depot.root configurable should be set per the SDP standard, e.g. with a value like: /p4/1/depots, which is on the /hxdepots volume.
- Depots specs should have only the default depot Map: field value of: DepotName/…
- Both within the AWS environment and for the on-premises edge servers, server machines must be configured to have exactly one logical storage volume for depots, referenced by server.depot.root. This volume must be sufficiently large to contain all archive files, optimally with the capability to grow beyond the starting capacity.

In AWS, this last point can be assumed as EBS storage can scale arbitrarily large. The goal here is to avoid an idiosyncratic installation for the edge servers on-premises with symlinks on a per-depot basis, as some customers have done with on-premises solutions due to limited physical hardware. Such idiosyncrasies may seem harmless at first, but introduce complexity and tend to cause problems in failure/recovery situations. In some cases, depots can be stored on the wrong volume where they are not backed up, and deprive P4ROOT of high-value disk space.

It must always be true that the Helix Core server.depot.root configurable be set to /p4/N/depots (per the SDP standard) and honored so that every depot can be always accessed via the path /p4/N/depots/DepotName, where N is the SDP instance name.

SDP Replication Standards

The edge server and replicas are set up using the SDP mkrep.sh script. This codifies creation of replicas with appropriate best-practice configurables based on the replica type.

Prior to using mkrep.sh, the geographic site tags must be configured in the file /p4/common/config/SiteTags.cfg.

Ensure that the P4TARGET value of each replica is set to a symbolic alias that will survive a failover. Just as end users should reference their local primary edge server or the master server using a host alias that will survive a failover, so should a replica. So for example, the Sydney edge server would have a P4TARGET value something like p4awsnva:1666, but not p4awsnva-01:1666 or p4awsnva-02:1666. The failover procedure will redirect the p4awsnva alias from the -01 to the -02 box.

SDP Sample Edge Setup

The following sample commands are illustrative. Actual commands and site-specific operational procedures would be used for an actual deployment.

The edge server in the sample topology with the SDP installed, which has an SDP instance name of 1 (the default first instance), would be configured with this command, run as **perforce@p4-awsnva-01**:

```
cd /p4/common/bin
./mkrep.sh -i 1 -t edge -s syd -r p4-awssyd-01
```

And its site-local HA replica command would be:

```
./mkrep.sh -i 1 -t fa -s syd -r p4-awssyd-02
```

Both of these commands would be run from the master server before doing further configuration.

Then, the edge server seed checkpoint is created using a command like this sample:

```
./edge_dump.sh 1 p4d_edge_syd
```

That creates an edge seed checkpoint in /p4/1/checkpoints, which must be transferred to the edge server. The SDP must also be configured for the instance. Then, recover that edge seed by running the following as **perforce@p4-syd-01**:

```
./recover_edge.sh 1
/p4/1/checkpoints/p4_1.edge_syd.seed.
ckp.1234.gz
```

Replace 1234 with the checkpoint number of the edge seed checkpoint generated above.

Next, transfer the generated edge seed checkpoint to both the edge server and its site-local replica, and replay the checkpoint on each edge server.

SDP Tweak for AWS EC2 Snapshot

When deployed in AWS, the SDP backup_functions.sh should be tweaked slightly. The goal of the adjustment is to make it so that the EC2 snapshot for the /hxdepots volume occurs at the optimal point in time, immediately after the Helix Core checkpoint operation completes. This achieves the minimum lag between the time the digital asset for recovery is created and the time it is whisked away to greater safety.

This is accomplished using AWS command line tools, doing an aws ec2 create-snapshot call. The call might look like:

```
perforce@p4-awsnval-01:/home/perforce aws
ec2 create-snapshot --description “Backup
of /hxdepots on $(date).” --volume-id vol-
aabb6c5e5a1c6cd8c
```

In this example, the volume-id specified would be the volume ID of the EBS volume mounted as the /hxdepots volume on the given EC2 instance. In similar fashion, the root volume should also be snapshotted. The /hxmetadata volume should not be snapshotted. Snapshotting the /hxlogs volume is not usually done, as the most important data is in the active P4JOURNAL file that is replicated. Other data, such as server logs, is more transitory.

Following creation of the EC2 snapshot, the snapshot should be stored in an Amazon S3 bucket, and later moved to Amazon Glacier to reduce costs. The previously mentioned S3 lifecycle policies let you automate when, and to what tier of storage, the assets go.

Tuning for Performance, Security, Safety, etc.

The SDP contains a script that promotes various best practices for tuning performance, security, data safety, etc. It is intended to be run on new SDP instances, but it also provides guidance on settings that can be applied to any existing data set.

The following settings should be checked using the p4 configure show SettingName command. Adjust if necessary to these values using p4 configure set SettingName SettingValue based on the following table.

For the most current information, see Best Practices Settings Guidance in the SDP.

Helix Management System (HMS)

The Helix Management System is included with the latest version of the SDP. Among other things, it provides key features useful for managing a sophisticated global hybrid topology.

Tight Ship Management of SDP and Extensions

A tiny dedicated Helix Core instance runs on a bastion host, p4hms.p4demo.com. Its SDP instance name is hms. It manages the SDP on all hosts where any Helix Core topology components exist. This includes Helix Core (P4D) master servers, replicas, edge servers, proxies, and brokers, and Perforce Plugin for Graphical Tools (P4GT). Tight-ship management keeps the SDP scripts and configuration files current and in sync on all hosts, using Helix Core to deploy and verify files.

Setting Name	Recommended Value	Comments
`run.users.authorize`	1	Security
`filesys.P4ROOT.min`	5G	Safety
`filesys.depot.min`	5G	Safety
`filesys.P4JOURNAL.min`	5G	Safety
`server`	4	Logging
`monitor`	1 (or 2)	Monitoring
db.reorg.disable	1	Performance
`net.tcpsize`	0	Performance
`net.autotune`	1	Performance
`db.monitor.shared`	4096	Performance
`net.backlog`	2048	Performance
`lbr.autocompress`	1	Safety, Performance
`lbr.bufsize`	1M	Performance
`filesys.bufsize`	1M	Performance
`server.start.unlicensed`	1	Licensing
`rejectList`	P4EXP,version=2014.2, Operating System	Security/Cyber Defense, Safety
`server.global.client.views`	1	Edge Functionality
`server.locks.global`	1	Edge Functionality
`auth.id`	p4_*SDPInstanceName*	Functionality
`rpl.forward.login`	1	Functionality
`dm.shelve.promote`	1	Swarm
`dm.keys.hide`	2	Swarm
`filetype.bypasslock`	1	Swarm

Helix Topology Definition

A single Helix Topology configuration file defines all SDP instances and all hosts, and is aware of which topology components operate on which hosts in "normal" as "post-failover" modes. See this Sample Helix Topology file.

Centralized Management

The Helix Topology file provides a syntax for naming any component related to any SDP instance, by combining the SDP instance name with the component name (defined in the Helix Topology configuration file). This allows any instance's P4D (or any other component) to be stopped, started, or have its status checked from the HMS server, with a command like these examples:

```
hms status 1:master
hms stop 1:p4d_edge_syd
```

Failover Plan Definition

HMS emphasizes and requires that preparation and planning for failover be defined in advance. Should failover ever be required, it can be executed swiftly according to a defined plan. The plan should account for recovery from various failure scenarios that might occur, starting with the most obvious ones like a failure of the master or edge server machines.

Based on the situation that leads one to contemplate that a failover is or might be necessary, the following plans are available as options:

Simplified Failover Execution

Failover to Local Offline DBs

Failover plans with "Local" in the name involve replacing the live (and presumable corrupt) databases on a given server machine with the spare set of databases maintained by the SDP on the same host in the offline_db folder. No change is needed to redirect users or network traffic for Local failover plans.

Local failover plans can be executed with a command similar to this:

```
hms failover local i:1 u
```

HA Failover of AWS Master

Failover plans with HA in the name involve promotion of a standby, with live and real-time replicated databases and archive files, to become the new master. User and traffic must be redirected for HA failover plans.

HA failover of the AWS master can be executed with a command similar to:

```
hms failover ha i:1 u
```

Under the covers, the hms script executes the Perforce commands necessary to achieve promotion of the failover machine to the master machine, e.g. using the p4 failover command.

Following execution of the hms script, a corporate network DNS change must be made to change the p4awsnva host alias, such that traffic routes to p4awsnva-02 instead of p4awsnva-01, to complete failover.

Failover Plan	From/To	Sample Usage Scenario
Master Local	`p4awsnva-01 (same host)`	Use when host p4awsnva-01 is OK, but databases are corrupted, e.g., due to sudden power loss or human admin error. Failover recovers databases on the same host.
Master HA	`p4awsnva-01→ p4awsnva-02`	Use when host p4awsnva-01 is unusable, or needs to be taken offline (e.g., to upgrade RAM in the EC2 instance). Failover recovers to a standby replica.
Sydney Edge Local	`p4-syd-01 (same host)`	Use when host p4-syd-01 is OK, but databases are corrupted, e.g., due to sudden power loss or human admin error. Failover recovers databases on the same host.
Sydney Edge HA	`p4-syd-01→ p4-syd-02`	Use when host p4-syd-01 is unusable, or needs to be taken offline (e.g., to upgrade RAM). Failover to a site-local standby replica.

Because the HA server in AWS is configured as a mandatory standby type of replica, global downstream edge servers and replicas will pick up where they left off with replication post-failover.

HA Failover of Sydney Edge

HA failover of the Sydney edge can be executed with a command similar to:

```
hms failover syd-edge-ha i:1 u
```

Following this command, a corporate network DNS change must be made. This changes the p4syd DNS alias used to refer to the Sydney edge server from p4- syd-01 to p4d-syd-02.

AWS Security Group Configuration

Network firewall rules should be considered when using AWS Security Groups. A security group should be created with a tag name of Perforce Helix. It should be enabled with ports opened as identified below.

If Helix Swarm is used for code review and repository browsing, then an HTTPS rule opening port 443 should be included in the Perforce Helix security group.

In the examples that follow, p4d processes run on port 1999, and p4broker processes run on port 1666. Using p4broker processes is optional, and if they are not used, then p4d processes run on 1666 instead of brokers. Brokers provide a variety of functionality to Helix Core administrators, but also introduce an extra topology component to maintain and upgrade.

If EFS is used for the /hxdepots volume (optional), then add an NFS inbound rule, named EFS, with the source being the ID of the security group itself.

HMS Hub and Spokes

The HMS server is the one Helix Core server that rules all others, in that it manages things such as backup scripts, custom trigger scripts, and even crontabs for all instances on all Helix Core-related machines, in and out of AWS. As such, this server should be configured, from an access control perspective, as a bastion host, with the highest available security. It should ideally be

deployed in AWS as a separate EC2 instance from server instances containing development work, though this host can also be deployed on-premises.

The HMS server (p4hms.p4demo.com) must allow traffic as follows:

- inbound on port 7467 (always SSL encrypted) from all p4d servers managed by HMS.
- inbound on port 7468 (always SSL encrypted) from all p4d servers managed by HMS.

Inbound Rules for On-premises Edge Servers

On-premises edge servers require inbound access as follows:

- Port 1666 from the internal corporate network
- Port 1999 from the internal corporate network
- Port 22 (SSH) from the HMS server (a Linux bastion host).

Inbound Rules for AWS Servers

AWS servers require inbound access as follows:

- Into the VPC on port 1999 (optionally SSL encrypted) from VA1 edge servers.
- Into the VPC on port 1666 (optionally SSL encrypted) from the HMS server.
- Into Port 22 (SSH) from the HMS server and Linux bastion host(s).

Outbound Rules for all Servers

Outbound rules can be restricted or unrestricted, per corporate policy. If they are to be restricted, restrictions should follow these guidelines that balance security and ease of administration: Host and Port Access List for Perforce Helix Servers.

Perforce SSL Encryption

Perforce provides SSL encryption that can add an extra layer of defense, protecting data in transit across the network. (EBS storage provides encryption of essential data at rest on the on Helix Core servers.)

SSL can be used to enhance security. Benchmarks indicate that the performance costs for using SSL are minimal, often negligible.

SSL may not be required if good perimeter security is in place, including within AWS, on-premises, and between AWS and the on-premises infrastructure.

However, SSL does introduce a few complexities:

- As with any OpenSSL technology, certificates need to be generated and managed. Helix Core server products can generate their own certificates, so this complexity is limited to dealing with occasional certificate expiration.
- Failover solutions are somewhat more complex and less transparent with SSL, as the goal of failover — to transparently change the server that a user is connected to — is diametrically opposed to the goal of SSL, which is to ensure users are aware when a server they think they are communicating with changes. Theoretical options exist to add this, such as getting all users to trust all possible failover target servers. Customers using SSL thus far have been willing to accept that failover may require users to redo the trust of the new server after failover. This approach may involve retooling for robotic users to incorporate trust and login logic into Helix Core interactions in code, as a best practice.
- In some cases, Helix Core servers under heavy load may — due to SSL timeouts — drop connections, resulting in user errors for conditions that may otherwise have successfully completed after a temporary pause. Similarly, potentially useful Helix Core server log messages are sometimes replaced with unhelpful SSL timeout errors. These have thus far not been big issues for customers to deter them from using SSL.
- When HMS is deployed on a bastion host, the HMS instance is always configured with SSL, as this server can contain sensitive information that could be used to access other "real data" instances.

3.11.6 Other Rules

Depending on what other systems are integrated with Perforce, additional network firewall rules may be needed. For example, if Helix Core LDAP integration is used, a network firewall rule may be needed to enable access (e.g. on port 389 or 636) from the AWS master server into an on-premises corporate Active Directory server.

What's Next?

Evolution

The AWS infrastructure and managed services offerings continue to evolve as AWS and its partners continue to innovate. This guide will continue to evolve to provide ongoing guidance for customers looking to deploy Helix Core on AWS for development.

The following are being considered:

- Incorporation of EFS as a best practice. EFS has already been employed to good effect in some customer environments for the /hxdepots volume, but there are some limitations to be considered, such as limits on active connections.
- AWS CloudFormation will be utilized.
- SDP and HMS improvements may simplify operation in AWS, eliminating any tweaks.
- AWS Route 53 will be explored.
- A reference implementation may be captured with CloudFormation, including illustration of backup of EC2 snapshots and the move to Glacier.
- The Perforce Battle School training class, which currently simulates a sophisticated global topology on-premises using cloud-hosted virtual machines, may itself be migrated to AWS.
- HA and DR solutions may continue to evolve.
- The benefits and costs of container technology such as Kubernetes will be explored with respect to applicability to Perforce server deployment.
- In addition to following general infrastructure as code best practices, using CloudFormation may be applied in specific development use cases — for example, spinning up a new game or virtual production studio. There is a common need to quickly spin up a new development studio, in a remote location with respect to master server. Parameterized CloudFormation templates could be employed to spin up edge servers with a subset of depots to enable limited collaboration for the new studio.

SDP Cloud Deploy

The SDP Cloud Deploy project is an early stage prototype implementation demonstrating use of simple template files to build a sample Helix Core environment, including building a replica from scratch while starting with an AWS account.

Additional Comments

Review additional details regarding decisions for this document.

[1]Ideally, consistent site tags should be used in Helix Core server spec names and host names, e.g., awsnva as a tag for the AWS us-east-1 region in northern Virginia, USA.

[2]Note that Amazon EFS (essentially NFS storage in AWS, but better) was considered for the /hxdepots volumes. EFS is well suited for this purpose, but with some drawbacks. The chief limitation of EFS is that it cannot be snapshotted without first doing a rsync to an EBS volume (though EFS to EFS backup is possible). This means that avoidance of capacity reservation will not apply if backing up to an EBS volume. That said, EFS has compelling advantages, and it will be revisited for use with the /hxdepots volume in the future. The soon-to-be-released EFS IA (Infrequent Access) provides a cost optimization that is particularly well suited to typical Helix Core workloads.

As of this writing, however, EBS volumes are more proven, have ideal functionality (e.g., snapshot capability) and the best possible performance (with lower latency than EFS). EFS volumes also require counter-intuitive artificial "garbage data injection," and optimization of "burst credits" to make them artificially larger than they would need to be based on data needs, as performance with EFS is a factor of size — bigger is better for EFS.

About Perforce

Perforce powers innovation at unrivaled scale. With a portfolio of scalable DevOps solutions, we help modern enterprises overcome complex product development challenges by improving productivity, visibility, and security throughout the product lifecycle.

Our portfolio includes solutions for Agile planning & ALM, API management, automated mobile & web testing, embeddable analytics, open source support, repository management, static & dynamic code analysis, version control, and more. With over 9,000 customers, Perforce is trusted by the world's leading brands, including NVIDIA, Pixar, Scania, Ubisoft, and VMware. For more information, visit perforce.com.

Contact Us

perforce.com/support