HELIX4GIT CONNECTOR

WITH ONE-WAY MIRRORING

Topics

- Helix4Git Summary
- Connector Components
- Installation
- Configuration of the Connector
- Connector Commands
- SSH Configuration
- Git Push & Clone of Repos
- Special Connector Commands
- HTTPS & SSH Configuration for GitLab One-Way Mirroring
- Helix4Git Environments
- Helpful Links

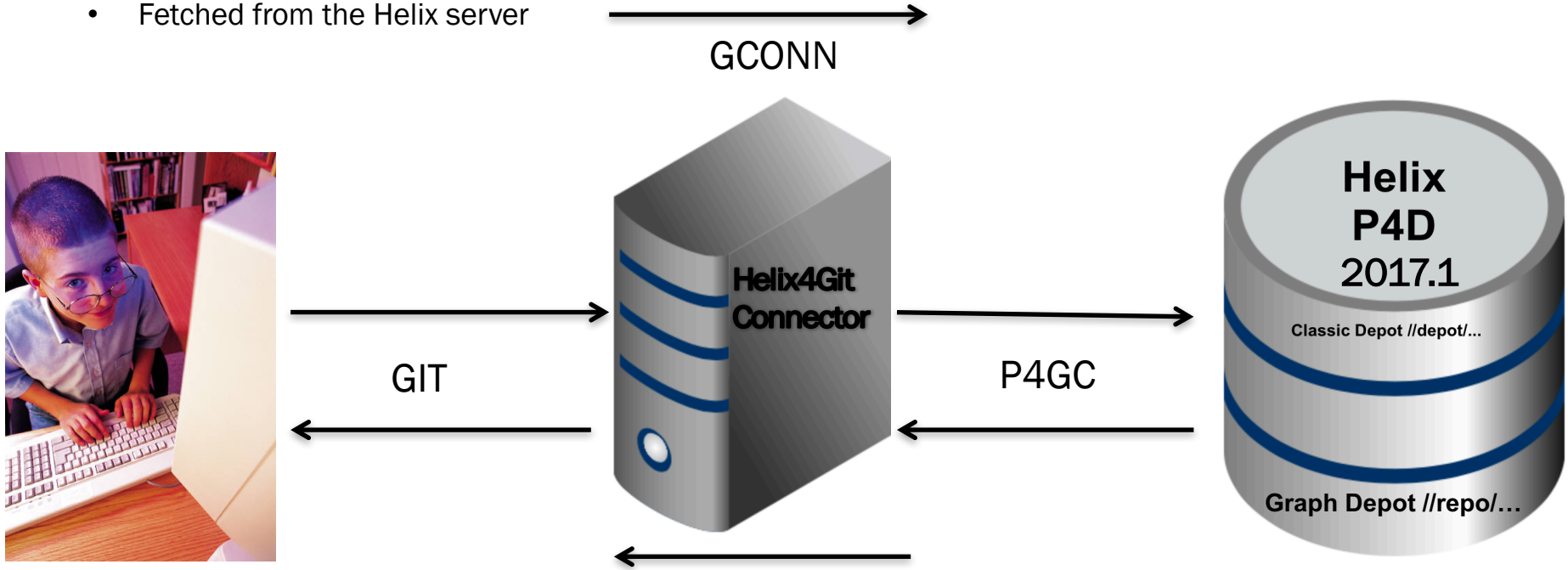
Helix4Git Connector

Summary

HELIX4GIT

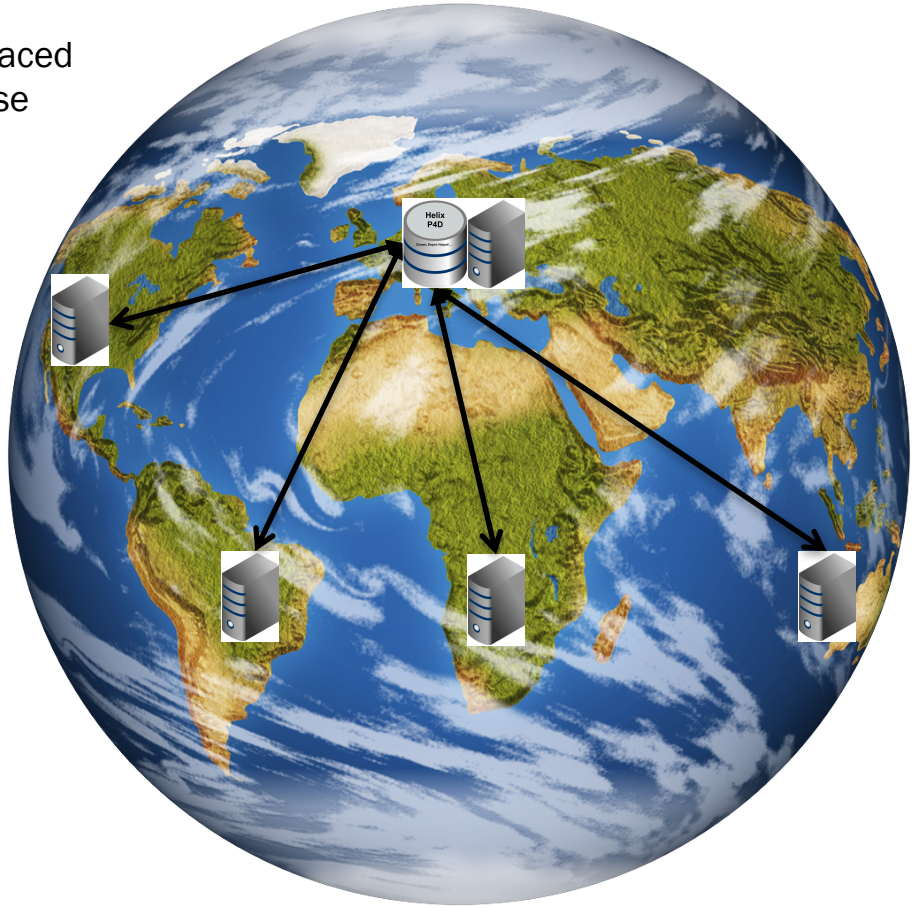
We have the new depot of type graph for git data, why do we need the Connector?

- The Connector serves as a remote Git server so that developers can seamlessly clone, pull and push using git commands and have their changes:
- Pushed to the Helix server
- Fetched from the Helix server



HELIX4GIT

- The Connectors can be placed in remote locations in close proximity to the teams



Helix4Git Connector

Components

CONNECTOR COMPONENTS - /OPT/PERFORCE/GIT-CONNECTOR/BIN

gconn

- Handles authentication and permission checks, mirrorhooks, syncing of the SSH keys

git-remote-p4gc

- Remote helper git-remote-p4gc invoked by git when it needs to interact with remote repositories git does not support natively.
- Converts the git commands into Helix graph commands
  - Push: git-receive-pack -> p4 graph receive-pack
  - Pull: git-upload-pack -> p4 graph pack-objects

P4 commands are run as the Helix gconn-user

login-gconn-user.sh

- Useful if you change the gconn-user password or need to generate a new ticket

Installing the
Helix4Git Connector

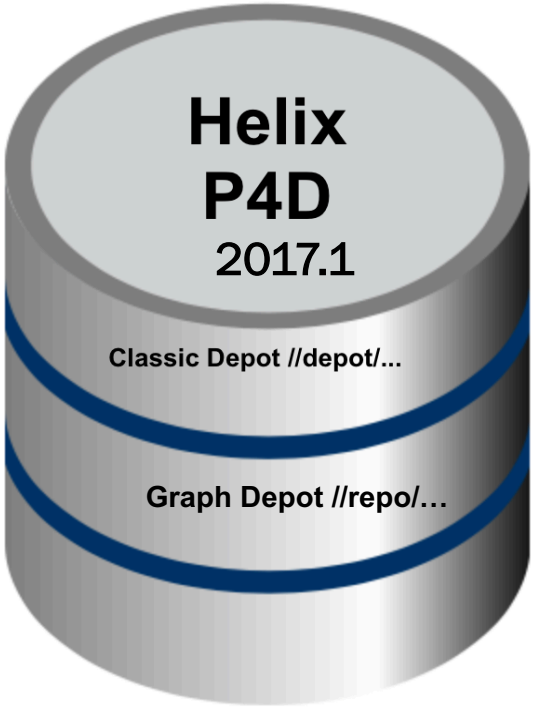
HELIX4GIT CONNECTOR PLATFORMS

- Package installation is supported on the following OS platforms

CentOS 6
Community ENTerprise Operating System

- SSH supported
- HTTPS supported

- SSH support only
- May require manual installation of a newer version of Git, 1.8.3 or higher



CentOS 7

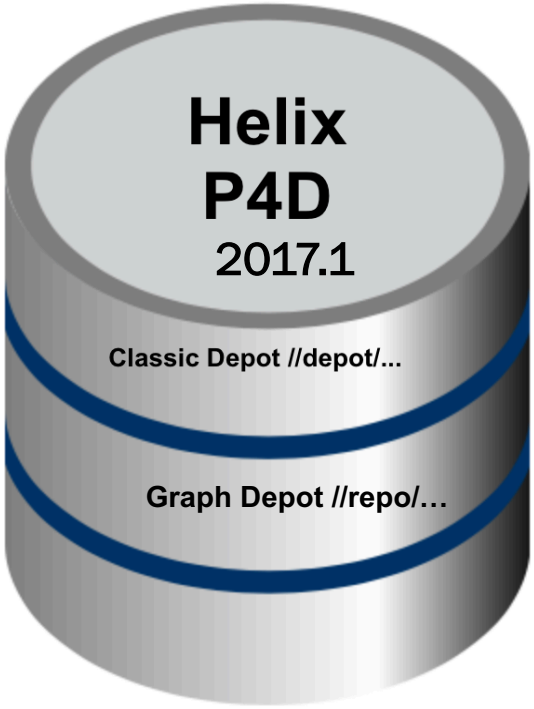
ubuntu 14.04 LTS

ubuntu 16.04 LTS

HELIX4GIT CONNECTOR INSTALLATION

Installation requires the following steps:

- Upgrade Helix server to 2017.1



- Create package repository file pointing to https://package.perforce.com/
- Import the packaging key
- Install the Helix4Git Connector package

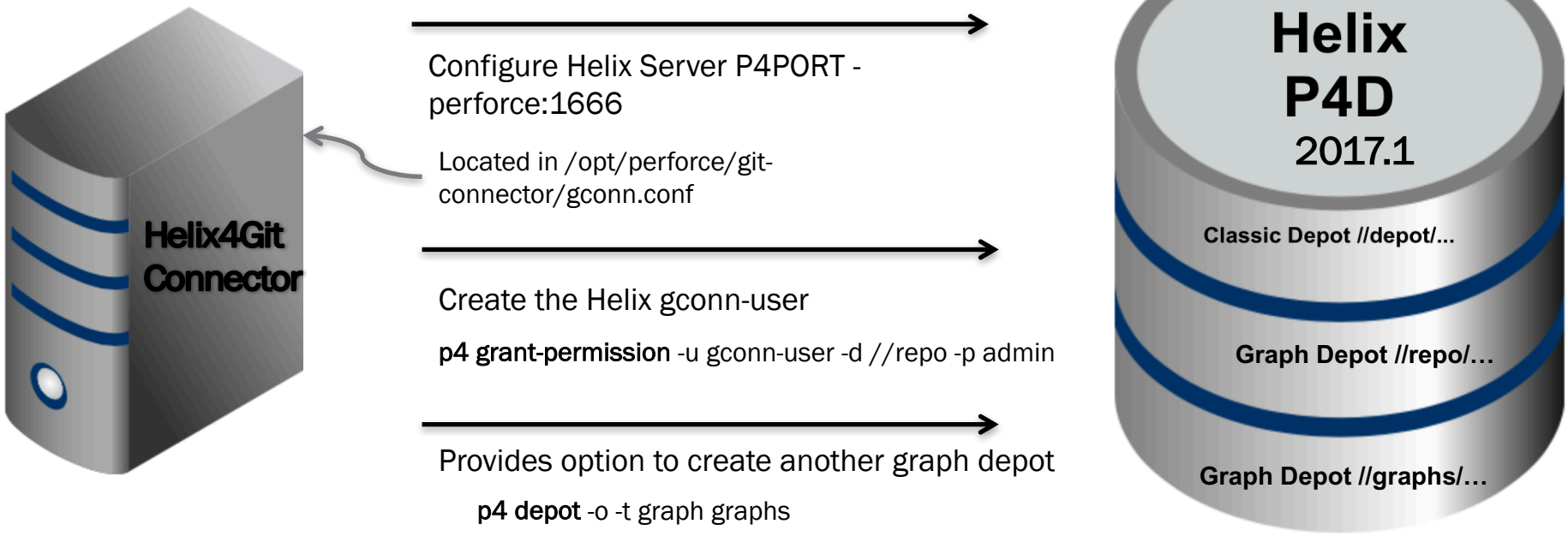
Configuring the Helix4Git Connector using configure-git-connector.sh

Helix4Git Connector Configurations

```
UBU 16.04 GConn : perforce ~ 2 > sudo /opt/perforce/git-connector/bin/configure-git-connector.sh
[sudo] password for perforce:

Summary of arguments passed:

Helix server P4PORT                [(not specified)]
Helix super-user                   [(not specified)]
Helix super-user password          [(not specified)]
New Git depot name                 [(not specified)]
GitConnector user password         [(not specified)]
Configure HTTPS?                   [false]
Configure SSH?                     [false]
GitConnector SSH system user       [git]
Home directory for SSH system user [/home/git]
SSH key update interval            [10]
Server ID                          [gconn-gconn-ubuntu16]

For a list of other options, type Ctrl-C to exit, and then run:
$ sudo /opt/perforce/git-connector/bin/configure-git-connector.sh --help

You have entered interactive configuration for GitConnector. This script will
ask a series of questions, and use your answers to configure GitConnector for
first-time use. Options passed in from the command line or automatically
discovered in the environment are presented as defaults. You may press enter
to accept them, or enter an alternative.
```

HELIX4GIT CONNECTOR CONFIGURATION

CONFIGURE-GIT-CONNECTOR.SH



HELIX4GIT CONNECTOR CONFIGURATION

The Connector supports two transfer protocols

HTTP SSH

Configure HTTPS

- Calls **configure-https-auth.sh** to install and configure Apache

Configure SSH

- Creates the git user: /home/git
- Sets up the gconn cronjob in /etc/cron.d/gconn
  - The cron contains **sync-ssh-keys** to sync the pubkeys
    - From Helix
    - Into the Connector's /home/git/.ssh/authorized_keys

Helix4Git Connector Configurations

```
/opt/perforce/git-connector/gconn.conf
    gconn: {
        log: { path: /opt/perforce/git-connector/logs/gconn.log,
            levels: [ time=1, gconn=1 ] },
        reposDir: /opt/perforce/git-connector/repos,
        p4User: gconn-user,
        p4Port: perforce:1666,
        p4TicketsFile: /opt/perforce/git-connector/.p4tickets,
        p4TrustFile: /opt/perforce/git-connector/.p4trust,
        authKeysFile: /home/git/.ssh/authorized_keys,
        gitExecPath: /usr/bin,
        envPath: /usr/bin:/usr/local/bin:/opt/perforce/git-connector/bin,
        authGroup: gconn-auth },
    p4gc: {
        log: { path: /opt/perforce/git-connector/logs/p4gc.log,
            levels: [ time=1, gconn=1 ] }
    }
```

Connector SSH Configurations

```
git@gconn-ubuntu16:~$ pwd
/home/git

git@gconn-ubuntu16:~$ ls -al
drwxr-xr-x 4 git  git  4096 Mar  6 15:35 .
drwxr-xr-x 4 root root 4096 Aug  2  2016 ..
-rw-r----- 1 git  git   365 Apr 21 14:33 .bash_history
drwx------ 2 git  git  4096 Aug  8  2016 .cache
-rw-rw-r-- 1 git  git   128 Feb 13 11:10 .p4enviro
-r-------- 1 git  git   119 Aug  2  2016 .p4tickets
drwx------ 2 git  git  4096 Apr 27 09:20 .ssh

git@gconn-ubuntu16:~$ head -1 /home/git/.ssh/authorized_keys
command="export GCONN_CONFIG=/opt/perforce/git-connector/gconn.
connector/bin; gconn ${SSH_ORIGINAL_COMMAND} --user=super",no-p
c2EAAAACA<snip>vkanczes@git-ubuntu14

git@gconn-ubuntu16:~$ cat /etc/cron.d/gconn
GCONN_CONFIG = /opt/perforce/git-connector/gconn.conf
# CAUTION: This file is updated by configure-git-connector.sh
# update auth keys every 10 minute(s)
*/10 * * * * git /usr/bin/gconn sync-ssh-keys
```

Helix: server spec

```
p4 -ztag servers
... ServerID gconn-gconn-centos6
... Name
... Address
... Type connector
... Services git-connector
... Description    This GitConnector service was configured on [Mon Feb 13 11:57:08 PST 2017] by user [vkanczes].

   Configuration summary:
     Platform: [CentOS release 6.8 (Final)]
     HTTPS authentication: [true]
     SSH authentication: [true]
     SSH access system user: [git]

... User gconn-user

... ServerID gconn-gconn-centos7
... Name
... Address
... Type connector
... Services git-connector
... Description    This GitConnector service was configured on [Mon Apr 24 15:09:30 PDT 2017] by user [super].

   Configuration summary:
     Platform: [CentOS Linux release 7.2.1511 (Core) ]
     HTTPS authentication: [true]
     SSH authentication: [true]
     SSH access system user: [git]

... User gconn-user
```

Helix4Git Connector

Configuration is Complete

Connector Commands

CONNECTOR COMMANDS

Set environment variable:
export GCONN_CONFIG=/opt/perforce/git-connector/gconn.conf

gconn --help

- Displays the commands and their uses

gconn –V | --version

- Displays the Connector version

  Rev. GCONN/LINUX26X86_64/2017.1. /1509248

gconn **–mirrorhooks** add | remove | list

- Configuration and listing of hooks required for one-way mirroring

gconn **sync-ssh-keys**

- Syncs the keys from Helix to the Connector

Helix4Git Connector Commands

```
gconn --help
usage: gconn command [options...] [arguments...]

GCONN_CONFIG variable with the path to configuration file must be set.

options:
```

```
gconn --mirrorhooks list
Invalid mirror config: ./repo/http-mirror-test-14.git
http://gitlabee.das.perforce.com/vkanczes/repo3.git -> //repo/repo3.git
http://vkanczes:password11@vkanczes-gitlab-dr.das.perforce.com/vkanczes/repo4.git -> //repo/repo4.git
Invalid mirror config: ./repo/ssh-mirror-test-14.git
git@gitlabee.das.perforce.com:vkanczes/ssh-test.git -> //repo/ssh-test.git
git@gitlabee.das.perforce.com:vkanczes/ssh-test3.git -> //repo/ssh-test3.git
http://gitlabee.das.perforce.com/vkanczes/test-mirror1.git -> //repos/test-mirror1.git
http://vkanczes:password11@gitlabee.das.perforce.com/vkanczes/test2.git -> //repos/test2.git
git@gitlabee.das.perforce.com:vkanczes/test3.git -> //repos/test3.git
git@gitlabee.das.perforce.com:vkanczes/test4.git -> //repos/test4.git
git@gitlabee.das.perforce.com:vkanczes/test5.git -> //repos/test5.git
```

```
          --mirrorhooks remove repo/repoA
          --mirrorhooks list

commands:
   sync-ssh-keys                  Force update of p4 pubkeys from Helix.
                                  This command must be run from OS account
                                  of user used for SSH authentication
                                  (usually 'git').
```

SSH KEY Configuration

http://kbportal.perforce.com/article/15174

CONNECTOR- SYNC SSH KEY

User can generate their SSH keys

- ssh-keygen

Admin uploads the user's SSH public key

- **p4 pubkey** –u vkanczes –s work –i < id_rsa.pub

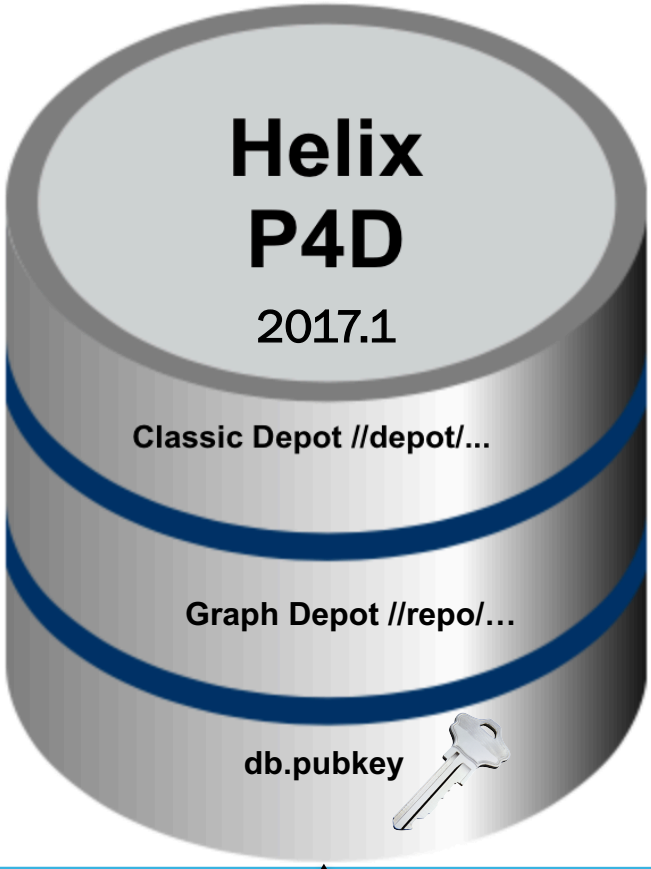
User uploads their own SSH public key

- **p4 pubkey**–s work –i < id_rsa.pub

Git cron executes **sync-ssh-keys**

SSH public keys are added

- /home/git/.ssh/authorized_keys



Git Push & Clone of Repos

Using

Helix4Git Connector

SSH PUSH to create a REPO with Helix4Git Connector

Requirements:

- User has the permission to ‘create-repo’

  p4 grant-permission –u vkanczes –d repo –p create-repo

- gconn-user has the admin permission for the graph depot

  p4 grant-permission –u gconn-user –d repo –p admin

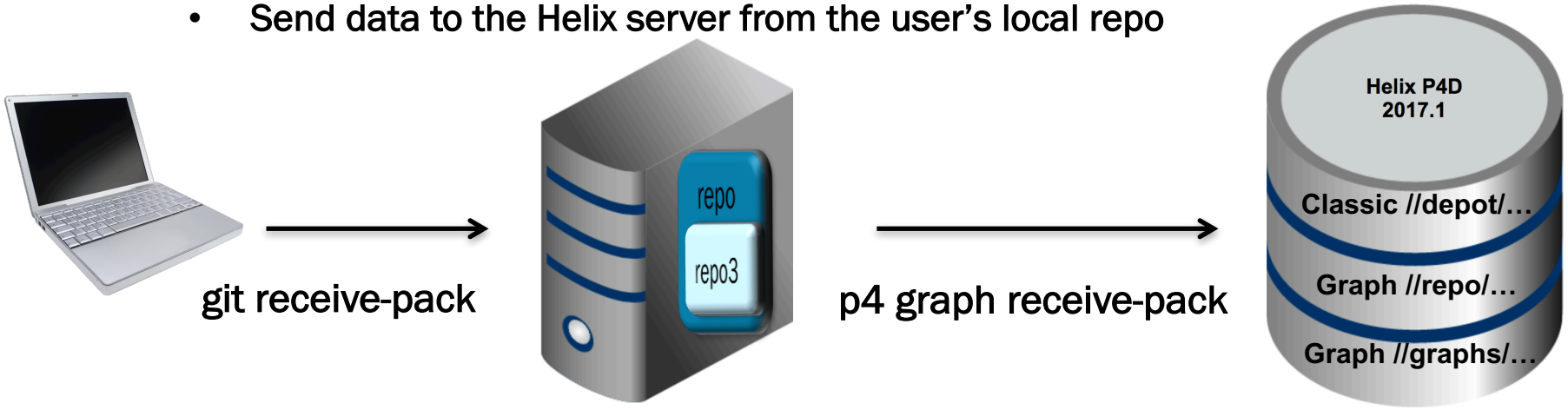
- User’s SSH key has been added to the Connector
- Server configurable dm.repo.noautocreate: 0
  - Default setting is 0 (off)

GIT -> GCONN -> GIT-REMOTE-P4GC -> GCONN -> GIT

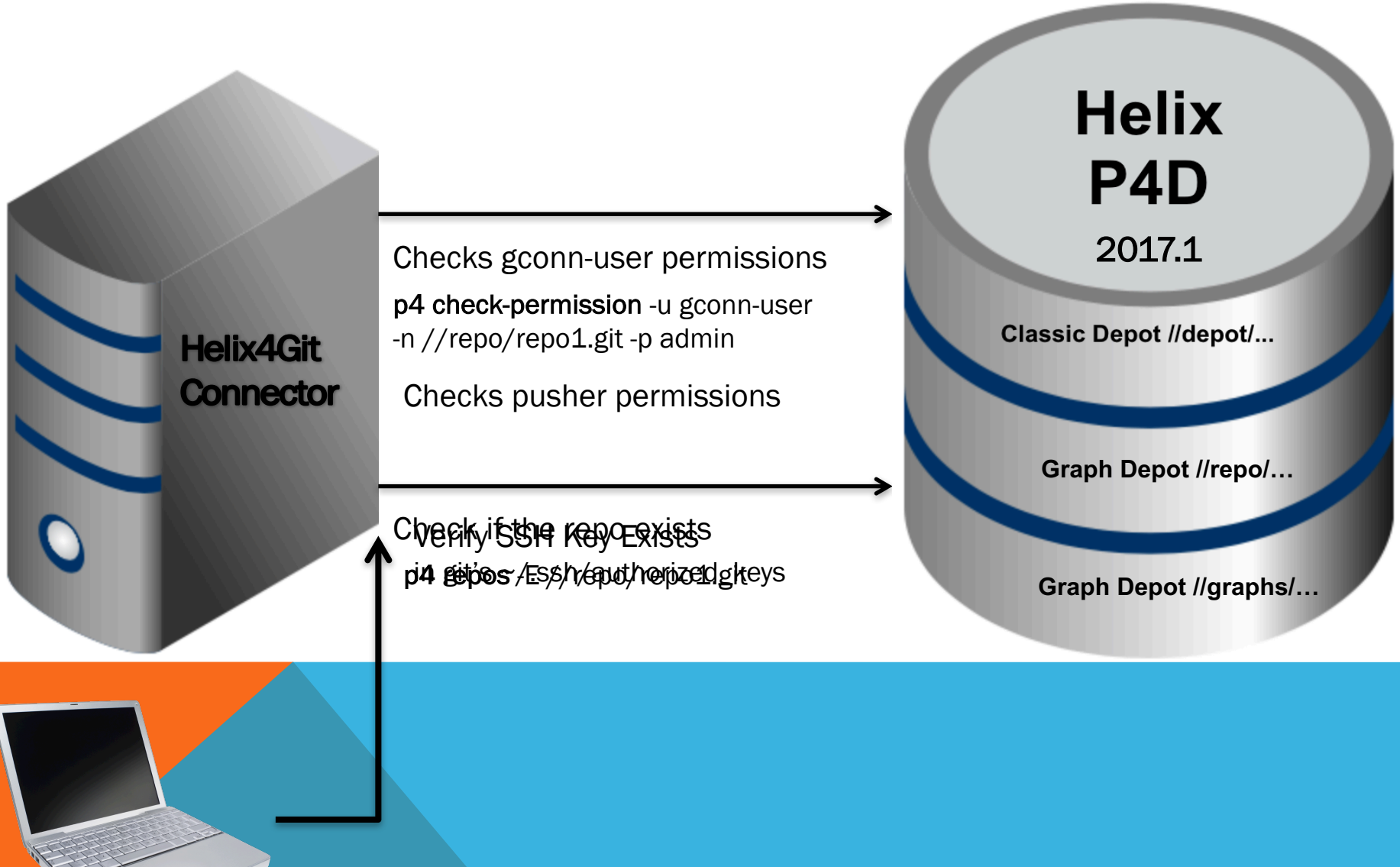
Remote helper git-remote-p4gc invoked by git when it needs to interact with remote repositories git does not support natively.

git push

- Send data to the Helix server from the user's local repo



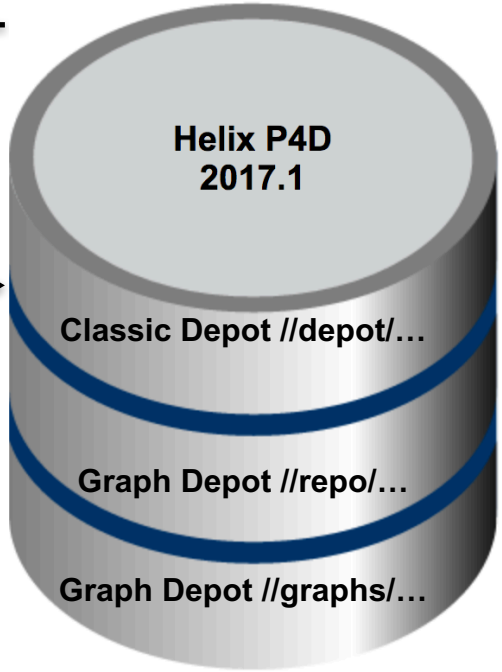
HELIX4GIT
GCONN AUTHENTICATION & PERMISSIONS



PUSH THE REPO – GCONN/GIT ACTIONS

Creates the remotes to/from Helix

git-remote-p4gc origin p4gc://perforce:1666/repo/repo1.git



Installs the pre-receive hook

- gconn --pre-receive

Installs the post-receive hook

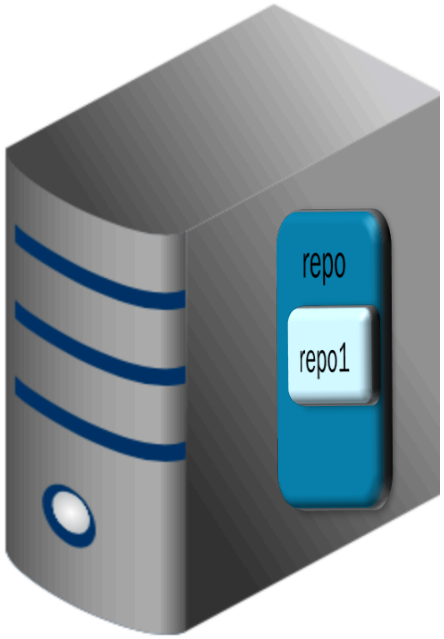
- gconn --post-receive

Create the repo

git init –bare repo/repo1.git

git push git@gconn:repo/repo1 master

CREATE THE REPO VIA PUSH
GIT ACTIONS



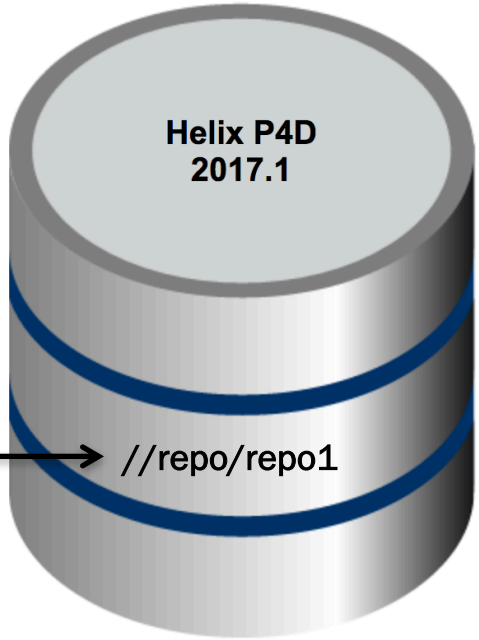
git-receive-pack /repo/repo1

pre-receive hook executed

Calls p4gc to push into Helix

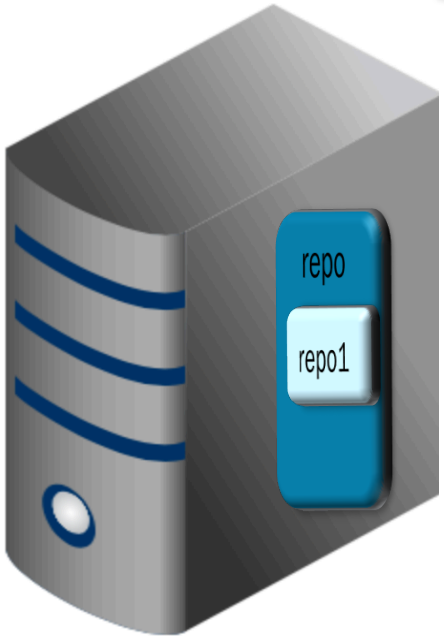
- git push –force origin 32ce12a:refs/heads/master
- Updates: refs/heads/master
- Create the repo in Helix
- **p4 repo** -o //repo/repo1.git

Calls the post-receive hook



git push git@gconn:repo/repo1 master

CREATE THE REPO VIA PUSH
P4GC ACTIONS

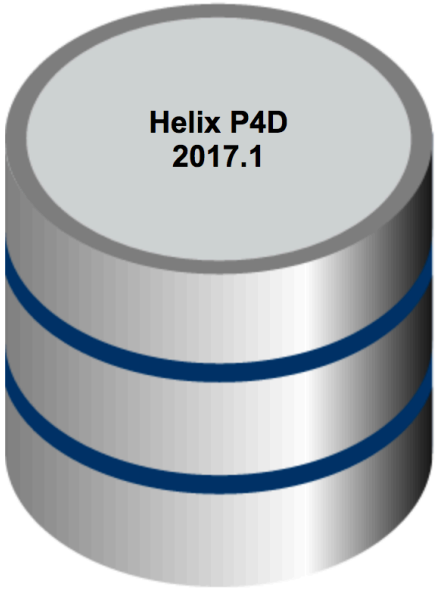


Check repo branches

- **p4 graph show-ref** -n //repo/repo1.git

Creates the pack file to push to Helix

- git pack-objects -q --revs ./gconn.Zw4T3w/p4gc-push



Updates Helix with the branch ref 32ce12a, refs/heads/master} and sends the new pack files

p4 graph receive-pack -u vkanczes -n //repo/repo1.git -i ./gconn.Zw4T3w/p4gc-push-18981f3acc6d6688cce6f38b9130c244a3bb3e2e.pack -p ./gconn.ugSqnw/refUpdateList

git push git@gconn:repo/repo1 master

GConn Push Log

```
pid 2015: Commandline: gconn git-receive-pack 'repo/repo11' --user=vkanczes
pid 2015: IsAllowed: lookup user=vkanczes for perm=admin on repo: //repo/repo11.git
pid 2015: P4Client::RunCommand: {p4 -ztag -p 10.5.10.174:17100 -u gconn-user check-permission -u gconn-user -n //repo/repo11.git -p admin}
pid 2015: P4Client::RunCommand: {p4 -ztag -p 10.5.10.174:17100 -u gconn-user repos -E //repo/repo11.git}
pid 2015: system(/usr/bin/git init --bare repo/repo11.git > /dev/null) succeeded
pid 2015: Configured remote for port 10.5.10.174:17100
pid 2015: installed pre-receive hook
pid 2015: installed post-receive hook
pid 2015: Launching /usr/bin/git-receive-pack repo/repo11
pid 2036: Commandline: gconn --pre-receive
pid 2036: RunCmd(/usr/bin/git push --force origin f3956152bf5cbecbd26a2e42ed82bca9c437d86f:refs/heads/master) succeeded
pid 2043: Commandline: gconn --post-receive
pid 2043: CmdPostReceive repo='repo/repo11.git', HEAD='refs/heads/master'
pid 2043: P4Client::RunCommand: {p4 -ztag -p 10.5.10.174:17100 -u gconn-user repo -o //repo/repo11.git}
pid 2043: Repo //repo/repo11.git spec:
pid 2043:     Repo --> //repo/repo11
pid 2043:     Owner --> vkanczes
pid 2043:     Created --> 2017/05/15 15:17:17
pid 2043:     Pushed --> 2017/05/15 15:17:17
pid 2043:     Description --> Automatically created by the initial push.
pid 2043: cmd="git show-ref refs/heads/master"
pid 2043: {git show-ref} <<< f3956152bf5cbecbd26a2e42ed82bca9c437d86f refs/heads/master
pid 2015: RunCmd(/usr/bin/git-receive-pack repo/repo11) succeeded
```

P4GC Log

```
pid 2026: git-remote-p4gc origin p4gc://10.5.10.174:17100/repo/repo11.git
pid 2026: P4Client::RunCommand: {p4 -ztag -p 10.5.10.174:17100 -u gconn-user graph show-ref -n //repo/repo11.git}
pid 2038: bool PushHelper::BuildPackFile() enter
pid 2038: cmd="git pack-objects -q --revs ./gconn.t5h7t0/p4gc-push"
pid 2038: {git pack-objects} <<< 1876255831be64124d0634fe7e328d262adde7c1
pid 2038: bool PushHelper::SendPackFile() enter
pid 2038: packfile = './gconn.t5h7t0/p4gc-push-1876255831be64124d0634fe7e328d262adde7c1.pack'
pid 2038: push refs/heads/master -> f3956152bf5cbecbd26a2e42ed82bca9c437d86f force
pid 2038: P4Client::RunCommand: {p4 -ztag -p 10.5.10.174:17100 -u gconn-user graph receive-pack -u vkanczes
                -n //repo/repo11.git
                -i ./gconn.t5h7t0/p4gc-push-1876255831be64124d0634fe7e328d262adde7c1.pack
                -p ./gconn.Tb60zX/refUpdateList}
pid 2038: {write receive-pack f3956152bf5cbecbd26a2e42ed82bca9c437d86f, refs/heads/master} >>>
```

GConn Push Log

```
pid 2015: Commandline: gconn git-receive-pack 'repo/repo11' --user=vkanczes
pid 2015: IsAllowed: lookup user=vkanczes for perm=admin on repo: //repo/repo11.git
pid 2015: P4Client::RunCommand: {p4 -ztag -p 10.5.10.174:17100 -u gconn-user check-permission -u gconn-user -n //repo/repo11.git -p admin}
pid 2015: P4Client::RunCommand: {p4 -ztag -p 10.5.10.174:17100 -u gconn-user repos -E //repo/repo11.git}
pid 2015: system(/usr/bin/git init --bare repo/repo11.git > /dev/null) succeeded
pid 2015: Configured remote for port 10.5.10.174:17100
pid 2015: installed pre-receive hook
pid 2015: installed post-receive hook
pid 2015: Launching /usr/bin/git-receive-pack repo/repo11
pid 2036: Commandline: gconn --pre-receive
pid 2036: RunCmd(/usr/bin/git push --force origin f3956152bf5cbecbd26a2e42ed82bca9c437d86f:refs/heads/master) succeeded
pid 2043: Commandline: gconn --post-receive
pid 2043: CmdPostReceive repo='repo/repo11.git', HEAD='refs/heads/master'
pid 2043: P4Client::RunCommand: {p4 -ztag -p 10.5.10.174:17100 -u gconn-user repo -o //repo/repo11.git}
pid 2043: Repo //repo/repo11.git spec:
pid 2043:     Repo --> //repo/repo11
pid 2043:     Owner --> vkanczes
pid 2043:     Created --> 2017/05/15 15:17:17
pid 2043:     Pushed --> 2017/05/15 15:17:17
pid 2043:     Description --> Automatically created by the initial push.
pid 2043: cmd="git show-ref refs/heads/master"
pid 2043: {git show-ref} <<< f3956152bf5cbecbd26a2e42ed82bca9c437d86f refs/heads/master
pid 2015: RunCmd(/usr/bin/git-receive-pack repo/repo11) succeeded
```

Helix Log

```
pid 3247 gconn-user@gconn-ubuntu16 10.5.10.224 [unknown/v82] 'user-check-permission -u gconn-user -n //repo/repo11.git -p admin'
pid 3248 gconn-user@gconn-ubuntu16 10.5.10.224 [unknown/v82] 'user-repos -E //repo/repo11.git'
pid 3249 gconn-user@gconn-ubuntu16 10.5.10.224 [unknown/v82] 'user-graph show-ref -n //repo/repo11.git'
pid 3250 gconn-user@gconn-ubuntu16 10.5.10.224 [unknown/v82] 'user-graph show-ref -n //repo/repo11.git'
pid 3251 gconn-user@gconn-ubuntu16 10.5.10.224 [unknown/v82] 'user-graph receive-pack
                -u vkanczes
                -n //repo/repo11.git
                -i ./gconn.t5h7t0/p4gc-push-1876255831be64124d0634fe7e328d262adde7c1.pack
                -p ./gconn.Tb60zX/refUpdateList'
Receive-pack: transfer ./gconn.t5h7t0/p4gc-push-1876255831be64124d0634fe7e328d262adde7c1.pack
                       to
                       repo/repo11.git/objects/pack/p4gc-push-1876255831be64124d0634fe7e328d262adde7c1.pack
Receive-pack: transfer ./gconn.t5h7t0/p4gc-push-1876255831be64124d0634fe7e328d262adde7c1.idx
                       to
                       repo/repo11.git/objects/pack/p4gc-push-1876255831be64124d0634fe7e328d262adde7c1.idx
pid 3252 gconn-user@gconn-ubuntu16 10.5.10.224 [unknown/v82] 'user-repo -o //repo/repo11.git'
```

SSH CLONE a REPO from the Helix4Git Connector

Requirements:

- User has the permission to ‘read’

  p4 grant-permission –u vkanczes –d repo –p read

- Gconn-user has admin permission to the graph depot

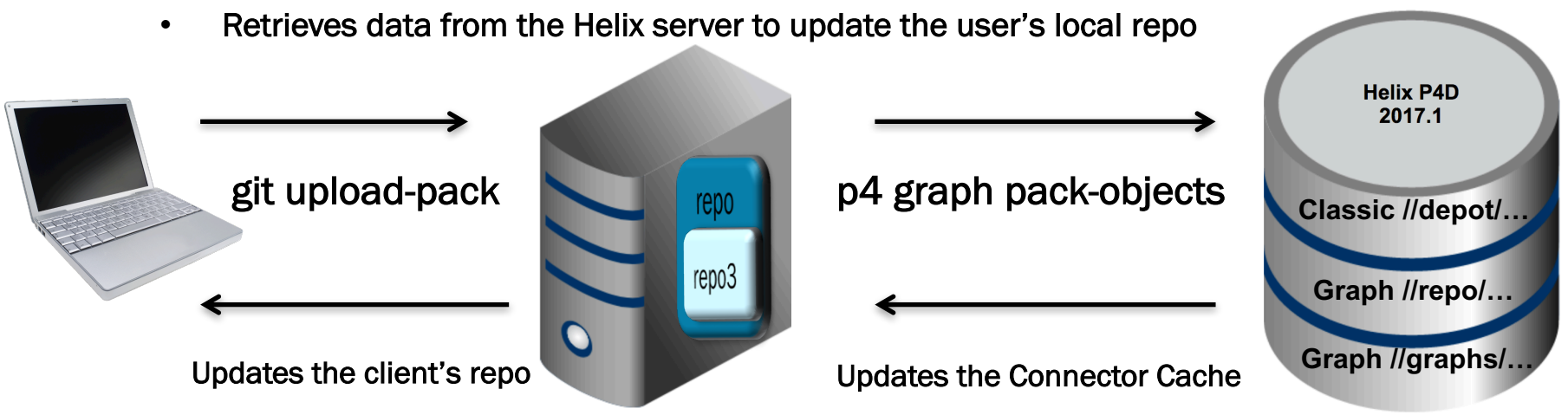
  p4 grant-permission –u gconn-user –d repo –p admin

- User’s SSH key has been loaded in Helix and synced to the Connector

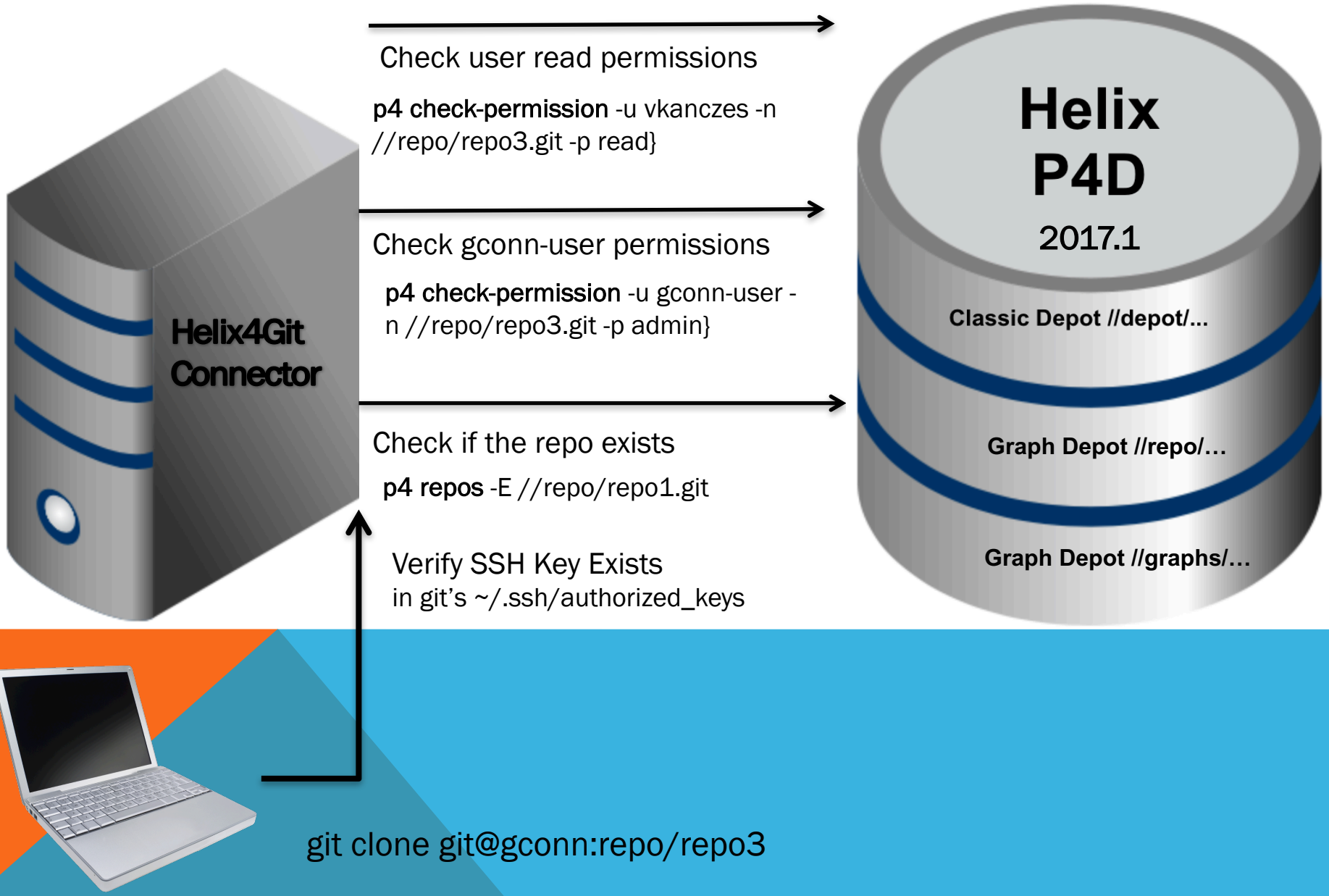
GIT -> GCONN -> GIT-REMOTE-P4GC -> GCONN -> GIT

git pull | clone

- Retrieves data from the Helix server to update the user's local repo



HELIX4GIT CONNECTOR AUTHENTICATION & PERMISSIONS



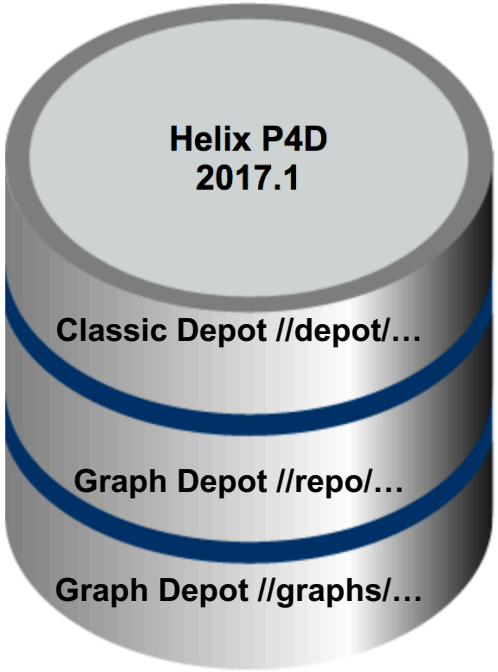
CLONE THE REPO – GCONN/GIT ACTIONS

Git client calls gconn
git-upload-pack 'repo/repo3
' --user=vkanczes

Syncs the repo

- **git-remote-p4gc** origin p4gc://perforce:1666/repo/repo3.git

- Updates Connector Cache



Sends the repo to the user's client

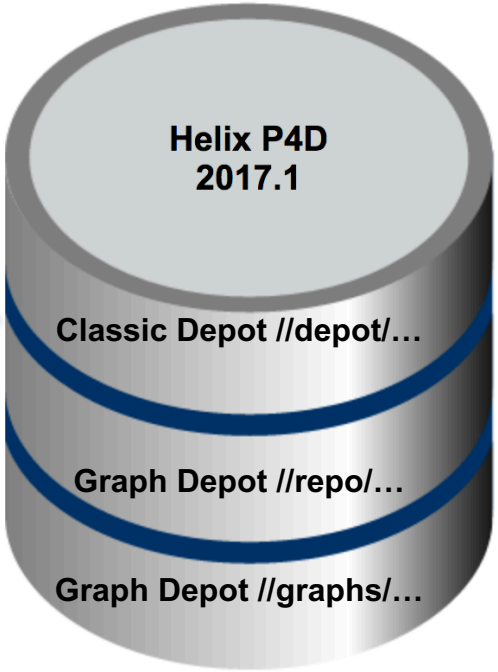
git clone git@gconn:repo/repo3

CLONE THE REPO - P4GC ACTIONS

git-remote-p4gc origin p4gc://perforce:1666/repo/repo3.git

Check and fetch repo branches

- **p4 graph show-ref** -n //repo/repo3.git



Call graph pack-objects

- **p4 graph pack-objects** -n //repo/repo3.git -o ./gconn.ebbdvY 11e80ee4cbcfdf7a21be5d66ee7ac 7252e368651
- git index-pack /opt/perforce/git-connector/repos/repo/repo3.git/objects/pack /pack-5522838bd2cb7b0993f33d71893942626e 013b13.pack

git clone git@gconn:repo/repo3

Gconn Log

```
2017/04/28 11:03:20 726708528 pid 23553: gconn git-upload-pack 'repo/repo3' --user=vkanczes
2017/04/28 11:03:20 727934143 pid 23553: {p4 -ztag -p 10.5.10.174:17100 -u gconn-user check-permission -u v
kanczes -n //repo/repo3.git -p read}
2017/04/28 11:03:20 734769623 pid 23553: SyncRepo(/opt/perforce/git-connector/repos/repo/repo3.git)
2017/04/28 11:03:20 735227424 pid 23553: {p4 -ztag -p 10.5.10.174:17100 -u gconn-user check-permission -u g
conn-user -n //repo/repo3.git -p admin}
2017/04/28 11:03:20 741797498 pid 23553: {p4 -ztag -p 10.5.10.174:17100 -u gconn-user repos -E //repo/repo3
.git}
2017/04/28 11:03:20 769106245 pid 23553: system(/bin/git init --bare repo/repo3.git > /dev/null) succeeded
2017/04/28 11:03:20 774657852 pid 23553: Configured remote for port 10.5.10.174:17100
2017/04/28 11:03:20 774775857 pid 23553: installed pre-receive hook
2017/04/28 11:03:20 774869107 pid 23553: installed post-receive hook
2017/04/28 11:03:20 979987262 pid 23553: Launching /bin/git-upload-pack repo/repo3
2017/04/28 11:03:21 246631762 pid 23553: RunCmd(/bin/git-upload-pack repo/repo3) succeeded
```

P4GC Log

```
2017/04/28 11:03:20 781949042 pid 23564: git-remote-p4gc origin p4gc://perforce:1666/repo/repo3.git
2017/04/28 11:03:20 782193331 pid 23564: >>> push
2017/04/28 11:03:20 782231574 pid 23564: >>> fetch
2017/04/28 11:03:20 783376730 pid 23564: p4 -ztag -p perforce:1666 -u gconn-user graph show-ref -n //repo/r
epo3.git}
2017/04/28 11:03:20 790986780 pid 23564: {read fetch} <<< fetch 11e80ee4cbcfdf7a21be5d66ee7ac7252e368651 re
fs/heads/master
2017/04/28 11:03:20 791030773 pid 23564: MkDir, /opt/perforce/git-connector/repos/repo/repo3.git
2017/04/28 11:03:20 791092115 pid 23564: MkDir, /opt/perforce/git-connector/repos/repo/repo3.git/objects
2017/04/28 11:03:20 791121179 pid 23564: MkDir, /opt/perforce/git-connector/repos/repo/repo3.git/objects/pa
ck
2017/04/28 11:03:20 791261031 pid 23564: cmd="git show-ref"
2017/04/28 11:03:20 792889895 pid 23564: {p4 -ztag -p perforce:1666 -u gconn-user graph pack-objects -n //r
epo/repo3.git -o ./gconn.ebbdvY 11e80ee4cbcfdf7a21be5d66ee7ac7252e368651}
2017/04/28 11:03:20 860137139 pid 23564: mv './gconn.ebbdvY/pack-5522838bd2cb7b0993f33d71893942626e013b13.p
ack' '/opt/perforce/git-connector/repos/repo/repo3.git/objects/pack/pack-5522838bd2cb7b0993f33d71893942626e
013b13.pack'
2017/04/28 11:03:20 860212378 pid 23564: cmd="git index-pack /opt/perforce/git-connector/repos/repo/repo3.g
it/objects/pack/pack-5522838bd2cb7b0993f33d71893942626e013b13.pack"
2017/04/28 11:03:20 973860881 pid 23564: {git index-pack} <<< 860476462529f3d9466c6e1a174115275a76e229
2017/04/28 11:03:20 973944135 pid 23564: >>> lock pack-5522838bd2cb7b0993f33d71893942626e013b13.pack
```

Special Client Connector Commands

- git clone git@Connector:@help
- git clone git@Connector:@info
- git clone git@Connector:@list
- git clone git@Connector:@defaultbranch:depot/repo
- git clone git@Connector:@defaultbranch:depot/repo=
- git clone git@Connector:@defaultbranch:depot/repo=branch

Special Connector Commands

```
git clone git@Connector:@list
Cloning into '@list'...
graphDepots/repo12.git | Automatically created by the initial push.
repo/grepo1.git | DefaultBranch=refs/heads/main | Automatically created by the initial push.
repo/grepo2.git | Created by super.
repo/grepo3.git | Automatically created by the initial push.
repo/http-mirror-test-14.git | Mirror of http://gitlabee.das.perforce.com/vkanczes/http-mirror-test-14.git
repo/mark.git | DefaultBranch=main | Automatically created by the initial push.
repo/projectB.git | Automatically created by the initial push.
repo/repo1.git | Automatically created by the initial push.
repo/repo11.git | Automatically created by the initial push.
repo/repo17.git | Automatically created by the initial push.
repo/repo2.git | Automatically created by the initial push.
repo/repo3.git | Mirror of http://gitlabee.das.perforce.com/vkanczes/repo3.git
repo/repo4.git | Mirror of http://vkanczes-gitlab-dr.das.perforce.com/vkanczes/repo4.git
repo/repo5.git | Automatically created by the initial push.
repo/repo6.git | DefaultBranch=refs/heads/main | Created by gconn-user.
repo/repo7.git | Automatically created by the initial push.
repo/repox.git | Created by super.
repo/ssh-mirror-test-14.git | Mirror of git@gitlabee.das.perforce.com:vkanczes/ssh-mirror-test-14.git
repo/ssh-test.git | Mirror of git@gitlabee.das.perforce.com:vkanczes/ssh-test.git
repo/ssh-test3.git | Mirror of git@gitlabee.das.perforce.com:vkanczes/ssh-test3.git
repos/grepo1.git | Created by super.
repos/httpd.git | DefaultBranch=trunk | Created by super.
repos/repo11.git | Automatically created by the initial push.
repos/repo12.git | DefaultBranch=main | Automatically created by the initial push.
repos/repo4.git | Automatically created by the initial push.
repos/test-mirror1.git | Mirror of http://gitlabee.das.perforce.com/vkanczes/test-mirror1.git
repos/test2.git | Mirror of http://vkanczes:password11@gitlabee.das.perforce.com/vkanczes/test2.git
repos/test3.git | Mirror of git@gitlabee.das.perforce.com:vkanczes/test3.git
repos/test4.git | Mirror of git@gitlabee.das.perforce.com:vkanczes/test4.git
repos/test5.git | Mirror of git@gitlabee.das.perforce.com:vkanczes/test5.git
repos/thomas.git | DefaultBranch=refs/heads/dev | Automatically created by the initial push.
repos/thomas2.git | DefaultBranch=dev | Automatically created by the initial push.
```

Special Connector Default Branch Commands

```
# A Perforce Repo Specification.
#
#  Name:          The name of this repo.
#  Owner:         The user who created this repo.
#  Created:       The date this specification was created.
#  Pushed:        The date of the last 'push' to this repo.
#  Description: A short description of the remote server (optional).
#  MirroredFrom:  Upstream URL that this repo is mirrored (readonly) from.
#  DefaultBranch: The default branch to clone (eg, "refs/heads/trunk" or
#                 "trunk"); it must begin with "refs/" if git is to use it.
#
#                    See 'p4 help repo' for detailed information.

Repo:    //repo/apex

Owner:   vkanczes

Created:         2017/04/06 10:29:14

Pushed: 2017/04/06 10:29:14

Description:
        Automatically created by the initial push.

DefaultBranch:  /refs/heads/trunk
```

Configuring the

Helix4Git

One-Way Mirroring using HTTPS

Requirements:

Migration From GitSwarm to GitLab is complete

Helix for Git Connector is installed and configured

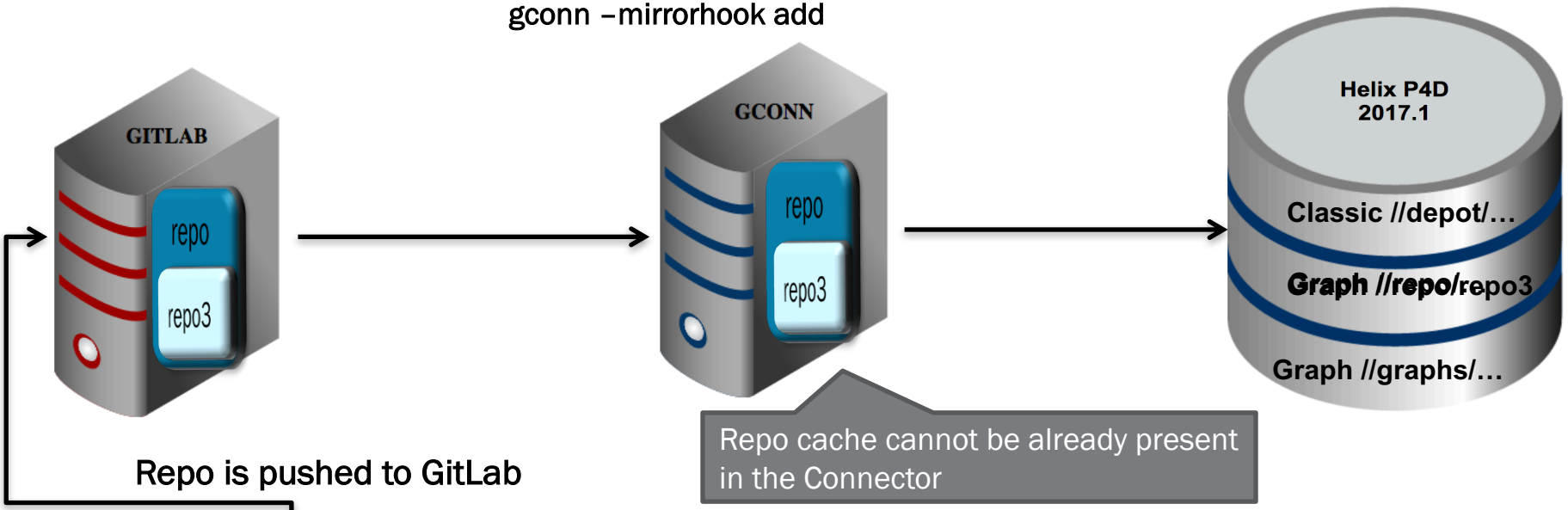
ONE-WAY MIRRORING CONFIGURATION

Once one-way mirroring is configured, the repo can only be modified/pushed to GitLab.

Webhook is added in GitLab

Mirror configured on Connector: gconn –mirrorhook add

Repo created with MirrorFrom: associated to the GitLab URL for the repository



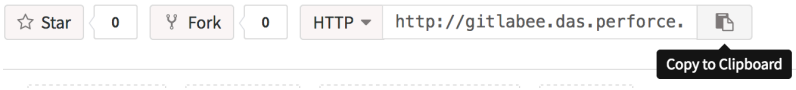
Repo cache cannot be already present in the Connector

Repo is pushed to GitLab

HELIX4GIT ONE-WAY MIRRORING HTTP CONFIGURATION

On the GitLab server, create an access token

Copy the URL from GitLab server



On the Connector, setup environment as root

- export GCONN_CONFIG=/opt/perforce/git-connector/gconn.conf
  - cd /opt/perforce/git-connector

Create the mirror hook

- ./bin/**gconn –mirrorhooks add** *repo/repo3 https://gitlab-ci-token:secret@gitlab.com/project/repo3.git*

Use the access token instead of username:password when adding the mirrorhook

- Save the secret token that is outputted after generating the webhook

Helix4Git Mirror Hook Configuration

```
Mirroring configured for:
Upstream repo: http://gitlabee.das.perforce.com/vkanczes/repo3.git
Downstream repo: //repo/repo3.git
Webhook URL: /mirrorhooks
Webhook secret token: 4f1a52b4-3ad9-4e7c-9cda-f764bc9cc2ad

You must still add the webhook using the URL and token indicated before updates will be mirrored.

opt/perforce/git-connector/repos/repo/repo3.git# cat .mirror.config
{
  "upstream_url": "http://gitlabee.das.perforce.com/vkanczes/repo3.git",
  "downstream_repo": "//repo/repo3.git",
  "secret_token": "4f1a52b4-3ad9-4e7c-9cda-f764bc9cc2ad",
  "status": "enabled"
}
```

HELIX4GIT ONE-WAY MIRRORING VALIDATION

```
/opt/perforce/git-connector/repos/repo/repo3.git# ls -al
drwxrwsr-x  7 root     gconn-auth 4096 May  4 16:02 .
drwxrwsr-x 21 www-data gconn-auth 4096 May 15 15:17 ..
drwxrwsr-x  2 root     gconn-auth 4096 Apr 27 16:32 branches
-rw-rw-r--  1 root     gconn-auth  222 Apr 28 15:09 config
-rw-rw-r--  1 root     gconn-auth   73 Apr 27 16:32 description
-rw-rw-r--  1 root     gconn-auth  132 May  4 16:02 FETCH_HEAD
-rw-rw-r--  1 www-data gconn-auth  131 May  4 16:02 fetch_log
-rw-rw-r--  1 root     gconn-auth   23 Apr 27 16:32 HEAD
drwxrwsr-x  2 root     gconn-auth 4096 Apr 27 16:32 hooks
drwxrwsr-x  2 root     gconn-auth 4096 Apr 27 16:32 info
-rw-rw-r--  1 root     gconn-auth  198 May  2 16:51 .mirror.config
-rw-rw-r--  1 root     gconn-auth    0 Apr 27 16:32 .mirror.lock
-rw-rw-r--  1 root     gconn-auth 1825 May  4 16:02 .mirror.log
drwxrwsr-x 25 root     gconn-auth 4096 May  4 16:02 objects
-rw-rw-r--  1 www-data gconn-auth  126 May  4 16:02 push_log
drwxrwsr-x  5 root     gconn-auth 4096 Apr 28 15:10 refs
root@gconn-ubuntu16:/opt/perforce/git-connector/repos/repo/repo3.git# head .mirror.log
sent request
daemon: started with pid=9758
received request
daemon: starting fetch from http://vkanczes-gitlab-dr.das.perforce.com/vkanczes/repo3.git
daemon: starting push to p4gc://gconn-user@10.5.10.174:17100/repo/repo3.git
daemon: request complete
no remaining requests
```

Configuring the

Helix4Git

One-Way Mirroring for SSH

Requirements:

Migration From GitSwarm to GitLab is complete

Helix for Git Connector is installed and configured

CentOS web-user: apache
Ubuntu web-user is www-data

HELIX4GIT ONE-WAY MIRRORING SSH CONFIGURATION

As root on the Connector, setup the web service user's SSH keys

- Create the /var/www/.ssh directory for www-data
- Change owner of the directory
  - chown www-data:gconn-auth /var/www/.ssh

Switch to the web service user
- su -s /bin/bash - www-data

Generate the ssh keys
- ssh-keygen -t rsa -b 4096 -C www-data@connector.com

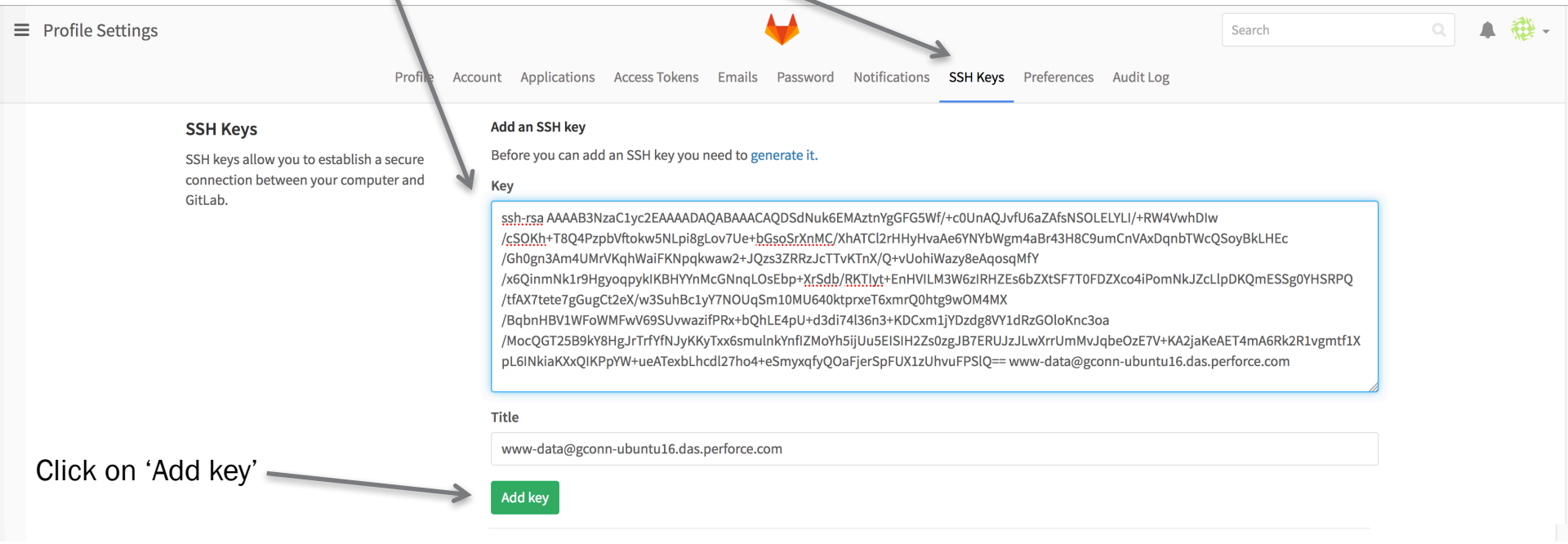
Copy the ssh public key to the GitLab
- add /var/www/.ssh/id_rsa.pub key to user who can clone and fetch from the repository.

HELIX4GIT ONE-WAY MIRRORING SSH CONFIGURATION

In GitLab go to Profile Settings
Click on SSH Keys

Be sure to add for a user who can clone/fetch from the repository.

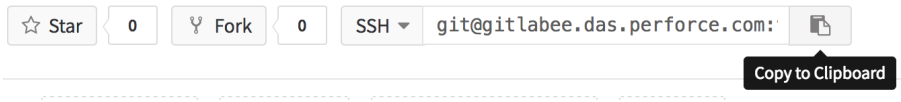
Add the /var/www/.ssh/id_rsa.pub public key



Click on ‘Add key’

HELIX4GIT ONE-WAY MIRRORING SSH CONFIGURATION

Copy the URL from the GitLab server



On the Connector, setup environment as web service user (www-data or apache)

- export GCONN_CONFIG=/opt/perforce/git-connector/gconn.conf

Create the mirror hook

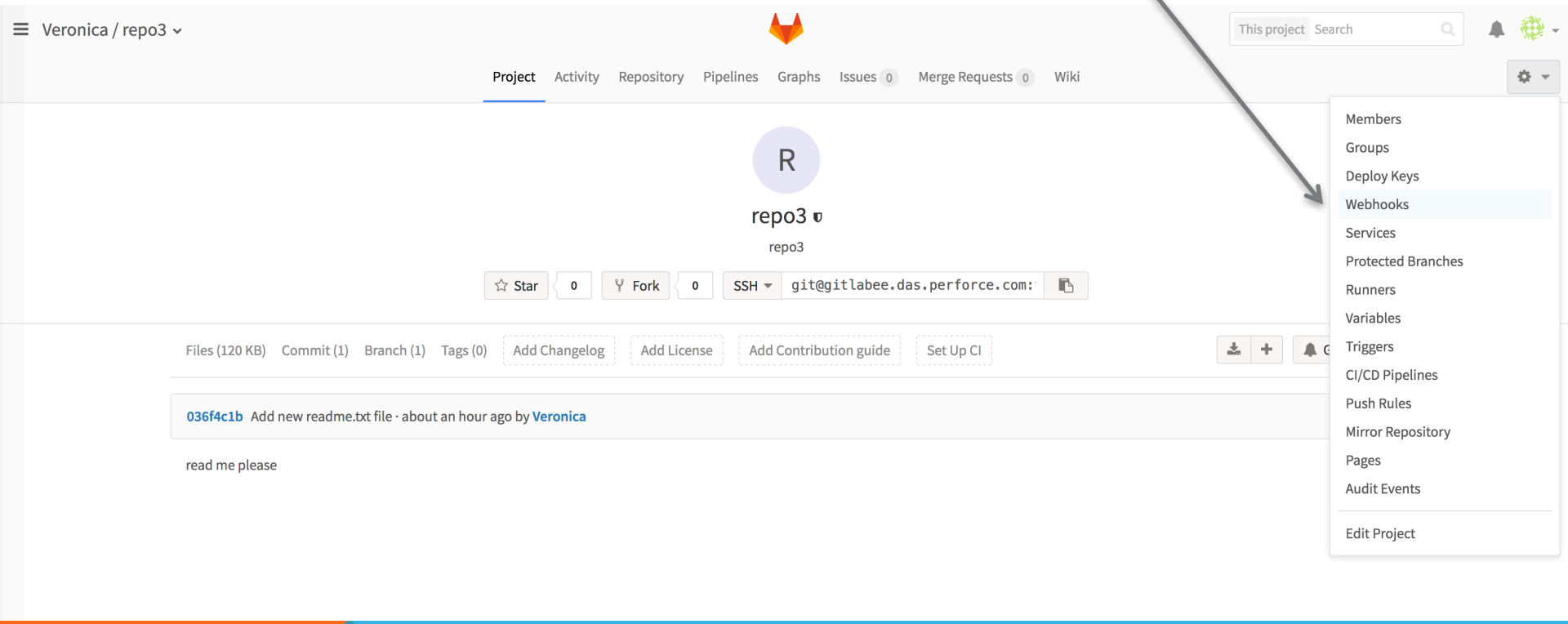
- ./bin/**gconn –mirrorhooks add** *repo/repo3* *git@gitlabServer.com/project/repo3.git*

- Save the secret token that is outputted after generating the webhook

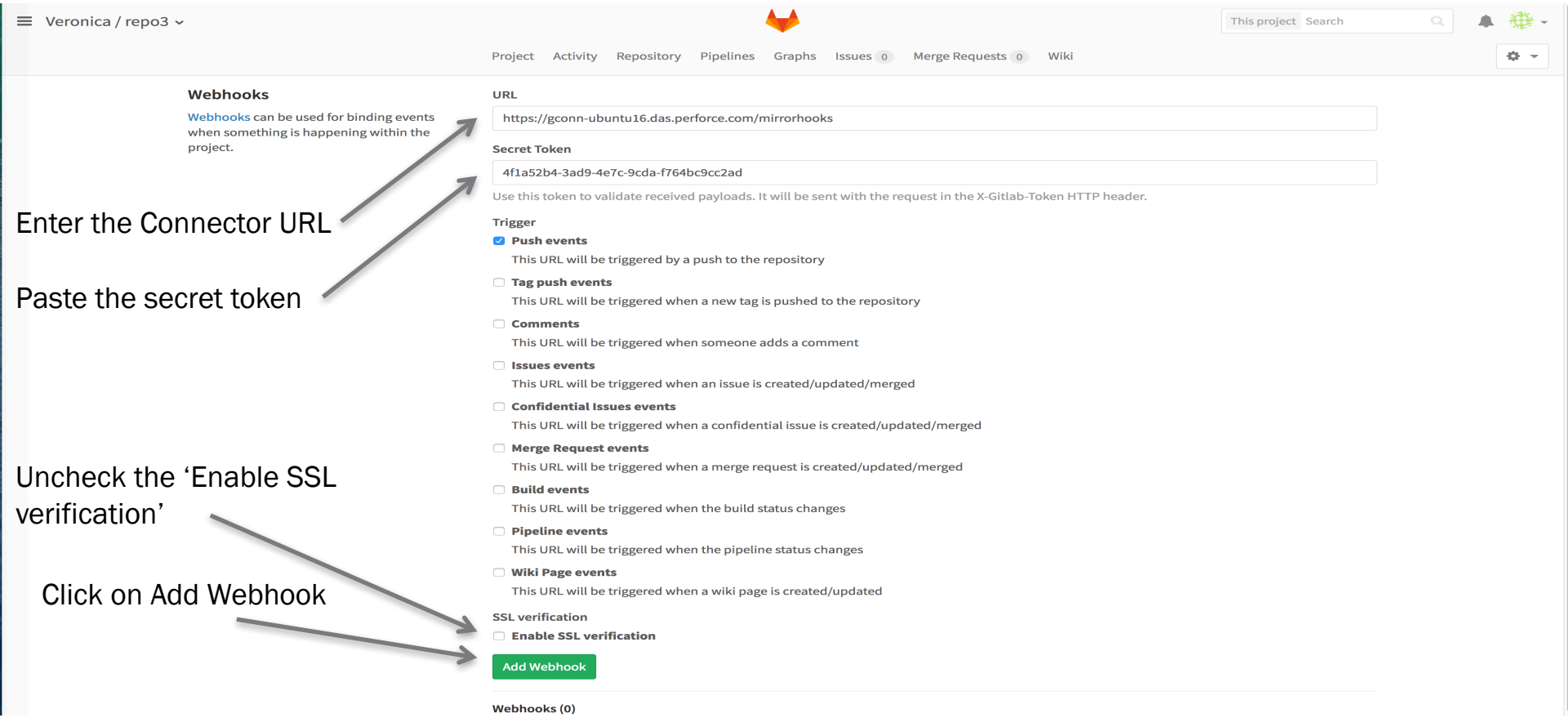
Configuring the Webhook in GitLab

HELIX4GIT ONE-WAY MIRRORING CONFIGURATION ON GITLAB

Choose webhooks from project drop-down



HELIX4GIT ONE-WAY MIRRORING CONFIGURATION ON GITLAB



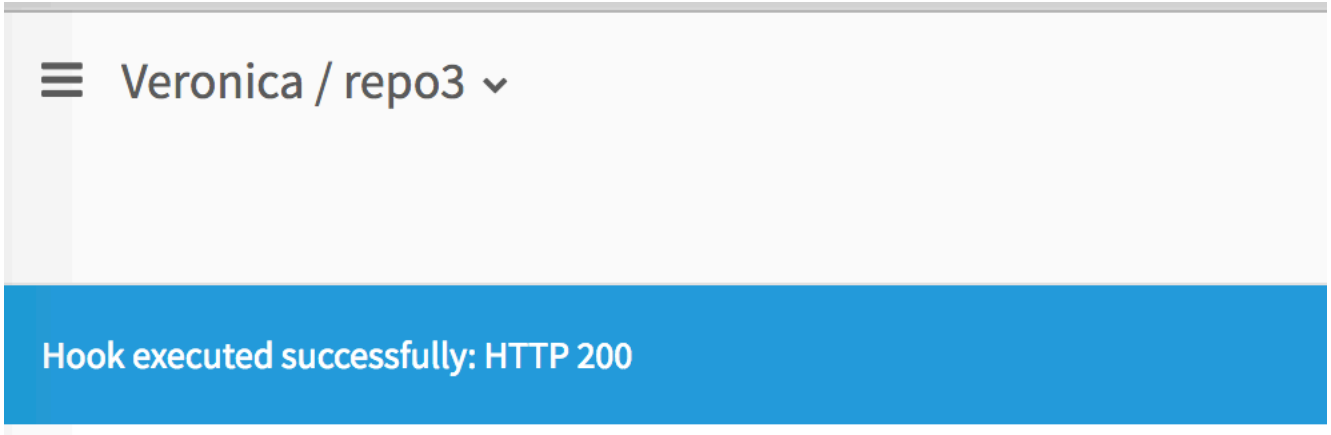
Enter the Connector URL

Paste the secret token

Uncheck the ‘Enable SSL verification’

Click on Add Webhook

HELIX4GIT ONE-WAY MIRRORING CONFIGURATION ON GITLAB



Click on ‘Test’ to verify it

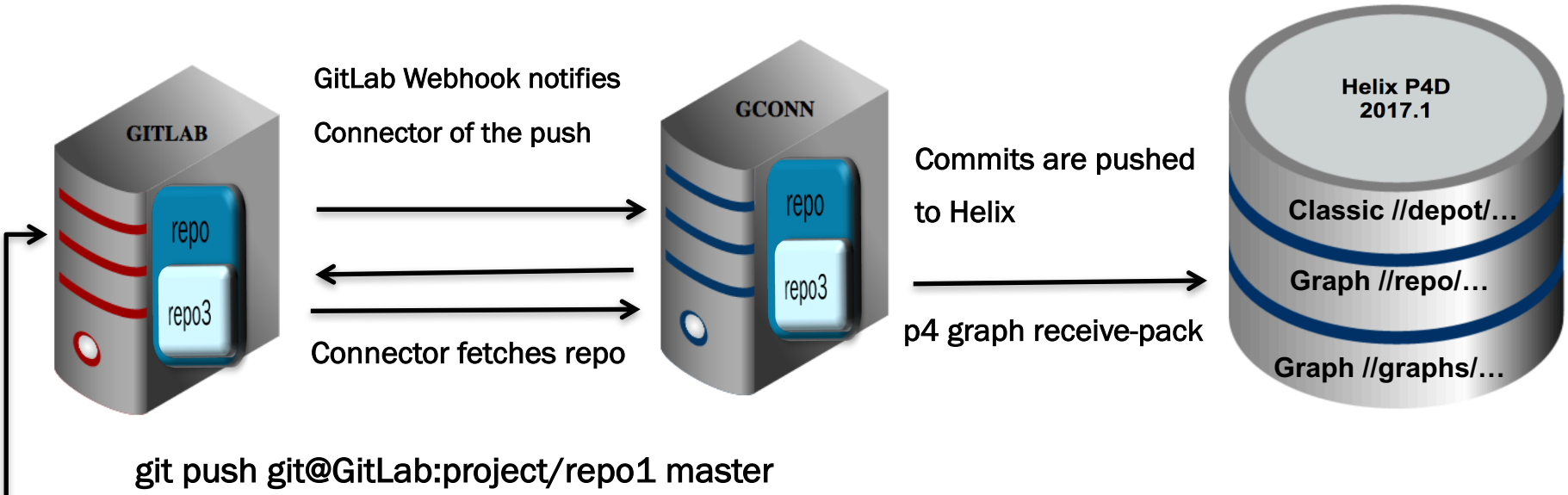
HELIX4GIT ONE-WAY MIRRORING VALIDATION

Edit the readme.txt in GitLab and commit

Check the file is in Helix

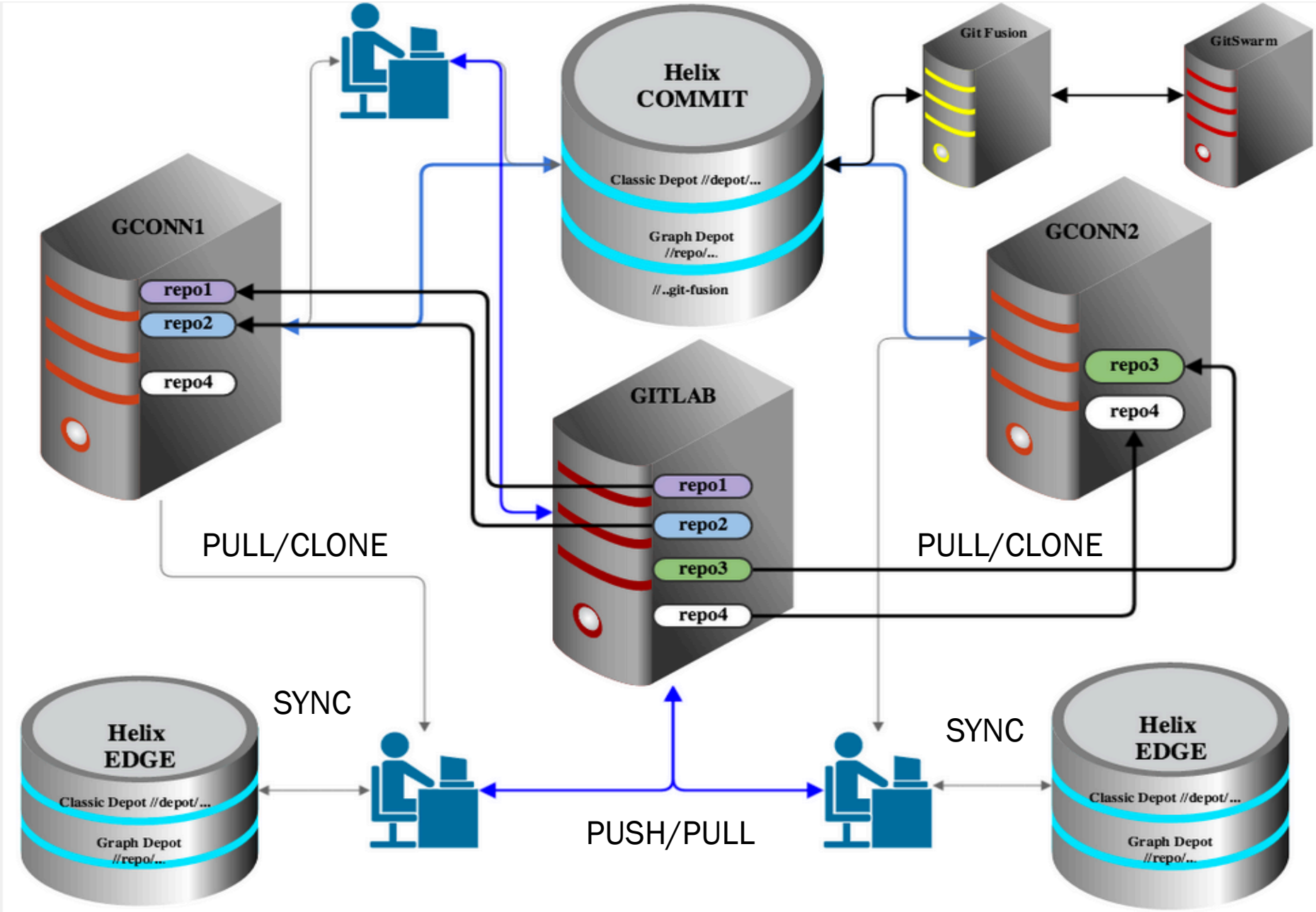
```
p4 sync
//repo/repo3/GitConnector.pptx - added as /opt/perforce/servers/17100/ws/repo/repo3/GitConnector.pptx
//repo/repo3/README.md - added as /opt/perforce/servers/17100/ws/repo/repo3/README.md


p4 graph log -n //repo/repo3 -m1
commit 11e80ee4cbcfdf7a21be5d66ee7ac7252e368651
Author: Veronica <vkanczes@perforce.com>
Date: 2017/04/27 16:42:37

Update README.md
```

ONE-WAY MIRRORING FROM GITLAB



Helix4Git Environments

Helix4Git Environment with One-Way Mirroring



This concludes the

Helix4Git Connector training session

Documentation

Helix4Git 2017.1

- Installation and Configuration
- One-Way Mirroring
- Troubleshooting

Helix Versioning Engine Administrator Guide: Triggers

P4 Command Reference

Helpful Links

First 20 Minutes with Helix4Git Connector

Technical Support Helix4Git Connector Quickstart – VMs

Helix4Git QuickStart Guide

Helix4Git Connector SSH Setup and Validation

Helix4Git Connector HTTPS Setup and Validation

Migration From GitFusion/GitSwarm to GitLab and Helix4Git

Migrating Repos from Git Fusion to Helix4Git

Helix4Git One-Way Mirroring with GitLab Using HTTPS

Helix4Git One-Way Mirroring with GitLab Using SSH

Confluence Helix4Git Support

Git Commands

Q&A