Advanced Helix Enterprise (P4D) Administration

These exercises follow on from the Introduction to Administration Exercises and assume the same initial setup and local test repositories, including changes to configuration carried out during that course. If your environment has been refreshed, your instructor can advise additional steps required prior to carrying out the following exercises.

Lab Set E1: Replication

Your objective for this exercise:

- Create a forwarding replica of your 1666 server (we will later convert this to an edge server).

1. Create a server spec called **`replica1`** and set the services to **`forwarding-replica`**.

```
p4 server Replica1

Set following line:

Services:  forwarding-replica
```

2. Create a user account of type **`service`** named **`replica1`**. Then create a group named **`service_users`** with a timeout value set to **`unlimited`**, and that has the **`replica1`** in its user list. Set the password for the **`replica1`** user to: **`Replicapass!`**

   Give the **`service_users`** user a 'super' access in the protection table of the master.

```
p4 user –f replica1
p4 passwd replica1
p4 group service_users

        Change Timeout: field to unlimited

        Add replica1 to Users:

p4 protect

        Add line:

        super group service_users  *  //...
```

3. Set up the replica configuration variables for our new replication server called "Replica1". You will need to set several configuration variables (Hint: create a local .bat file with notepad and edit it before running it – if copying and pasting from this document make sure that things like quotes are not smart-quotes!)

```
p4 configure set Replica1#P4TARGET=localhost:1666
p4 configure set Replica1#P4PORT=1999
p4 configure set Replica1#journalPrefix=C:\Replica1\checkpoints\p4_1
p4 configure set Replica1#db.replication=readonly
p4 configure set Replica1#lbr.replication=readonly
p4 configure set Replica1#serviceUser=replica1
```



```
p4 configure set Replica1#startup.1="pull -i 1"
p4 configure set Replica1#startup.2="pull -u -i 1"
p4 configure set Replica1#P4LOG=c:\replica1\logs\log
p4 configure set Replica1#P4JOURNAL=c:\replica1\logs\journal
p4 configure set Replica1#server.depot.root=c:\replica1\depots
```

4. Checkpoint your live server

```
p4 admin checkpoint -Z
```

5. Create a new root directory for your replica here: **`C:\Replica1`**

   a. Copy p4d.exe into this directory
   b. Restore the checkpoint in this directory
   c. Create a '**`checkpoint`**' subdirectory to hold checkpoints.
   d. Create a '**`logs`**' subdirectory for the journal and log.
   e. Create a '**`depots`**' subdirectory for the replica depot files.

```
mkdir C:\Replica1
mkdir C:\Replica1\checkpoints
mkdir C:\Replica1\logs
mkdir C:\Replica1\depots
cd c:\Replica1
p4d -r c:\Replica1 -jr -z C:\P4Training\checkpoints\p4_1.ckp.n.gz
```

where n is the latest available checkpoint in C:\P4Training.
If checkpoint was compressed (in 4 above), add the -z parameter and a '.gz' extension to the checkpoint filename

6. Give the replica server a serverid of Replica1 (using p4d –Xd)

```
p4d -r C:\Replica1 -xD Replica1
```

7. Start the replica server (on the command line) – you only need to specify the root as it should read other variables include PORT from the configurables.

```
p4d -r C:\Replica1
```

8. Log the replica (service) user **`replica1`** into the **master** server to commence replication. (As the master and replica will typically be on separate hardware, this normally has to happen from the replica machine).

```
p4 -p 1666 -u replica1 login
```


9. As user **bruno,** check on the master server side that the replica has started the replica service by checking the log (hint: **p4 logtail**).

```
p4 -p 1666 logtail
```

10. Check on the replica that your pull processes are running. You will need to log in to the replica first. Hint: use the `p4 monitor show` command on the replica and look for the 'p4 pull' processes.

```
p4 -p 1999 monitor show -l
p4 -p 1999 pull -lj
```

11. Update some data. For example, change/add the Description of the **bruno_ws** workspace spec on the master server, and then print out the same spec on the replica server.

```
p4 -p 1666 client bruno_ws

p4 -p 1999 print //specs/client/bruno_ws.p4s
        or
p4 -p 1999 client -o bruno_ws
```

12. Test that you can sync, edit and submit any files to this replica server since it is a forwarding replica.

```
p4 -p 1999 -c bruno_ws sync
p4 -p 1999 -c bruno_ws sync //depot/Jam/MAIN/...
p4 -p 1999 -c bruno_ws edit //depot/Jam/MAIN/src/jam.c
```

13. Check to see if there are any (version) files in the depot sub-directory under **c:\Replica1**

```
dir /ad c:\replica1
```

14. Run verify with **-qt** flag on **//depot/jam/MAIN/...** to schedule the pulling of all appropriate versioned files. Wait a few moments, and then check that these files have now been created in your replica directory (either using Windows Explorer or cmd commands)

```
p4 -p 1999 pull -ls
p4 -p 1999 verify -qt //depot/jam/MAIN/...
dir /s/b c:\replica1\depot
```

(In a real installation you would copy all the versioned files from your master server to the replica server - not use **p4 verify -qt**)

Lab Set E2: Edge Server

Your objective for this exercise:

- Turn your forwarding replica into an edge server and show that you can create a new edge server specific workspace

15. Check that your master server has server ID of master (hint: use “p4 serverid”):

Check current serverid:

```
p4 serverid
```

Set serverid (if necessary) and then recheck:

```
p4 serverid Master
p4 serverid
```

16. Change the Master server specification (or create one) to make it a commit-server (see P4Dist manual)

```
p4 server Master
```

Set following line:

```
Services:  commit-server
```

17. Change the Replica1 server specification to make it an edge-server (see P4Dist manual)

```
p4 server Replica1
```

Set following line:

```
Services:  edge-server
```

18. Stop the replica, and remove the necessary db.* files as specified in the Distributed Guide (see P4Dist manual).

```
p4 -p 1999 admin stop
cd c:\Replica1
del db.have db.working db.resolve db.locks db.revsh
   db.workingx db.resolvex
```

19. Start the replica (like you did in previous Labs)

```
p4d -r C:\Replica1
```


20. Create a new workspace **build-ws** while talking to the replica specifying appropriate root and view. Check that the ServerID field is set after you have created the workspace.

```
p4 -p 1999 client build-ws

Set Root and View fields, and save.

p4 -p 1999 client –o build-ws

Check ServerID field is set to Replica1.
```

21. Sync your new workspace

```
p4 -p 1999 -c build-ws sync
```

22. Check if the workspace is visible on the main server and if you can see which files are synced within it by talking to the main server.

```
p4 -p 1666 clients
p4 -p 1666 files @build-ws
```

23. Test that you can now edit and submit files when communicating with the replica server in your new workspace.

```
p4 -p 1999 -c build-ws sync //depot/Jam/MAIN/…
p4 -p 1999 -c build-ws edit //depot/Jam/MAIN/src/jam.c
```


Lab Set E3: Advanced Maintenance

24. Create an archive depot called *archive.*

```
p4 depot archive

Set The 'Type:' value to: archive
```

25. The files in //depot/Misc/manuals/… are not used at the moment. Archive them in your new archive depot.

```
p4 archive -D archive //depot/Misc/manuals/...
```

26. We need //depot/Misc/manuals/triggers.doc again. Restore it from your archive depot.

```
p4 restore -D archive  //depot/Misc/manuals/triggers.doc
```


Lab Set E4: Offline Checkpoint

Your objective for this exercise:

- Create an offline checkpoint together with rotating the live journal
- Optionally restore the database files and replace the live server db.*

1. Ensure your current environment is pointing at the 1666 server - use “set” or ”p4 set” or “p4 set P4CONFIG” to ensure this.

2. Checkpoint your server on port 1666

```
p4 admin checkpoint -Z

    or

p4d -r C:\P4Training -jc -Z
```

   Note the number of the resulting checkpoint to use below (NNN)

3. Create an offline directory

```
mkdir C:\P4Training\offline_db
```

4. Recover from the checkpoint into that offline directory

```
p4d -r C:\P4Training\offline_db -jr -z
C:\P4Training\checkpoints\p4_1.ckp.NNN.gz
```

Where NNN is the number of the checkpoint previously created.

5. Rotate the journal on the 1666 server (Hint: don’t checkpoint it - so use “**-jj**” rather than “**-jc**” option to p4d) - and note the number of the resulting rotated journal

```
p4d -r C:\P4Training -J c:\p4training\logs\journal -jj
```

6. Apply that journal to your offline database

```
p4d -r C:\P4Training\offline_db -jr
C:\P4Training\checkpoints\p4_1.jnl.NNN
```

Where NNN is the number of the journal previously created.

7. Checkpoint the offline database - but without incrementing the journal counter (Hint: use "**-jd**" instead of "**-jc**")

```
p4d -r C:\P4Training\offline_db -jd -z
c:\p4training\checkpoints\p4_1.ckp.NNN+1.gz
```

Where NNN is the number of the journal previously created.

8. Backup the db.* files from your offline database and then recreate them from the checkpoint.

```
cd C:\P4Training\offline_db
mkdir save
move db.* save
p4d -r . -jr -z c:\p4training\checkpoints\p4_1.ckp.NNN+1.gz
```

9. Stop your server on 1666

```
p4 admin stop
```

10. Backup the db.* files from **C:\P4Training** and move the db.* files from your offline directory to **C:\P4Training**

```
cd C:\P4Training
mkdir save
move db.* save
move C:\P4Training\offline_db\db.* .
```

11. Recover the live journal into the **C:\P4Training** directory

```
p4d -r . -jr c:\p4training\logs\journal
```

12. Restart the 1666 server and check that everything looks OK.

```
server.bat
```


Lab Set E5: Broker

Your objective for this exercise:

- Create a broker, and configure it to disallow the 'p4 obliterate' command entirely.

13. Check and if necessary set the security level for your server to 3.

```
p4 configure show security
p4 configure set security=3
```

14. Set up and run a broker.

    Then, create a folder to store the Broker executables, configuration and logfiles:

```
MKDIR C:\P4Broker
COPY C:\P4Training\p4broker.exe C:\P4Broker\p4broker.exe
CD C:\P4Broker
```

    Generate the default configuration file, and adjust:

```
p4broker -C > p4broker.cfg
```

    Use Notepad to edit p4broker.cfg. Change these values in the file:

```
directory          = C:\P4Broker;
```

    Review other settings. Then, add this block at the end of the file:

```
# Forbid the use of 'p4 obliterate'.
command: obliterate
{
    action  = reject;
    message = "p4 obliterate disabled. This will be reported!";
}
```

    Next, start, and test the broker:

```
p4broker -c c:\p4broker\p4broker.cfg

open another command window and run:
p4 -p 1667 info
```

15. Accessing the **broker** (port 1667), try to obliterate the file

    //depot/Misc/Marketing/valuations.xls

    Note: it is **always** good practice to initially run the obliterate command in '*report*' mode

```
p4 -p 1667 obliterate //depot/Misc/Marketing/valuation.xls

Note the error message (as entered in the broker.cfg file)
```


16. Now try obliterating the above file directly in the server;

Initially in report mode:

```
p4 -p 1666 obliterate //depot/Misc/Marketing/valuation.xls
```

then

```
p4 -p 1666 obliterate -y //depot/Misc/Marketing/valuation.xls
```


Lab Set E6: Security

Your objective for this exercise:

- Put some basic commercial security measures in place. This lab requires the command line interface.

17. Check and if necessary set the security level to 3.

```
p4 configure show security
p4 configure set security=3
```

18. Make it so new user accounts can be created only by super users.

```
p4 configure set dm.user.noautocreate=2
```

19. Make the default changelist type be ‘restricted’, rather than ‘public.’

```
p4 configure set defaultChangeType=restricted
```

20. Make it required for users to login to run the “users” command.

```
p4 configure set run.users.authorize=1
```


Lab Set E7: Structured Logging

Your objective for this exercise:

- Enable all structured logs. This isn't something you would normally do on a production server; instead you would select only those logs you intend to monitor.

21. Enable recording of all seven structured log types.

```
p4 configure set serverlog.file.1=all.csv
p4 configure set serverlog.file.2=commands.csv
p4 configure set serverlog.file.3=errors.csv
p4 configure set serverlog.file.4=audit.csv
p4 configure set serverlog.file.5=track.csv
p4 configure set serverlog.file.6=user.csv
p4 configure set serverlog.file.7=events.csv
p4 configure set serverlog.file.8=integrity.csv
```

22. Configure to keep no more than 5 iterations of logs.

```
p4 configure set serverlog.retain.1=5
p4 configure set serverlog.retain.2=5
p4 configure set serverlog.retain.3=5
p4 configure set serverlog.retain.4=5
p4 configure set serverlog.retain.5=5
p4 configure set serverlog.retain.6=5
p4 configure set serverlog.retain.7=5
p4 configure set serverlog.retain.8=5
```

23. Rotate the logs manually once

```
CD C:\P4Training
DIR /OD *.csv
p4 logrotate
DIR /OD *.csv
```

See how the logs are rotated.

24. Use the 'p4 logappend' command to write something to the user log.

```
p4 logappend -a This is my message to the log.
TYPE user.csv
```


Lab Set E8: Simple Admin Command Automation - Optional

Your objective for this exercise:

- Try a few script-like commands using the power of the p4 command line.

25. Print out a list of usernames and email addresses.

    Windows:

    ```
    for /f "tokens=1,2" %a in ('p4 users') do @echo %a %b
    ```

    Unix (bash):

    ```
    p4 users | cut -d" " -f1,2 | while read c; do echo $c; done
    ```

26. Create a client workspace (which is a form command and will normally start up and editor) from a text file.

    ```
    p4 -c new-ws client -o | p4 client -i
    ```

    Check that the client workspace **new-ws** has been successfully created.

27. Create a new group all_users and via the command line add all users to that group.

    Save a basic group specification file to g.txt

    ```
    p4 group -o all-users > g.txt
    ```

    Append a list of all current Helix Server users to that file (remembering to add at least one space before each username):

    ```
    for /f "tokens=1" %a in ('p4 users') do @echo    %a >> g.txt
    ```

    Create the group by inputting from the file.

    ```
    p4 group -i < g.txt
    ```

    Check that the group has been successfully created.

Congratulations! You have completed the Advanced Helix Enterprise Training Course. We hope this course material has prepared you to use Helix Enterprise with confidence and ease.

