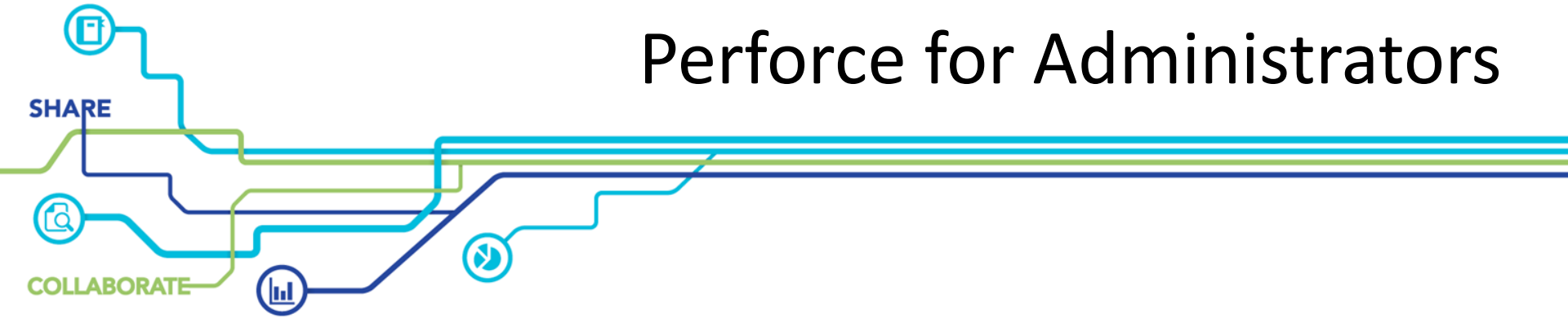
PERFORCE
Version everything.

Introduction to Perforce for Administrators



Introduction

PERFORCE
Version everything.

- Introductions
- Class Schedule
- GUI vs. CLI
- P4Admin Demos
- About the Exercises

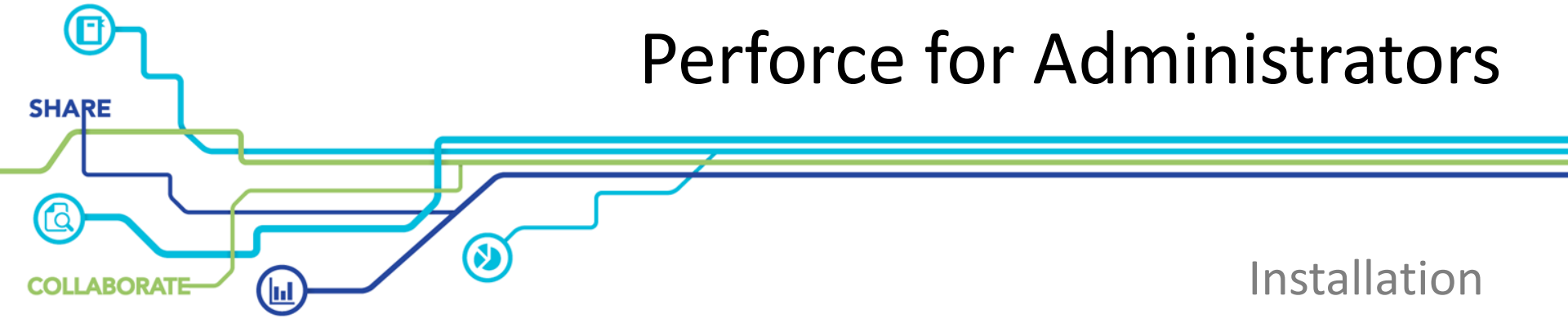
Course Contents

PERFORCE
Version everything.

- Installation
- Setup
- Backup and Recovery
- LDAP / AD
- Protections
- Depots
- Email Reviews
- Perforce Jobspec
- Maintenance
- Monitoring
- Trigger Capabilities Overview
- Other server Products and Resources

PERFORCE
Version everything.

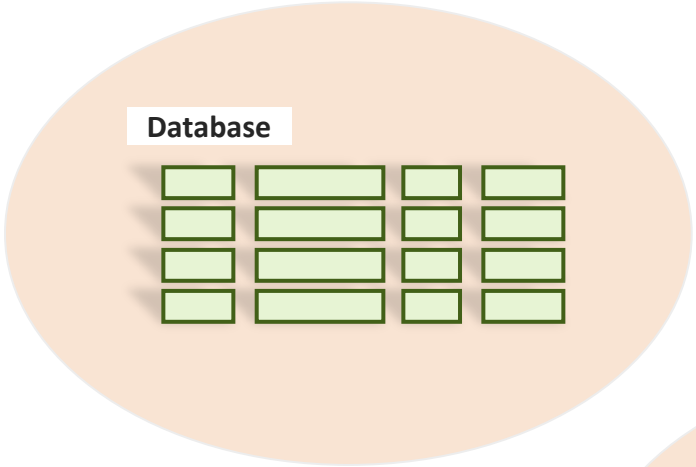
Introduction to Perforce for Administrators

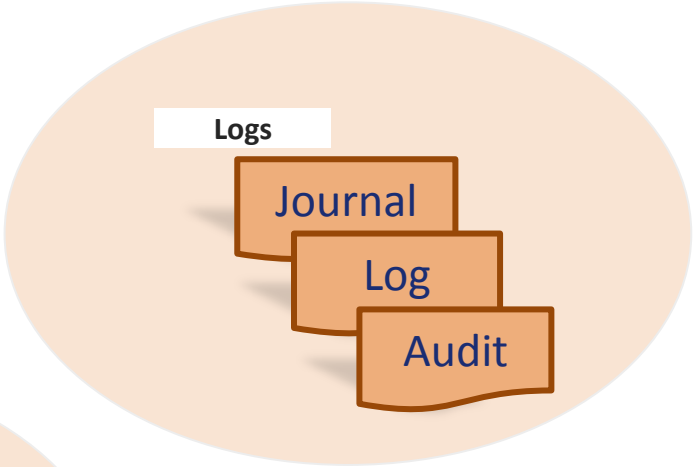


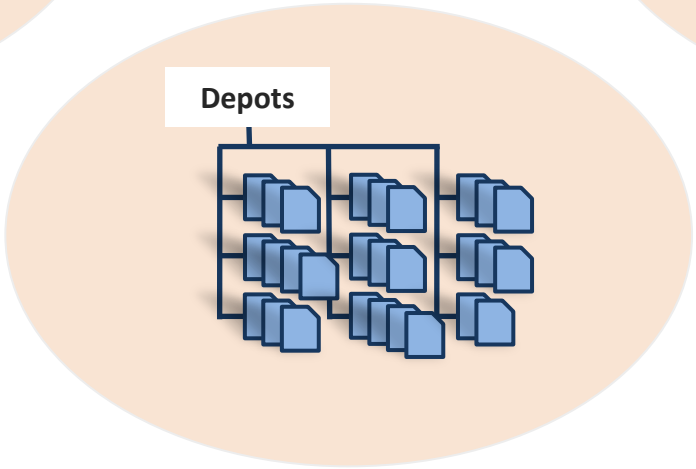
Installation

Perforce Architecture

PERFORCE
Version everything.







Volume layout for Perforce

PERFORCE
Version everything.

Volume	Sample Location	Contents	Performance Considerations
Metadata	/metadata	P4ROOT with database files.	Optimize I/O for *random* read/write. Vulnerable to high latency and low bandwidth.
Logs	/logs	Server logs and active journal.	High performance demands, but only for long *sequential* writes.
Depot Data	/depotdata	Archive files.	Typically more *sequential* read/write. Potentially very large amount of data.

Server Hardware Requirements

PERFORCE
Version everything.

- RAM:
  - Ideally, enough to cache all of your db.* files.
  - “If it has a slot, fill it” (modern Linux only)
- Disk Space ~ 0.5 KB/file * #files * #users
  - Metadata only
- CPU ~ Use the fastest processor available
  - It improves performance even at small sites

Configuring P4D, the Perforce Server

PERFORCE
Version everything.

- Typical server environment variables

Variable	Description	Command line argument	Default Value
P4ROOT	Perforce server database directory	-r	**Always Specify**
P4PORT	Port the Server listens on, with optional prefixes (e.g. “ssl:”, “localhost:”)	-p	1666
P4JOURNAL	Journal file (fully qualified path)	-J	$P4ROOT/journal
P4LOG	Server log (fully qualified path)	-L	$P4ROOT/log
P4AUDIT	File access audit log (optional, fully qualified path)	-A	none

Starting on Unix

Server runs continuously in background using ...

- ... environment variables:

```
export P4PORT=1666
export P4ROOT=/p4/1/root

p4d -q -d
```

- ... command-line arguments (recommended):

```
p4d -r /p4/1/root -p 1666 -q -d
```

-d : start in the background
-q : quiet mode

See also '`p4 configure`' and SSL encryption

Case Sensitivity

PERFORCE
Version everything.

- By default:
  - Server runs *case-sensitive* on Unix
  - Server runs *case-insensitive* on Windows/Mac

- Case-insensitive Server on Unix with -C1
  - Create the Server in case-insensitive mode
  - Needs to be set with every p4d command

```
p4d -C1 -r /p4/1/root ...
```

Starting on Windows

PERFORCE
Version everything.

- Installed as a Windows service using `p4s.exe` (just a copy of p4d.exe)
- By default, a Windows service named “Perforce” is created.
- Start/Stop with standard Services applet, or “net start” / “net stop” commands.

- Can run p4d in command prompt window
  - Admin privileges not required when started like this
  - Suitable for creating a test instance

```
p4d -r d:\p4\test -p 1667 -L log
```

Windows Service Configuration (Optional)

PERFORCE
Version everything.

- Configure non-default value for service name, and/or multiple instances.
- Set values for key P4D variables (P4PORT, P4ROOT, P4JOURNAL, P4LOG, etc.) for all instances. ***The 'p4 set' command is your friend!***

```
p4 set -S service_name [P4VAR=value]

p4 set -S p4_5 P4PORT=5666
p4 set -S p4_5 P4ROOT=F:\p4\5\root
p4 set -S p4_5 P4JOURNAL=J:\p4\5\logs\journal
p4 set -S p4_5 P4LOG=J:\p4\5\logs\log
p4 set -S p4_5 P4AUDIT=J:\p4\5\logs\audit
```

- Give each instance its own copy of p4s.exe, and configure the service:

```
svcinst.exe create -n p4_5 -e F:\p4\5\bin\p4s.exe -a
```

Environment Variables / P4CONFIG

PERFORCE
Version everything.

- Can use (command-line) environment variables:

```
set P4PORT=5666                    (Windows)
export P4ROOT=/home/p4/root        (Linux)
```

- **P4CONFIG** environment variable can be used to identify a file containing variable settings:

```
cd <…required folder/directory…>
set P4CONFIG=.p4config.txt
echo "P4PORT=5666" > %P4CONFIG%
echo "P4USER=bruno" >> %P4CONFIG%
```

License Files

PERFORCE
Version everything.

- No license file needed for 20 users/20 workspaces, or < 1000 files
- `license` file located in P4ROOT
- `p4 license [-o|-i|-u]`
  - Option `-o` prints out the existing license
  - Option `-i` imports a new license from stdin
  - Option `-u` reports current usage
- Commercial vs. Temporary licenses
  - Commercial licenses never expire
  - Software upgrades allowed for up to 1 year from date indicated.

Stopping and Restarting (All Platforms)

PERFORCE
Version everything.

```
p4 admin stop
```

```
p4 admin restart
```

P4Admin *live*

PERFORCE
Version everything.

- Select multiple server instances
- Overview of server data
- Display `System Info`

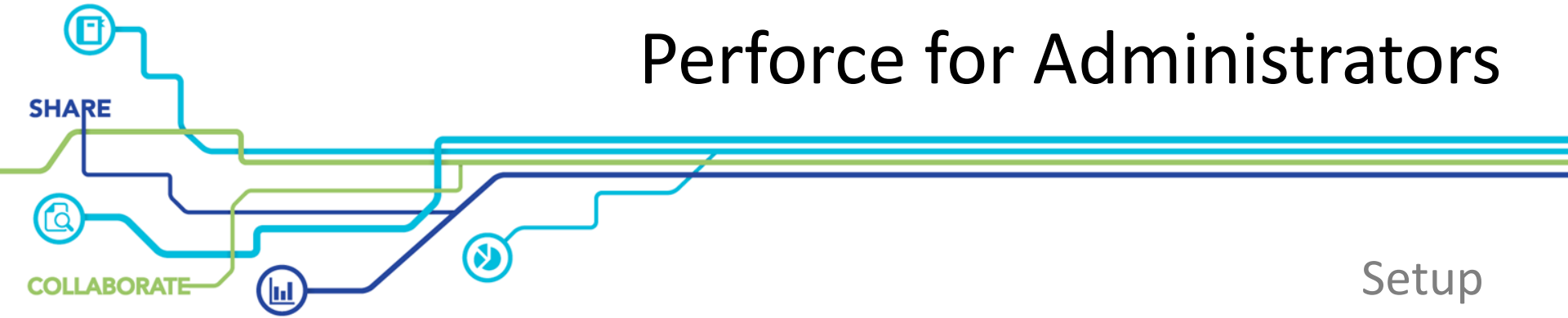
Exercises

PERFORCE
Version everything.

- Lab Set 1: Installation

PERFORCE
Version everything.

Introduction to Perforce for Administrators



Setup

Setup - Overview

PERFORCE
Version everything.

- Is the server running?
- File types and typemap
- Configuration

Is the Server Running?

PERFORCE
Version everything.

```
p4 info -s
User name: bob
Client name: bob_jam
Client host: mars
Current directory: c:\work\Jam\MAIN
Client address: 127.0.0.1:53183
Server address: london:1666
Server root: /usr/perforce
Server date: 2012/01/25 09:18:03 +0000 GMT
Server uptime: 354:51:41
Server version: P4D/FREEBSD/2011.1/370818 (2011/10/19)
Server license: ACME 100 users (expires 2012/01/29)
Server license-ip: 10.0.1.67
Case Handling: insensitive
```

-s : short and fast
Does not lock the database

Reminder: File types

PERFORCE
Version everything.

- Base file types

  `text` `binary`
  `symlink` `unicode` `utf16`

- Workspace modifiers

  `+x` - executable
  `+l` - exclusive open
  `+m` - sync vs. submit modtime
  `+w` - always writable
  `+k` - RCS keyword

- Server storage attributes

  `+Sn` - latest revision(s)
  `+C` - compressed
  `+D` - deltas
  `+F` - full file

Mapping files to Perforce File Types

PERFORCE
Version everything.

- Perforce detects file type text or binary
- Override with default file type mappings:

```
p4 typemap
```

```
Typemap:
        +m              //depot/Jam/MAIN/src/...
        binary+l        //....pdf
        text+k          //depot/Jam/MAIN/RCSimport/....txt
        binary+x        //depot/Jam/DEV/proj/executables/...
        binary+Fl       //depot/....gz
        text+C          //depot/....genfile
```

The ‘configure’ Command

PERFORCE
Version everything.

- Manage server configuration variables

```
p4 configure set variable=value
p4 configure unset variable
p4 configure show [variable]
```

```
> p4 configure show
      P4ROOT=/Perforce/main
      P4PORT=1666
      P4JOURNAL=journal (default)
      monitor=2 (configure)
      server: 3 (P4DEBUG)
      …
```

- Change is immediate – normally no need to stop server

Configure for Environment Variables

PERFORCE
Version everything.

- Can also be used for environment variables
  - P4PORT
  - P4NAME
  - P4LOG
  - P4AUDIT
  - P4JOURNAL
    - Needs to be set offline
  - P4DEBUG (but usually use *server* configurable)
  - Not P4ROOT and TMP/TEMP

- Changing of P4PORT requires restart of server

Offline Configuration

- `p4d -cshow`
- `p4d '-cset variable=value'`
- `p4d '-cunset variable'`

- Setting P4JOURNAL with the server offline.

```
> p4d -r . '-cset P4JOURNAL=/p4/1/logs/journal'
> p4d -r . -cshow
any: P4JOURNAL = /p4/1/logs/journal
```

Sample Configurables

PERFORCE
Version everything.

Use p4 help configurables for a list of all variables

Configurable	Default	Typical	Comment
monitor	0	1	Enable monitoring of active processes
security	0	3	User/password security level
dm.user.noautocreate	0	2	Restrict or Disable automatic creation of users
minClient	none	none	Lowest client version that can connect
minClientMessage	none	none	Message to issue for client-too-old
filesys.*.min	250M		Minimum space required for key filesystem(s)
defaultChangeType	none		restricted or public
dm.integ.engine	3		'2' enables the pre 2013.2 default integration engine

Enabling Process Monitoring

- Enable monitoring
- Monitoring is disabled by default
- Details are covered later

0	no monitoring
1	active processes only
2	also show idle processes

```
p4 configure set monitor=1
For server 'any', configuration variable 'monitor' set to '1'
```

```
Use p4 monitor show to view monitor output
```

Setting Server Security Level

PERFORCE
Version everything.

- Security settings determine how Perforce Server enforces passwords
- Display security counter value

```
p4 configure show security
security=3 (configure)
```

- Set security counter

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

0	No password required, any password allowed (default)
1	Strong password is required, can be stored in Windows registry
2	Strong password is required, cannot be stored in registry
3	*p4 login* tickets only, no password stored anywhere
4	Level 3 + proxy and brokers must use service user to connect

Setting dm.user.noautocreate

- By default, Perforce will allow anyone to create a new user

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

0	New user is created automatically by running any command
1	New user has to be created explicitly with *p4 user*
2	Only a super user can create new user with *p4 user -f*

Setting run.users.authorize

- By default, Perforce will allow anyone to list the users on the server

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

0	Users allowed to run “users” without authenticating
1	Users must login before running the “users” command

Minimum Space for Filesystem

PERFORCE
Version everything.

- Values can be specified in K,M,G,T or %
  - filesys.depot.min
  - filesys.P4ROOT.min
  - filesys.P4JOURNAL.min
  - filesys.P4LOG.min
  - filesys.TEMP.min

```
> p4 configure set filesys.P4ROOT.min=1G
```

- Prevents the server from running out of disk space

Know the defaults

PERFORCE
Version everything.

- **`p4 help configurables`**

```
Name                        Default Use
----                        ------- ---
…
filesys.P4ROOT.min          250M    Minimum space for P4ROOT filesystem
filesys.P4JOURNAL.min       250M    Minimum space for P4JOURNAL filesystem
filesys.P4LOG.min           250M    Minimum space for P4LOG filesystem
filesys.TEMP.min            250M    Minimum space for TEMP filesystem
…
```

- Filesys.min settings have reasonable defaults in 2014.2

Recommended Corporate Settings

PERFORCE
Version everything.

- `p4 configure set dm.user.noautocreate=2`
- `p4 configure set security=4`
- `p4 configure set run.users.authorize=1`
- `p4 configure set defaultChangetype=restricted`
- `p4 configure set dm.user.resetpassword=1`

Naming the server

PERFORCE
Version everything.

- `p4 serverid [serverID]`
- Good practice for documenting topology
- Server types: Brokers, Proxies, Replicas, etc.
- Server names are used in replication and failover scenarios
- Security
  - Ability to require special service accounts for access by remote servers
- More details in the Advanced Enterprise Administration course

Exercises

PERFORCE
Version everything.

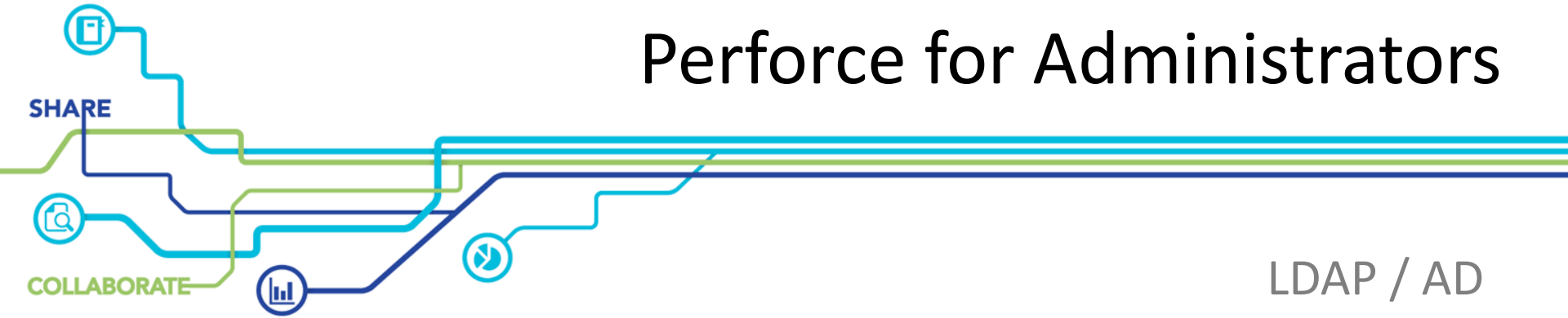
Lab Set 2: Setup

New commands in this chapter:

- `p4 typemap`
- `p4 configure`
- `p4 server`

PERFORCE
Version everything.

Introduction to Perforce for Administrators



LDAP / AD

LDAP / AD integration

PERFORCE
Version everything.

- Allows external authentication without triggers
- `p4 ldap example-config`
- Basic options in specification:

```
Name:          example-config
Hostname:      ldap.example.com
Port:          389
Encryption:    TLS
```

– Configuration name
– Host LDAP / AD listens on
– Port LDAP / AD listens on
– Encryption (SSL or TLS)

- Advanced options in specification…

LDAP / AD integration – SASL BIND

- Server generally only requires username & password to authenticate a user
- Efforts to resolve a user are delegated to the server

```
BindMethod:     sasl

SaslRealm:      example
```

Only where required (Active Directory domains)

- Active Directory supports SASL by default.
- So do most LDAP servers

LDAP / AD integration – simple BIND

PERFORCE
Version everything.

- If SASL BIND is not appropriate…

```
BindMethod:       simple
SimplePattern:    uid=%user%,ou=users,dc=example,dc=com
```

- UserID substituted when request is sent to authentication server

LDAP / AD integration – Search BIND

PERFORCE
Version everything.

- If SASL BIND is not appropriate…
- Searches the ‘ou’ called ‘users’ for a match

```
BindMethod:       search
SearchBaseDN:     ou=users,dc=example,dc=com
SearchFilter:     (&(objectClass=inetOrgPerson)(uid=%user%))
SearchScope:      subtree
SearchBindDN:     uid=read-only,dc=example,dc=com
SearchPasswd:     ******
```

- Generally requires authorization
- Uses `%user%` placeholder

LDAP / AD integration – Groups

PERFORCE
Version everything.

- LDAP groups work the same as the search method
- For a group named ‘p4’:

```
GroupBaseDN:   ou=groups,dc=example,dc=com
SearchFilter:  (&(objectClass=posixGroup)(cn=p4)(memberUid=%user%))
SearchScope:   subtree
```

LDAP / AD integration – Testing

PERFORCE
Version everything.

- Test configuration before enabling it:

```
p4 ldap -t bruno example-config
```

- Output is either a success message or the LDAP error message
  - For debug purposes only
  - Not reported to users

```
c:\>p4 ldap -t bruno example-config
Enter password:
Authentication as cn=bruno,ou=employees,dc=example,dc=com failed.
Reason: Invalid Credentials
```

LDAP / AD integration – Enabling

PERFORCE
Version everything.

- Configurations are enabled
  - Via server configurables:

```
p4 configure set auth.ldap.order.1=example-config
p4 configure set auth.default.method=ldap
```

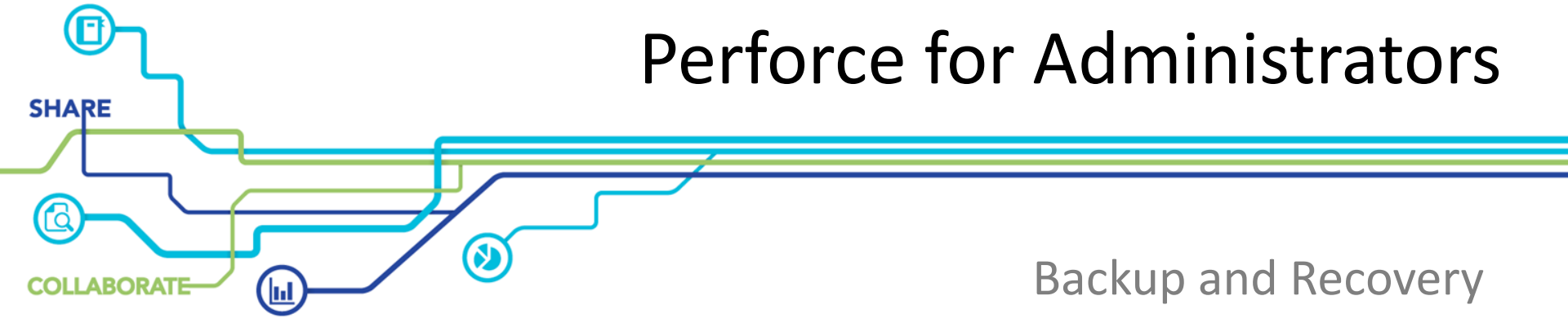
  - Override with **`AuthMethod:`** field in user specifications

```
AuthMethod:    ldap
```

`perforce / ldap`

  - Useful for bypassing ldap for certain users.
- With this enabled, AUTH triggers are automatically disabled

PERFORCE
Version everything.



Introduction to Perforce for Administrators

Backup and Recovery

P4ROOT and Depot Storage Layout

PERFORCE
Version everything.

- Database tables:

```
db.change     db.domain    db.integ
db.counters   db.fix       db.job
db.depot      db.fixrev    db.jobdesc
db.desc       db.have      db.label
```

...etc.

- Depot directories:

```
/p4/1/depots/depot
/p4/1/depots/nu
```

...etc.

A Few Definitions

PERFORCE
Version everything.

- Checkpoint
  - Easy to backup archive of databases (db.* files).
  - Transactionally consistent (a snapshot of metadata).
  - *Prepares* metadata for backup.
- Journal
  - Log of updates to metadata since last checkpoint.
  - Or since last journal rotation.
- Versioned file verification
  - Checksums in metadata for integrity check for archives

Checkpoint/Journal Format

PERFORCE
Version everything.

- Text file containing journal records
- Each record has a type
  - Checkpoint only has @pv@ entries
- Strings are surrounded by @ symbol
- Each value record refers to
  - A database table
  - The table version

Record	Type
@pv@	Put value = insert
@dv@	Delete value = delete
@rv@	Replace value = update
@vv@	Verify value = select
@ex@	commit
@mx@	flush
@nx@	Journal note

- http://www.perforce.com/perforce/doc.current/schema/

Sample Backup Steps

PERFORCE
Version everything.

- Checkpoint the database
- Backup using routine procedures

  ...checkpoint

  ...journal archive

  ...versioned files

Checkpoint the Database

PERFORCE
Version everything.

- Create a Checkpoint:

  Archives db.* files into a single file (“the checkpoint file”).

  Rotates the active journal to a numbered file.

  Starts a new active journal.

  Increments “journal counter”.

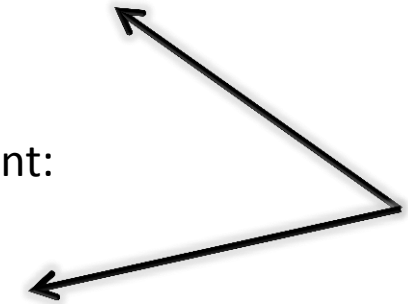
```
p4 admin checkpoint -Z
```

or

```
p4d -r /p4/1/root -J /p4/1/logs/journal -jc -Z
```

Checkpoint and Journal Numbering

- First Checkpoint:
  - checkpoint.1
  - journal.0
  - journal
- Second Checkpoint:
  - checkpoint.2
  - journal.1
  - journal
- Third Checkpoint:
  - checkpoint.3
  - journal.2
  - journal



checkpoint.2 + journal.2 is metadata-equivalent to checkpoint.3

Useful checkpoint features

PERFORCE
Version everything.

- Using a prefix

```
p4d -r /p4/1/root -jc /p4/1/checkpoints/p4_1
/p4/1/checkpoints/p4_1.ckp.3
/p4/1/checkpoints/p4_1.jnl.2
```

- Compressing checkpoint files (but not journals) with prefix

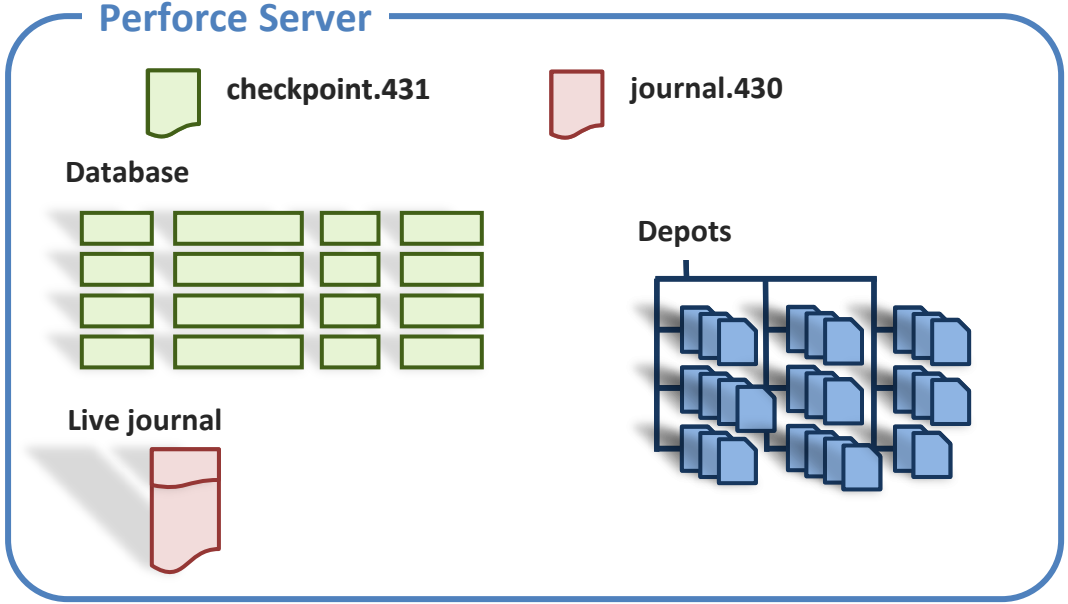
```
p4d -r /p4/1/root -jc -Z /p4/1/checkpoints/p4_1
/p4/1/checkpoints/p4_1.ckp.3.gz
/p4/1/checkpoints/p4_1.jnl.2
```

- Use –Z (rather than –z) to be replication-friendly. Compresses checkpoints leaving journals uncompressed.

Backup Process

PERFORCE
Version everything.

```
p4 admin checkpoint
```



Backup Example

- Monday starting at 2:00 AM
  - `p4d -jc -Z /p4/1/bckp/p4_1`
    `p4_1.ckp.426.gz, p4_1.jnl.425 created`
  - Backup Depot files, p4_1.ckp.426.gz, p4_1.jnl.425

- Tuesday starting at 2:00 AM
  - `p4d -jc -Z /p4/1/bckp/p4_1`
    `p4_1.ckp.427.gz, p4_1.jnl.426 created`
  - Backup Depot files, p4_1.ckp.427.gz, p4_1.jnl.426

Perforce Database Regeneration

PERFORCE
Version everything.

- Inconsistent databases
- Data lost through disk crash
- Data lost through accidental admin activity
- Bloated databases
  - Restored databases are smaller than original, but contain equivalent data
    - Removes empty pages
    - Rebalances the index trees

Checkpoint Recovery

PERFORCE
Version everything.

- Stop p4d process, then move database files:

```
cd /p4/1/root        ← P4ROOT
mkdir save
mv db.* save
```

- Normal recovery

```
p4d -r . -jr -z /p4/1/bckp/p4_1.ckp.3.gz
p4d -r . -jr /p4/1/logs/journal
```

- Restart p4d process

Checkpoint Recovery

PERFORCE
Version everything.

- If last checkpoint unavailable

```
p4d -r . -jr -z /p4/1/bckp/p4_1.ckp.2.gz
p4d -r . -jr /p4/1/bckp/journal.2
p4d -r . -jr /p4/1/logs/journal
```

- If last checkpoint and current journal unavailable (data loss)

```
p4d -r . -jr -z /p4/1/bckp/p4_1.ckp.2.gz
p4d -r . -jr /p4/1/bckp/journal.2
```

Verify Versioned File Contents

PERFORCE
Version everything.

```
p4 verify Build.com
 //depot/Jam/MAIN/src/Build.com#7 - edit change 4668 (text)
 86638AC6C2BCA86BBEFED8581F424FDC
 //depot/Jam/MAIN/src/Build.com#6 - edit change 2475 (text)
 6D2C49C8DEBC5C3CF41BD87A8355D354
 //depot/Jam/MAIN/src/Build.com#5 - edit change 1784 (text)
 CAA2DC4BFC25A836C5D37E46FB92B016
```

...etc.

- Use quiet mode to report only errors

```
p4 verify -q //...
```

- Normally run via a script on the weekends

Where versioned files are stored

PERFORCE
Version everything.

- When first added, lbrFile is same as depotFile:

```
p4 fstat -Oc //depot/jam/main/src/jam.c
... depotFile //depot/Jam/MAIN/src/jam.c
... lbrFile //depot/Jam/MAIN/src/jam.c
```

- But revision 1 of branched versions point at the original file:

```
p4 fstat -Oc //depot/jam/rel2.1/src/jam.c#1
... depotFile //depot/Jam/REL2.1/src/jam.c
... lbrFile //depot/Jam/MAIN/src/jam.c
```

How versioned files are stored

PERFORCE
Version everything.

- How text files are stored:

```
dir /s/b "\perforce\depot\Jam\MAIN\src\jam.c*"
C:\perforce\depot\Jam\MAIN\src\jam.c,v
```

- How binaries are stored:

```
dir /s/b "\perforce\depot\Misc\manuals"
:
C:\perforce\depot\Misc\manuals\triggers.doc,d\1.436.gz
```

Exercises

PERFORCE
Version everything.

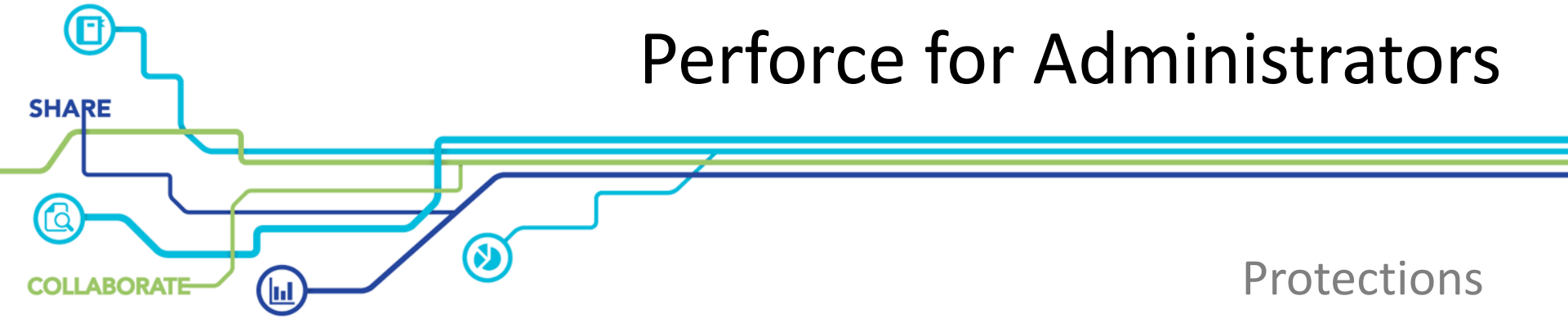
Lab Set 3: Backup and Recovery

New commands in this chapter:

- `p4 admin checkpoint`
- `p4 admin journal`
- `p4d -jc`
- `p4d -jr`
- `p4 verify`

PERFORCE
Version everything.

Introduction to Perforce for Administrators



Protections

User Protections

PERFORCE
Version everything.

- User types
- Access levels
- The p4 protect command
- The protections table
- Using groups
- Server traffic & protections

User Spec

PERFORCE
Version everything.

- `p4 user earl`

```
User:       earl
Email:      earl@acme.com
Update:     2012/07/11 14:21:06
Access:     2012/10/11 19:24:05
Type:       standard
FullName:   Earl Ashby
Reviews:
        //depot/Jam/MAIN/…
```

User Types

PERFORCE
Version everything.

- Type can be:
  - 'standard' (default), 'service' or 'operator'

- Standard user consumes a license
- Service and operator users are restricted
  - Do not consume licenses
  - Need entry in the protection table
  - Ignores AUTH_CHECK trigger. Uses local password or service-check instead
- A “Background User” is a licensing concept; in Perforce it is a standard user.

Service User

PERFORCE
Version everything.

- Special user for background processes
- Can also be used for
  - Replica servers
  - Edge servers
  - Proxy servers
  - Remote depots
- Needs entry in the protection table, typically ‘super’
- Can only run a few limited commands

p4 dbschema	p4 info	p4 login	p4 logout
p4 passwd	p4 pull	p4 user	

Operator User

- Special user for IT admins
- Needs entry in the protection table, typically ‘super’
- Runs only administrative commands that cannot access file content. Ex:

p4 admin stop
p4 admin restart
p4 admin checkpoint
p4 admin journal
p4 dbstat
p4 dbverify
p4 diskspace
p4 configure
p4 counter
p4 counters
p4 journaldbchecksums
p4 jobs
p4 login
p4 logout
p4 logappend
p4 logparse
p4 logrotate
p4 logschema
p4 logstat
p4 logtail
p4 lockstat
p4 monitor
p4 passwd
p4 ping
p4 server
p4 serverid
p4 verify -q
p4 user

Access Permissions

PERFORCE
Version everything.

Level:	Allows:
• `list`	access to metadata but not file contents
• `read`	viewing file contents; can't open for edit/move/etc.
• `open`	opening files but not submitting
• `write`	submitting file modifications
• `review`	p4 review command (external application use)
• `admin`	overriding changes to metadata
• `super`	commands that affect server operation

Creating the Protections Table

PERFORCE
Version everything.

`p4 protect`

```
Protections:
   write     user      *       *       //...
   super     user      lisa    *       //...
```

access level | user or group | name | host | files

Protections Table Rules

PERFORCE
Version everything.

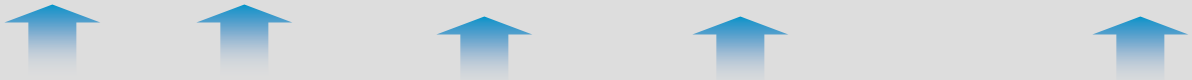
- Grant model: Users have no access unless it is explicitly granted
  - Grant using a tuple (user, host, command, [file pattern])
- Protections table processed the bottom up
  - If found matching, grant access (even if access is denied higher up)
  - If exclusionary mapping matches, any further access is denied
  - If top of table is reached, access is denied
- Consequences:
  - Order does not matter (except in the presence of exclusionary mappings)
  - Highest level of access granted before an exclusionary mapping applies

Using the protections table

PERFORCE
Version everything.

p4 protect

```
Protections:
   read      group     p4users     192.3.24.0/24      //...
   write     group     devgroup    *                  //...
   list      user      mike        *                  -//depot/...
   write     user      mike        *                  //depot/Jam/MAIN/doc/...
   read      user      emily       *                  //depot/Jam/MAIN/svr/...
   list      user      remote      *                  -//...
   list      group     _           comment            "Disables remote depots"
   super     user      lisa        *                  //...
```



Fine Tuning Access

PERFORCE
Version everything.

- For example, remove **write** permission for //project/MAIN/...

```
write   group   developers  *   //project/...
list    group   developers  *   -//project/MAIN/...
read    group   developers  *   //project/MAIN/...
```

- Alternative: specific rights =read, =branch, =open, =write

```
write   group   developers  *   //project/...
=write  group   developers  *   -//project/MAIN/...
```

- If **`=branch`** is denied, users cannot use these files as the *source* of integration

Recommendations

PERFORCE
Version everything.

- Better to avoid using multiple wild cards.
  - Don’t hide a “protected” directory in each product tree

```
list user * * -//depot/.../protected/...
```

  - Instead, create one higher level protected directory

```
write group protected * //depot/protected/...
```

  - At the very least use an * instead of an embedded “…”

```
list user * * -//depot/*/protected/...
```

User Groups

PERFORCE
Version everything.

- Single protection lines affect groups of users
- Set maximum data size for commands
- Set login timeout
- Set password expiry
  - ignored when external authentication is used

Creating and Editing Groups

PERFORCE
Version everything.

```
p4 group devgroup
```

```
Group:                  devgroup
MaxResults:             unset
MaxScanRows:            unset
MaxLockTime:            unset
Timeout:                43200
PasswordTimeout:        unset
Subgroups:
                        devcontract
Owners:
                        raj
Users:
                        joe
                        sharon
                        mike
```

Setting Data Access Limits

PERFORCE
Version everything.

```
p4 group limits
```

```
Group:           limits
MaxResults:      50000
MaxScanRows:     250000
MaxLockTime:     30000
Timeout:         32400
PasswordTimeout: unset
Subgroups:
Owners:
Users:
                 dave
                 anita
```

Managing Groups

PERFORCE
Version everything.

- Edit a group

```
p4 group -a devcontractors (group owner)
p4 group devcontractors (super user)
```

- Delete a group

```
p4 group -d devcontractors
Group devcontractors deleted.
```

Listing Groups

PERFORCE
Version everything.

- Show all groups

```
p4 groups
```

- Show group membership

```
p4 groups -i bruno
    admins
    perfuser
```

Listing Protections

PERFORCE
Version everything.

- Current user (non-privileged usage)

```
p4 protects
write group p4users * //depot/...
super user bruno * //...
```

- Any user or group

```
p4 protects -g p4readers
read group p4readers * //depot/...

p4 protects -u bob //depot/product/...
write group dev * //depot/product/...
```

P4Admin - Protections and Groups *live*

PERFORCE
Version everything.

- Create a group for your users
- Assign permissions
- Preview changes to permissions
- Graphical display of permissions

Exercises

PERFORCE
Version everything.

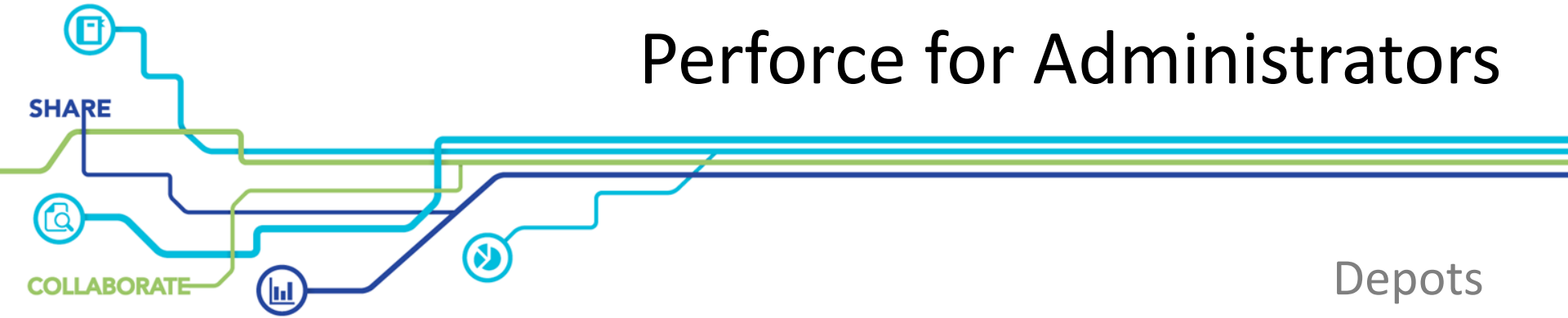
Lab Set 4: Protections

New commands in this chapter:

- `p4 protect`
- `p4 group`
- `p4 groups`
- `p4 protects`

PERFORCE
Version everything.

Introduction to Perforce for Administrators



Depots

Perforce Depots

PERFORCE
Version everything.

- Storage for versioned files:
  - local (default)
  - stream
- Special Depots:
  - archive (covered in advanced training)
  - unload (singleton)
  - remote
  - spec (singleton, usage covered in advanced training)
- Where are they? Located under P4ROOT by default.
  - Recommended to move to separate storage
  - p4 configure set server.depot.root=<your depot storage location>

Depot Map

PERFORCE
Version everything.

```
p4 depot depot
```

```
Depot:       depot
Owner:       p4admin
Date:        2009/04/20 17:18:22
Description: Production depot
Type:        local
Address:     local
Map:         /p4/1/depots/depot/...
```

Use *Map* with absolute path to place depot directory outside of P4ROOT

Creating New Depots

PERFORCE
Version everything.

```
p4 depot fgs
```

```
Depot:       fgs
Owner:       p4admin
Date:        2011/12/13 09:25:44
Description: Friendly Greeting System
Type:        stream
Address:     local
Map:         /p4/1/depots/fgs/...
```

Creating a Spec Depot

PERFORCE
Version everything.

```
p4 depot spec
```

```
Depot:       spec
Owner:       p4admin
Description: Implicitly versions P4 specs.
Type:        spec
Address:     local
Map:         /p4/1/depots/spec/...
SpecMap:
    //spec/…
    -//spec/client/continuous_integration_build_ws_*
```

Populate the Spec Depot with a baseline of current specs:

```
p4 admin updatespecdepot -a
```

Creating an Unload Depot

PERFORCE
Version everything.

```
p4 depot unload
```

```
Depot:       unload
Owner:       p4admin
Description: Used for unloading labels and workspaces.
Type:        unload
Address:     local
Map:         /p4/1/depots/unload/...
```

Working with Other Depots

PERFORCE
Version everything.

- Listing depots:

```
p4 depots
Depot stuff 2012/05/26 local /p4/1/depots/stuff/... 'Created by p4admin.'
Depot CompA 2012/07/29 stream /p4/1/depots/CompA/... 'Component A.'
Depot CompB 2012/07/29 stream /p4/1/depots/CompB/... 'Component B.'
Depot spec 2012/02/24 spec /p4/1/depots/spec/... 'Created by ralph.'
Depot nu 2011/06/12 local /p4/1/depots/nu/... 'Created by ralph.'
Depot usr 2012/01/10 remote perforce:1888 //usr/... 'Created by sara.'
```

- Listing files in a depot:

```
p4 files //usr/tools/...
```

Integrating Multiple P4D Instances

PERFORCE
Version everything.

```
p4 depot spice
```

'Map:' field ties these paths together

```
Depot:       spice
Owner:       bob
Date:        2005/05/23 9:01:12
Description: Path from Arrakis server
Type:        remote
Address:     arrakis:1666
Map:         //depot/oak/MAIN/...
```

- Completing an integration from another P4D instance:

```
p4 integ //spice/src/... //depot/IMPORT/src/...
```

Local path `//spice/src/…` maps to `//depot/oak/MAIN/src/…` in remote instance

Path points to local instance

P4Admin *live*

- List and create depots

Exercises

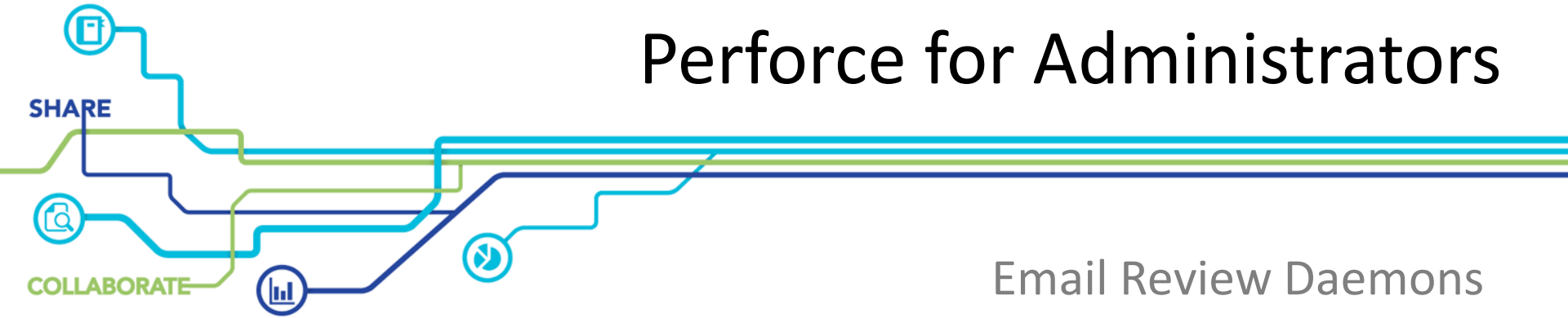
Lab Set 5: Depots

New commands in this chapter:

- `p4 depot`
- `p4 depots`

PERFORCE
Version everything.

Introduction to Perforce for Administrators



Email Review Daemons

Review Script

PERFORCE
Version everything.

- Largely supplanted by Perforce Swarm
- E-mail notification of:
  - changes submitted
  - jobs logged and modified
- Set "Reviews:" field in user profile
- Install and configure the review script
  - See: Perforce KB - Configure Review Daemon

Subscribing to Email Reviews

PERFORCE
Version everything.

`p4 user`

```
User:   bob
Email:  bob@caniche.com
Update: 2008/08/09 13:45:59
Access: 2008/11/07 16:45:05
FullName:   Bob Everly
Reviews:
        //depot/Jam/MAIN/technotes/...
        //depot/Jam/DEV/projects/proj1/...
```

Add “Reviews:” field if running review script

Configuring the Email Review Daemon

PERFORCE
Version everything.

- Create a review user
  - Needs **review** access through protection table

```
review user myreviewer * //...
```

  - Add to its own group with an unlimited ticket

- Available from the Perforce Workshop

Email Review sample email

PERFORCE
Version everything.

```
Change 12819 by bruno@bruno-ws33 on 2014/10/30 11:11:54

        Updates to training slides and some new test content

Affected files ...

... //depot/training/slides.pptx#4 edit
... //depot/training/test_content.pptx#1 add

Swarm: http://swarm.perforce.com/change/12819

P4Web: http://p4poke.perforce.com:8080/12819?ac=10
```

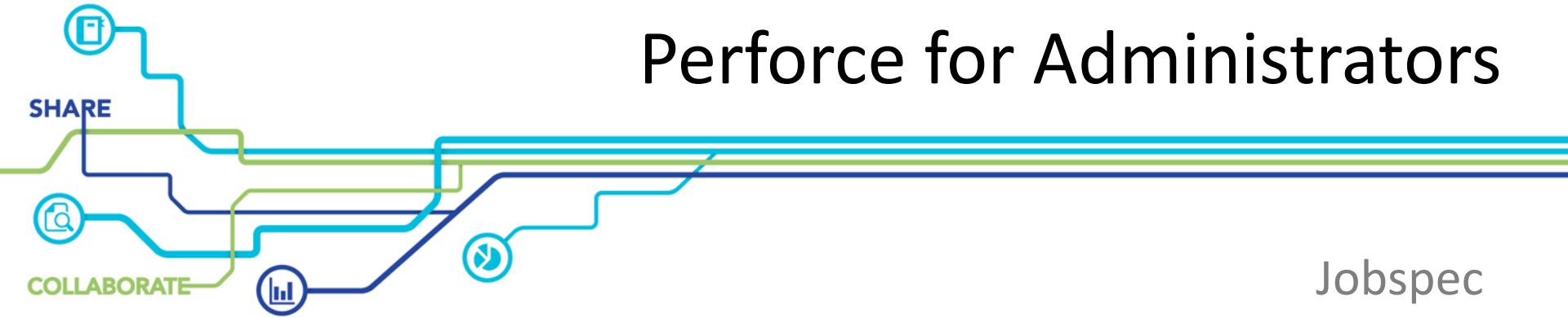
Perforce Swarm

PERFORCE
Version everything.

- More than just e-mail notification
- Pre-commit code review
- Interactive social experience
- Hooks for automated validation
- Better team collaboration

PERFORCE
Version everything.

Introduction to Perforce for Administrators



Jobspec

Jobs - Definitions

PERFORCE
Version everything.

- **Job** - a numbered/named work request
  - bug / defect
  - enhancement request
  - …
- **Job Specification**
  - template defining fields/attributes associated with each job
  - customizable to meet local requirements

Customizing the Job Specification

PERFORCE
Version everything.

p4 jobspec

```
Fields:
        101 Job word 32 required
        102 Status select 10 required
        103 User word 32 required
        104 Date date 20 once
        105 Description text 72 required
        140 Release select 10 optional
Comments:
     # Include plentiful comments!
     # Job: The job name. 'new' generates a sequenced job number.
     # Your users need to know what the fields mean, for example:
     # Release: One of ‘4.0’, ‘4.1’, ‘4.2’, or ‘5.0’
Values:
        Status open/cannot_dup/duplicate/in-progress/closed
        Release 4.0/4.1/4.2/5.0
Presets:
        Status open,fix/in-progress
        Date $now
        Description $blank
```

Jobspec Form Fields

PERFORCE
Version everything.

- Datatype

  `word` `date` `select` `line` `text` `bulk`

- Persistence

  `optional` `default` `required` `once` `always`

- Built-in variables

  `$user` `$now` `$blank`

Editing a Job (with jobspec)

PERFORCE
Version everything.

```
p4 job job000053
```

```
# Include plentiful comments!
# Name: The job name. 'new' generates a sequenced job number.
# Your users need to know what the fields mean, for example:
# Release: One of '4.0', '4.1', '4.2', or '5.0'
Name: job000053
Status:  open
User: karen
Date: 2009/01/23 10:29:46
Description:
      Fix file locking bug.
Release: 4.2
```

Perforce Defect Tracking Gateway

PERFORCE
Version everything.

- P4DTG
  - Plugins for Jira, Redmine, HP QualityCenter, Bugzilla etc ...
  - Check Perforce Website for detail
- Customize jobspec with defect tracker fields
- Info in both systems kept in sync
- P4DTG SDK available for custom integrations

Exercises

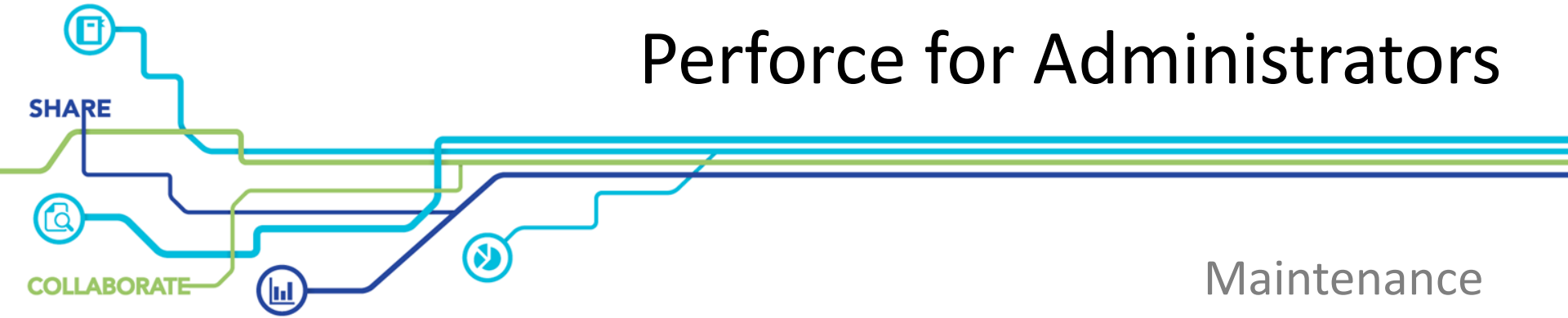
Lab Set 6: Jobspec

New commands in this chapter:

- `p4 jobspec`

PERFORCE
Version everything.

Introduction to Perforce for Administrators



Maintenance

Super and Admin Commands

PERFORCE
Version everything.

- Resetting users’ passwords
- Maintaining spec objects
- Archiving and obliterating files

Resetting Passwords

PERFORCE
Version everything.

```
p4 passwd [username]
Enter new password:
Re-enter new password:
Password updated.
```

- Force a user to reset their password:

```
p4 admin resetpassword -u user
```

- Force all users to reset passwords:

```
p4 admin resetpassword -a
```

Database Cleanup

PERFORCE
Version everything.

- List specs owned by a user

```
p4 clients -u username
```

- Remove old specs

```
p4 user -f -d username
p4 client -f -d client_ws_name
p4 label -f -d labelname
```

- Delete empty pending changelists

```
p4 change -f -d changelist#
```

Changelist Maintenance

PERFORCE
Version everything.

- Updating a description

```
p4 change -f changelist#
```

- Unlocking files

```
p4 unlock -f -c changelist#
p4 -c earl_ws unlock -f //depot/Jam/...
```

Revert Exclusive Locked File

PERFORCE
Version everything.

- Determine **User** and **Workspace**:

```
p4 fstat //depot/path/file
```

- Look for **user@workspace** in output and then run:

```
p4 client -o <workspace>
```

- Look for **Host**: **hostname** in output, and then run:

```
p4 login <user>
p4 -u <user> -c <workspace> -H <hostname> \
                        revert -k //depot/path/file
```

Unloading Clients and Labels

PERFORCE
Version everything.

- Requires depot of type *unload*
- Unload clients and labels not currently needed
  - Metadata is removed from database and placed in unload depot as a file

```
p4 unload [-f -L -z] [-c client | -l label | -s stream ]
```

- Reload clients and labels as needed

```
p4 reload [-f] \
        [-c client | -l label | -s stream]
```

-f	Use if not owner
-L	Unload locked specs
-z	Compress unloaded

Batch Unloading

- Batch unloading of clients and labels

```
p4 unload [-f –L -z] [-a | -ac | -al] [-d date | -u user ]
```

```
p4 unload -f –L –z –al -d 2011/01/01
Label default-tag unloaded.
Label v1.0 unloaded.
```

-a	Unload labels and clients
-al	Unload labels
-ac	Unload clients

Reports on Unloaded Clients and Labels

PERFORCE
Version everything.

- Unloaded clients and labels are normally not shown in reports
- Show unloaded clients and labels

```
p4 clients -U
p4 labels -U
```

- Show files in *unload* depot

```
p4 files -U //unload/...
//unload/client/dummy_ws.ckp#none - edit default change (ltext)
//unload/label/dummy_label.ckp#none - edit default change (ltext)
```

Recommendations

PERFORCE
Version everything.

- Delete or unload old workspaces on a regular basis
  - Script based on last access date
- Unload old labels
  - Script based on last access date
  - Consider using automatically unloaded labels
- Remove empty pending changelists
  - Script to run p4 changes –s pending, and try deleting them all
  - Only empty changelists will be deleted
- Recover from a checkpoint to regain free space

Deleting and Obliterating

PERFORCE
Version everything.

- Delete...
  - ...marks head revision as deleted
  - ...keeps revision history

- Obliterate removes all traces of the file
  - Revisions
  - Integrations
  - Labels
  - Client workspaces

Obliterating files

PERFORCE
Version everything.

```
p4 obliterate //depot/Jam/MAIN/src/…

p4 obliterate -y //depot/Jam/MAIN/src/jam.ps

p4 obliterate -y //depot/Jam/MAIN/src/rules.c#3

p4 obliterate -ab //depot/Jam/Rel/3.1/src/…

p4 obliterate -abh //depot/Jam/Rel/3.1/src/…
```

-a	Do not delete archive files
-b	Only obliterate #1 if action == branched
-h	Do not delete client have entries

p4admin – manage users *live*

- Change a user’s password
- Delete a user

Exercises

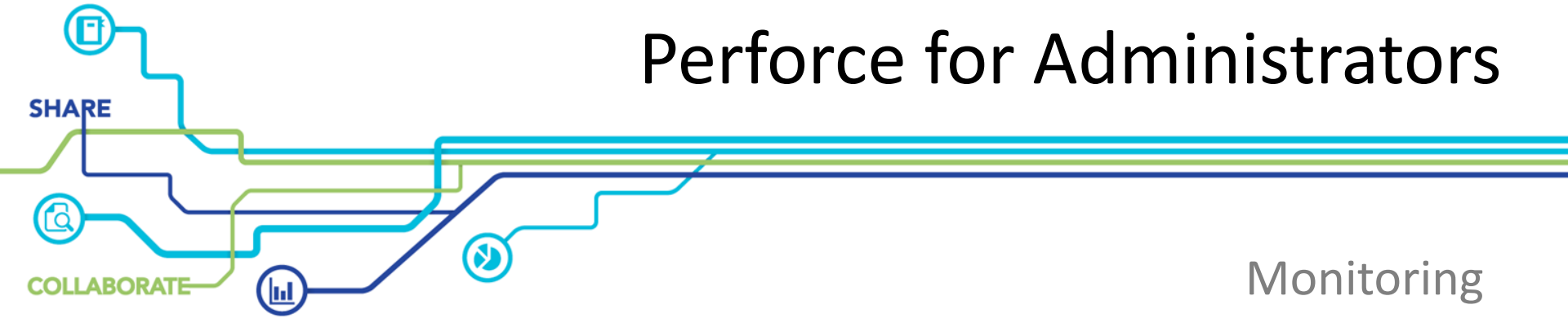
Lab Set 7: Maintenance

New commands in this chapter:

- `p4 passwd`
- `p4 <command> -f and -d`
- `p4 obliterate`

PERFORCE
Version everything.

Introduction to Perforce for Administrators



Monitoring

Monitoring - Overview

PERFORCE
Version everything.

- Monitoring
- Logging
- Other useful commands

Reminder: Enabling process monitoring

PERFORCE
Version everything.

- Monitoring needs to be enabled first

```
p4 configure set monitor=1
For server 'any', configuration variable 'monitor' set to '1'
```

- Effect is immediate, no need to restart server
- In multi-server setup can set configuration specific to individual server:

```
p4 configure set my_replica1#monitor=2
```

Monitoring Perforce activity

PERFORCE
Version everything.

- Show current processes

```
p4 monitor show
34  R rlo      00:06:31 sync
452 R clarks   00:00:00 monitor
```

- Show processes and command environment

```
p4 monitor show -a -e
34 p4/2009.2 10.0.0.18 R rl rlws 00:07:37 sync //depot/...
```

- Mark processes for termination

```
p4 monitor terminate 34
** process '34' marked for termination **
```

Monitor: Pause and resume

PERFORCE
Version everything.

- Pause an active process

```
p4 monitor pause 34
** process '34' record paused **
```

- Resume a paused process

```
p4 monitor resume 34
** process '34' record resumed **
```

Diagnosing server swamp

PERFORCE
Version everything.

- Check free memory
- Number of p4d processes on Linux
  - Determine an average number of p4d processes when the server is not having an issue for comparison
- Other processes running on server?
  - Virus checkers
  - Backup utilities

...etc.

Log Analysis

PERFORCE
Version everything.

- Turn on debug level 3 (default is 1)

```
p4 configure set server=3
```

- Use the psla (Perforce Server Log Analyzer)
  - https://kb.perforce.com/psla
  - Details in the advanced course

Other useful commands

PERFORCE
Version everything.

Command	Description
`p4 diskspace`	Displays available disk space
`p4 logtail`	Displays last blocks of the error log
`p4 dbstat`	Displays sizes of database tables
`p4 logstat`	Displays sizes of error logs
`p4 lockstat`	Report lock status of database tables
`p4 dbverify`	Verify database integrity. Normally long-running Run with –U for a fast “table-not-unlocked” check.

- Useful for
  - Regular checks (but beware of database locking)
  - Investigation of performance problems

p4admin – monitoring servers *live*

- Log console for alerts
- Monitor console

Exercises

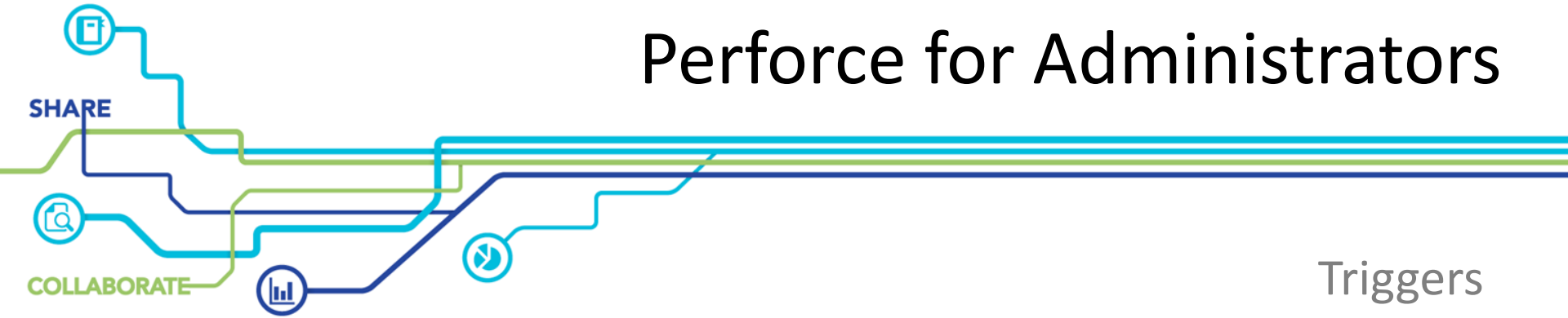
Lab Set 8: Monitoring

New commands in this chapter:

- `p4 monitor`
- `p4 diskspace`
- `p4 logtail`
- `p4 dbstat`
- `p4 logstat`
- `p4 dbverify`

PERFORCE
Version everything.

Introduction to Perforce for Administrators



Triggers

Supported trigger types

PERFORCE
Version everything.

- Password authentication
- Change processing
- Shelving
- Form manipulation or validation
- Job fix checking
- Command triggers

Use triggers to enforce policy

PERFORCE
Version everything.

- RELNOTES file always submitted with *.c files
- Submit to “rel1” codeline fixes at least one job
- Add “Reviewed by:” when changelist created
- All *.h files include copyright date
- Begin build after submit to “dev” codeline
- Notify code reviewers when files are shelved
- Prevent users from deleting jobs

Submit triggers

PERFORCE
Version everything.

- Change-submit

  Run after change is created and files locked

- Change-content

  Run after file transfer

- Change-commit

  Run after changelist commit

- Submit fails if submit or content trigger returns non-zero value

Shelving triggers

PERFORCE
Version everything.

- Shelve-submit
  - Run after change created
- Shelve-commit
  - Run after files are shelved
- Shelve-delete
  - Run before discarding shelved files
- Shelving fails if a shelve-submit trigger returns a non-zero value.

Form triggers

PERFORCE
Version everything.

- Form-out

  Run on generation of form

- Form-in

  Run on input of form

- In and out triggers may modify forms

Validated form triggers

PERFORCE
Version everything.

- Form-save

  Run after form parsed, before form saved

- Form-delete

  Run after form parsed, before form deleted

- Form-commit

  Run after form saved

Job fix triggers

PERFORCE
Version everything.

- Fix-add

  Run prior to adding fix

- Fix-delete

  Run prior to deleting fix

Command triggers

- Used to run a command before or after a command executes.
- **`pre-user-<command>`**

  Run prior to running the command
- **`post-user-<command>`**

  Run after running the command
- Command matching uses regular expressions
  - delimit them to prevent extra matching.

Command triggers

PERFORCE
Version everything.

- Examples including regular expressions:

`pre-user-login`	Before the login command
`post-user-(add|edit)`	After the add or edit command
`pre-user-obliterate`	Before the obliterate command
`(pre|post)-user-sync`	Before or after the sync command
`pre-user-change$`	Only matches ‘change’, and not ‘changes’

Password authentication triggers

- Auth-check

  Verify against external password manager

- Auth-set

  Send new password to external manager

Trigger Communication - default

PERFORCE
Version everything.

- Triggers default to using trigger variables to communicate with the server
  - %clienthost% Hostname of user's workstation
  - %clientip% IP address of user's workstation
  - %formfile% Path to temporary form spec. file
  - %formname% Name of the form (branch name, etc.)
  - STDIN Only used by archive/authentication triggers
  - STDOUT Sent as a message to the client
  - Etc.

Trigger Communication - dictionaries

PERFORCE
Version everything.

- Triggers can be set to communicate via dictionaries of name/value pairs

```
p4 configure set triggers.io=1
```

- Global setting – All triggers must be updated to use this
- SDTDIN Textual dictionary of name-value pairs of variables

  (Except **%clienthost%** and **%peerhost%**)
- STDOUT At a minimum contains an action. A message is optional.

- Details: http://www.perforce.com/perforce/doc.current/manuals/p4sag/chapter.scripting.html#triggers.communication

Trigger Communication - dictionaries

PERFORCE
Version everything.

- STDIN sample:

```
client:jgibson-aaaatchoooo
clientprog:P4/LINUX45X86_128/2017.9.MAIN/1773263782 (2017/OCT/09)
command:user-dwim user:jgibson
```

- STDOUT sample:

```
action:fail
message:too bad!
```

Calling Triggers

PERFORCE
Version everything.

- Triggers run on the server
- Called two ways
  - Depot path specified in the trigger script location
  - Call to executable file on server host

Creating triggers

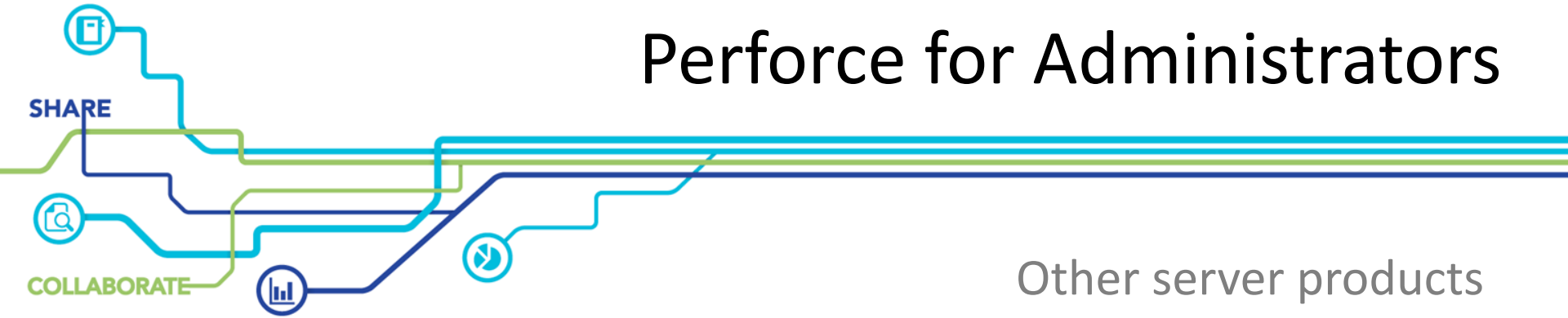
PERFORCE
Version everything.

p4 triggers

```
Triggers:
 passwd    auth-check          auth                  "python %//triggers/ldap_check.py% %user%"
 notes     change-content      //depot/bld/...       "relcheck.pl %user% %changelist%"
 review    change-submit       //depot/src/*.c       "//p4admin/triggers/reviewer.pl %changelist%"
 review    change-submit       //depot/src/*.h       "reviewer.pl %changelist%"
 new       form-out            change                "addinfo.pl %formfile%"
 jobs      change-submit       //depot/rel1/...      "/usr/bin/p4/jobs.sh %changelist%"
 keep      form-delete         branch                "python /usr/bin/spec.py %formfile%"
 fixone    fix-add             fix                   "//p4admin/triggers/checkfix.sh %jobs%"
 build     change-commit       //depot/src/...       "//p4admin/triggers/build.sh %changelist%"
 oblit     command             pre-user-obliterate "perl %//triggers/only_admin.pl% %user%"
```

name type path script

PERFORCE
Version everything.



Introduction to Perforce for Administrators

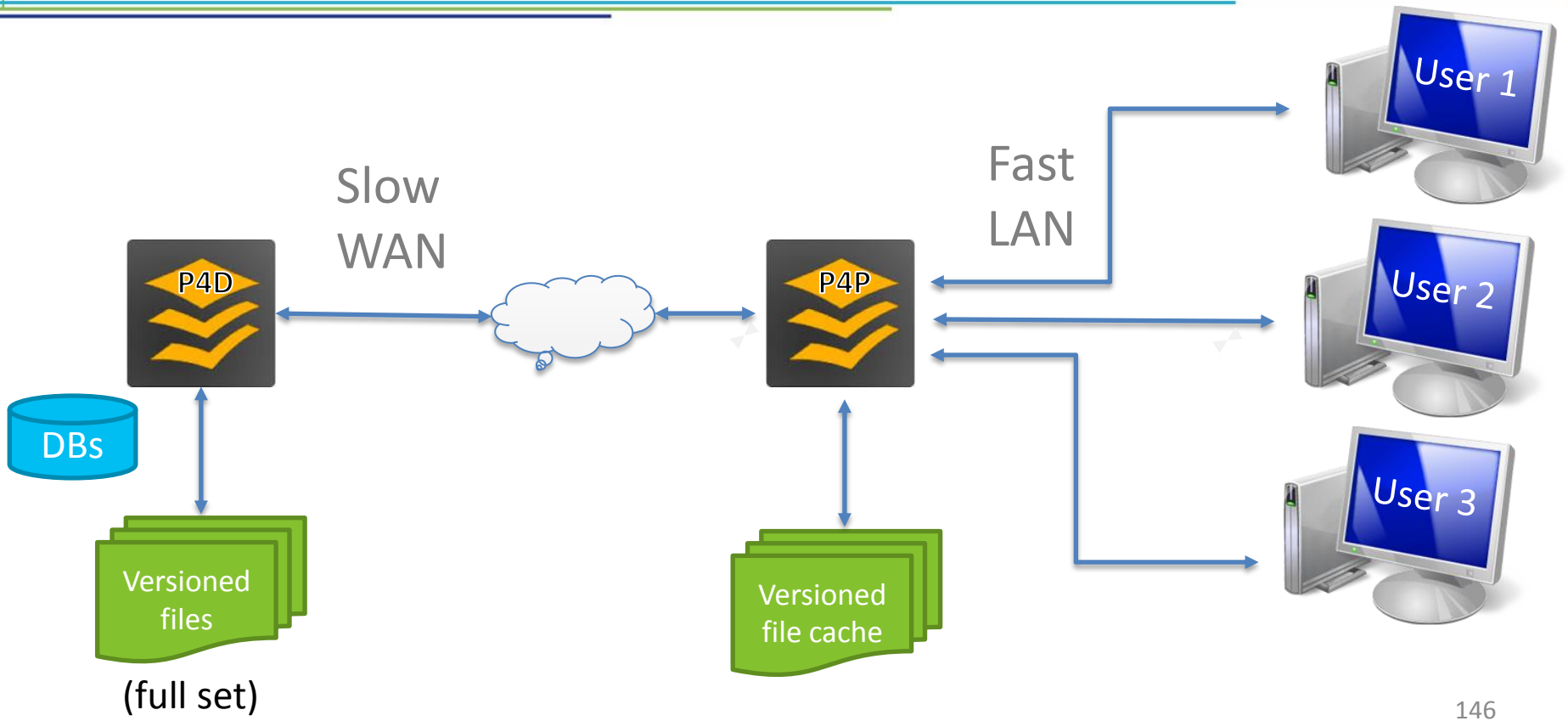
Other server products and resources

P4P – Perforce Proxy

PERFORCE
Version everything.

- Improves performance at low-bandwidth sites
- Caches frequently-requested file revisions
- Transparent to the users
- Low maintenance
  - Easy to set up for testing
  - No backups required (except for fast recovery)
  - Old revisions can be purged via script

Proxy


Slow
WAN
Fast
LAN
P4D
P4P
DBs
Versioned
files
(full set)
Versioned
file cache
User 1
User 2
User 3

P4P Hardware Requirements

PERFORCE
Version everything.

- Fast CPU
- Sufficient RAM for file system cache
- Good I/O subsystem with sufficient capacity
  - Holds subset of P4D depot data

- Linux preferred OS
  - Even for Windows-hosted Perforce Server

Proxy Server and protection table

PERFORCE
Version everything.

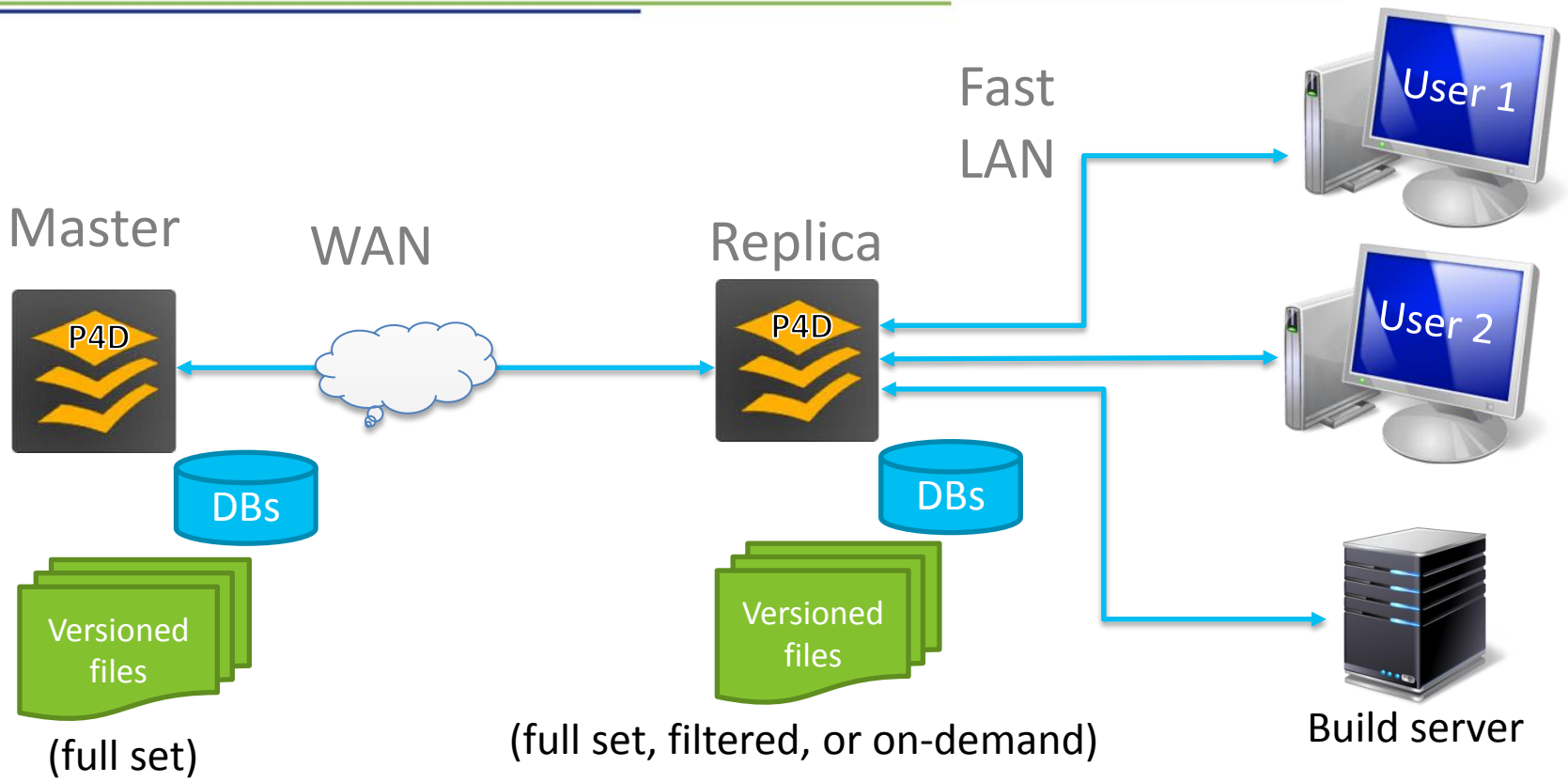
- Use protections to enforce users go through Proxy server
  - In this example server lives in 10.0.0.0/8 subnet

```
list    group    remotedev    192.168.10.0/24         -//...
write   group    remotedev    proxy-192.168.10.0/24    //...
list    group    remotedev    proxy-10.0.0.0/8        -//...
write   group    remotedev    10.0.0.0/8               //...
```

- To verify that files have been served by the Perforce Proxy, use P4 with –Zproxyverbose

```
$ p4 -Zproxyverbose sync cached.txt
//depot/main/cached.txt - refreshing /home/p4adm/tmp/cached.txt
File /home/p4adm/tmp/cached.txt delivered from proxy server
```

Replication Architecture

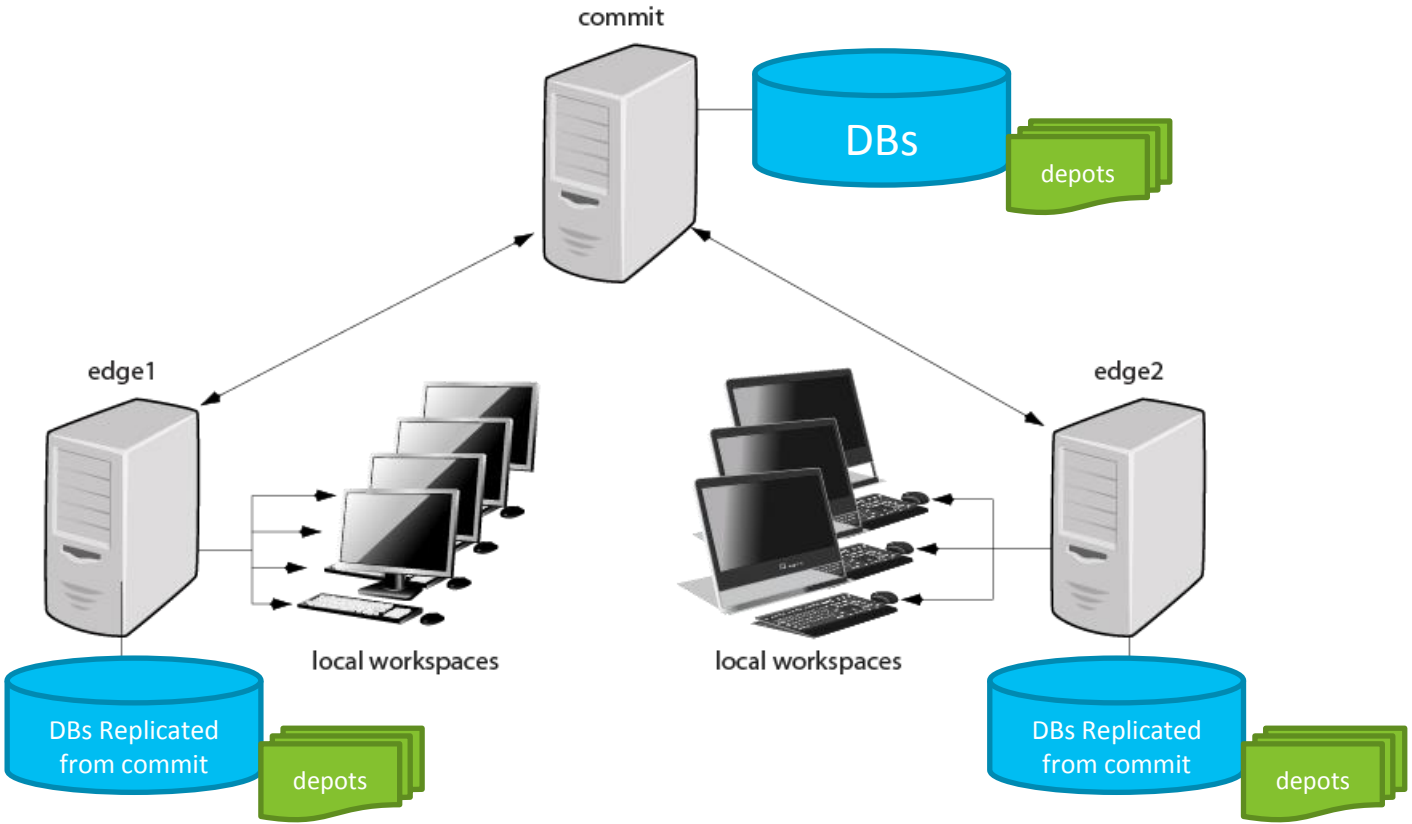


Edge / Commit – introduction

PERFORCE
Version everything.

- Commit server – Stores archives and permanent metadata
- Edge server(s) – Replicated copy of commit server and local work in progress
- Good for geographically-distributed work groups
- Performance advantages
  - Most user operations use a local server
- Builds upon Perforce replication technology

Edge / Commit – introduction

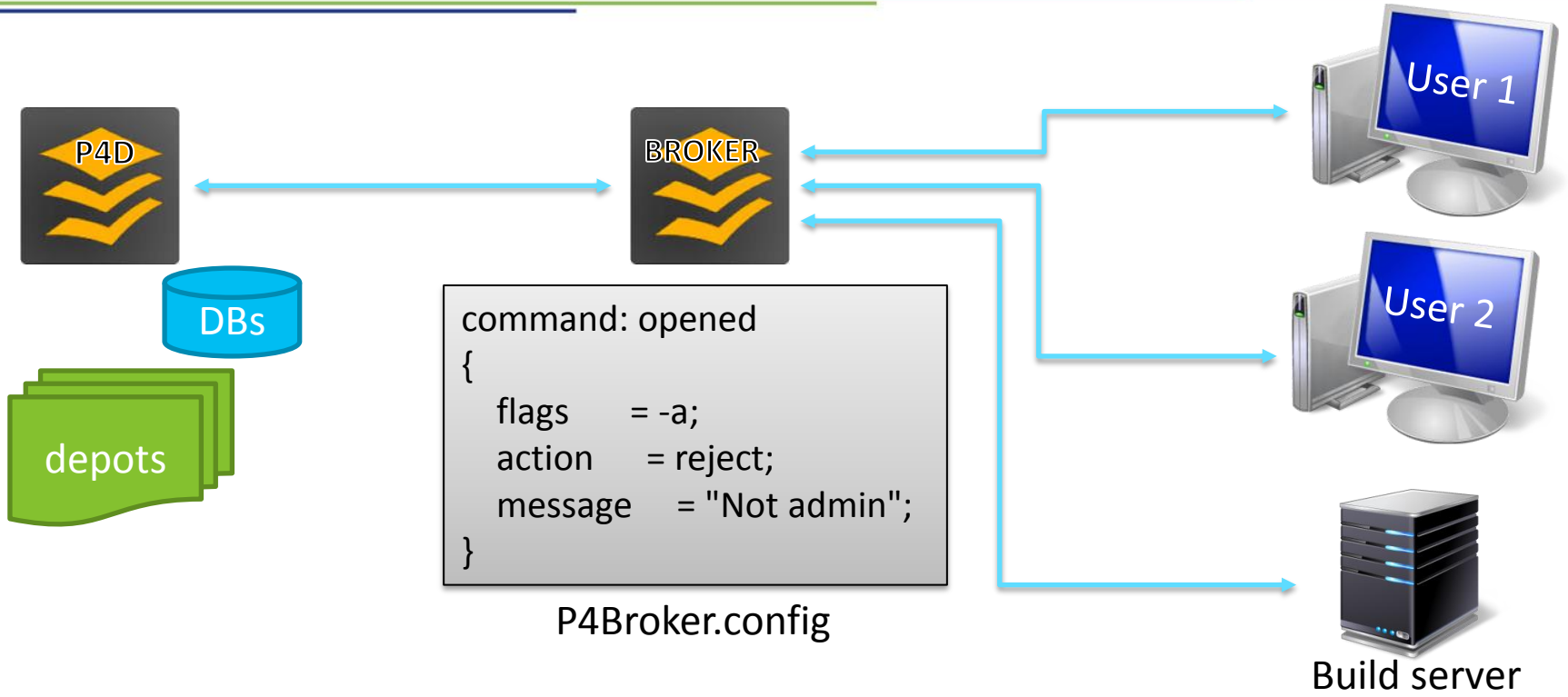


P4Broker

PERFORCE
Version everything.

- Intercepts all incoming Perforce commands
- Command handling support:
  - Redirection
  - Blocking
  - Rewriting (undocumented)
- Great for notifying users when the server is down for maintenance.
- Sometimes used as part of HA/DR strategies to avoid DNS change delay.

Perforce Broker



P4Broker Use Cases

PERFORCE
Version everything.

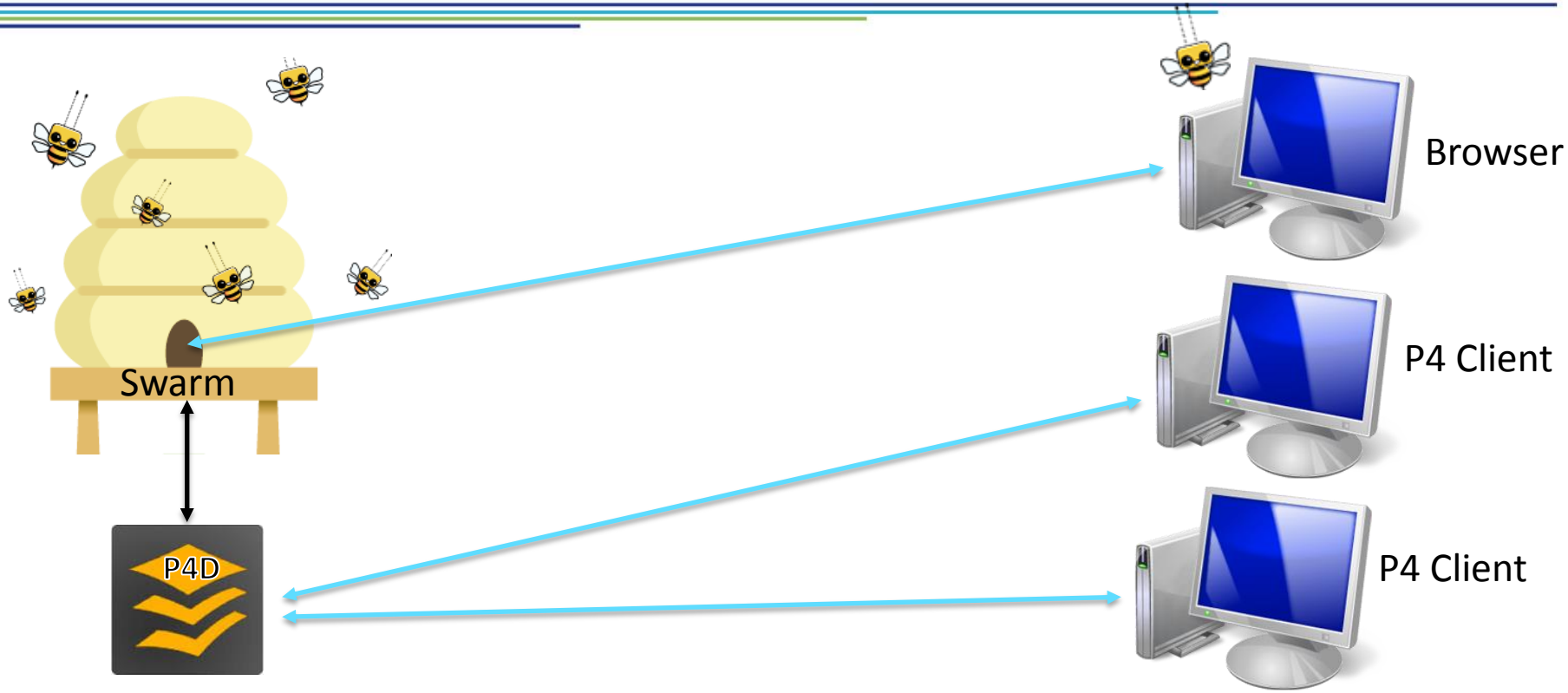
- Policy Customizations
  - different capabilities than triggers
- Traffic Redirection for Load Distribution
  - not “load balancing”
- Traffic Redirection for execution of automated failover operations
  - advanced/custom usage

Perforce Swarm

PERFORCE
Version everything.

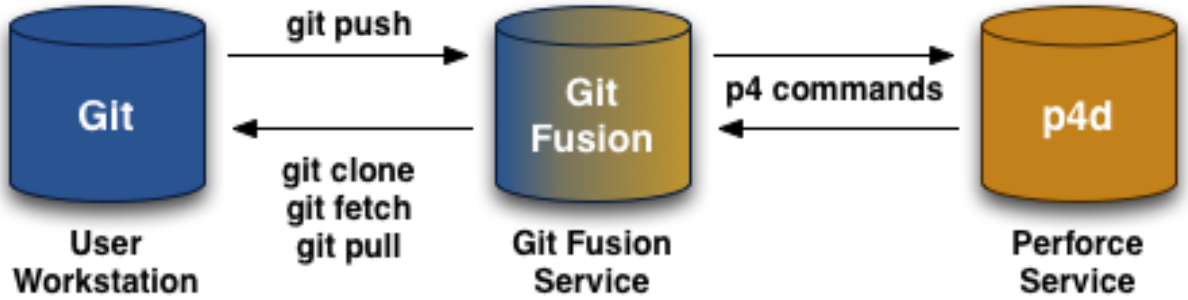
- Collaborative development
- Review project history
- Interactive social experience
- Code reviews / check-in approval
- Convenient web interface

Perforce Swarm

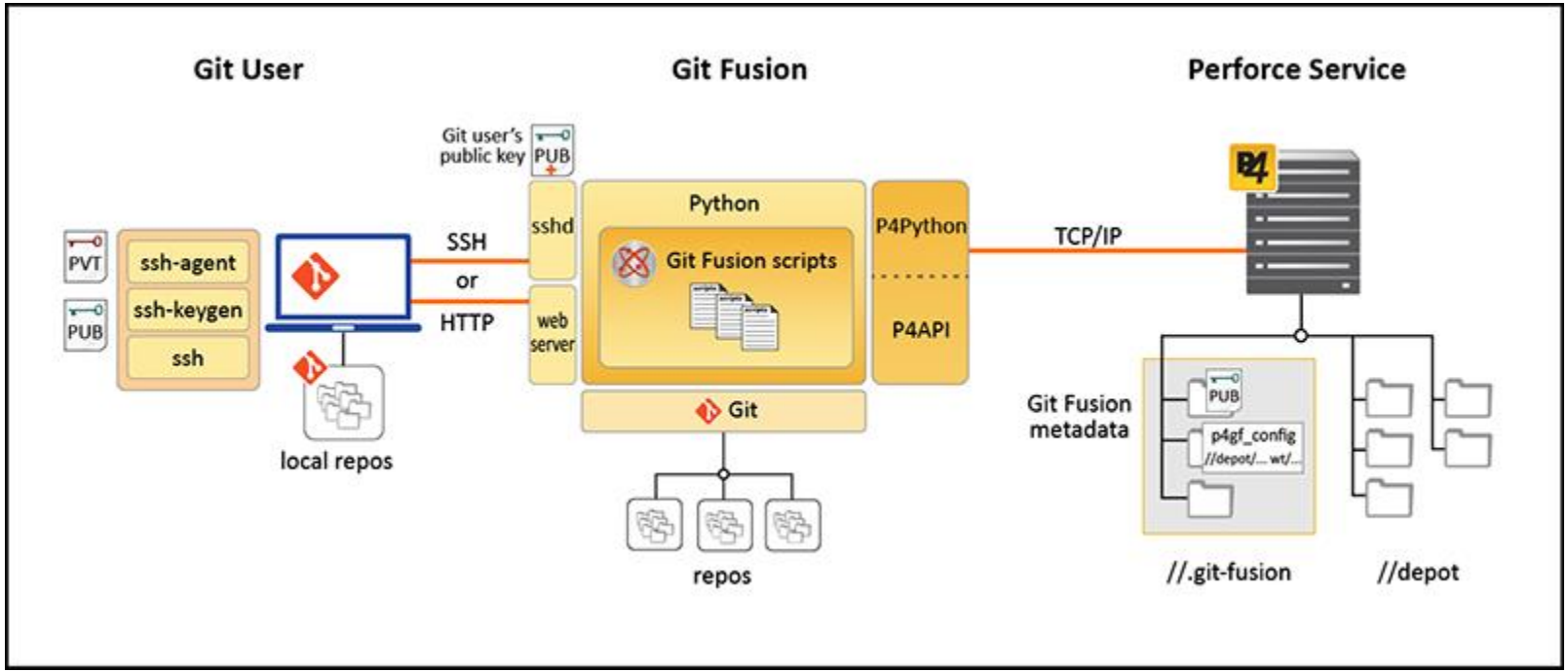


P4 Git Fusion

- Git remote repository service that uses Perforce Server as its back end.
- Looks like Git to Git users (because it is)
- Looks like Perforce to Perforce users (because it is)

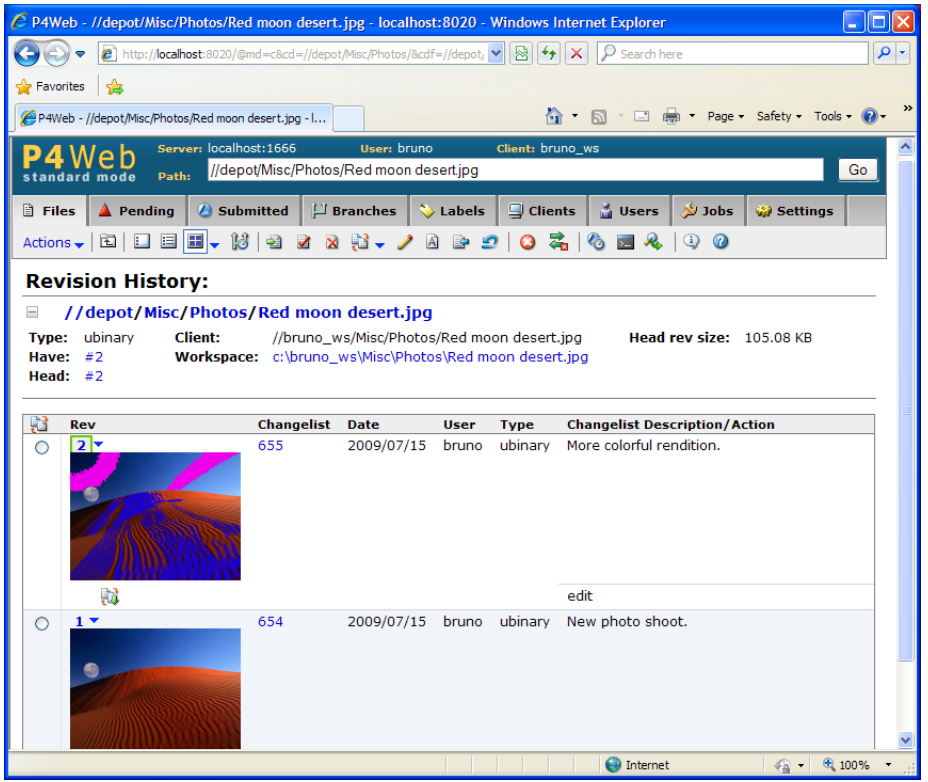


P4 Git Fusion – detailed architecture



P4Web

- Web interface client to Perforce
- Two modes:
  - Viewer mode (read-only)
  - Standard mode (read-write)
- Great for providing links to changes and jobs
- Now Largely supplanted by Swarm
- Open sourced – in the Workshop



Questions?

The End

PERFORCE
Version everything.

All Perforce manuals and technical notes are available at
www.perforce.com

Follow and participate with the Perforce Community and Forums at
www.perforce.com/community
workshop.perforce.com

Report problems and get technical help from support@perforce.com