GIT WITH BIND

DNS zones and revision control

GIT WITH BIND

Manage your bind DNS zones and configuration with git.

How I learned simply checking in the zone files is insufficient.

How I harnessed git-hooks to enforce the use of revision control and automate checking configuration and zone files.

J.P. MCGLINN

System Administration

Extensive K-12 Environment Experience

Data Management and Data System Integration

Software Defined Storage

BIND

Sample Workflow

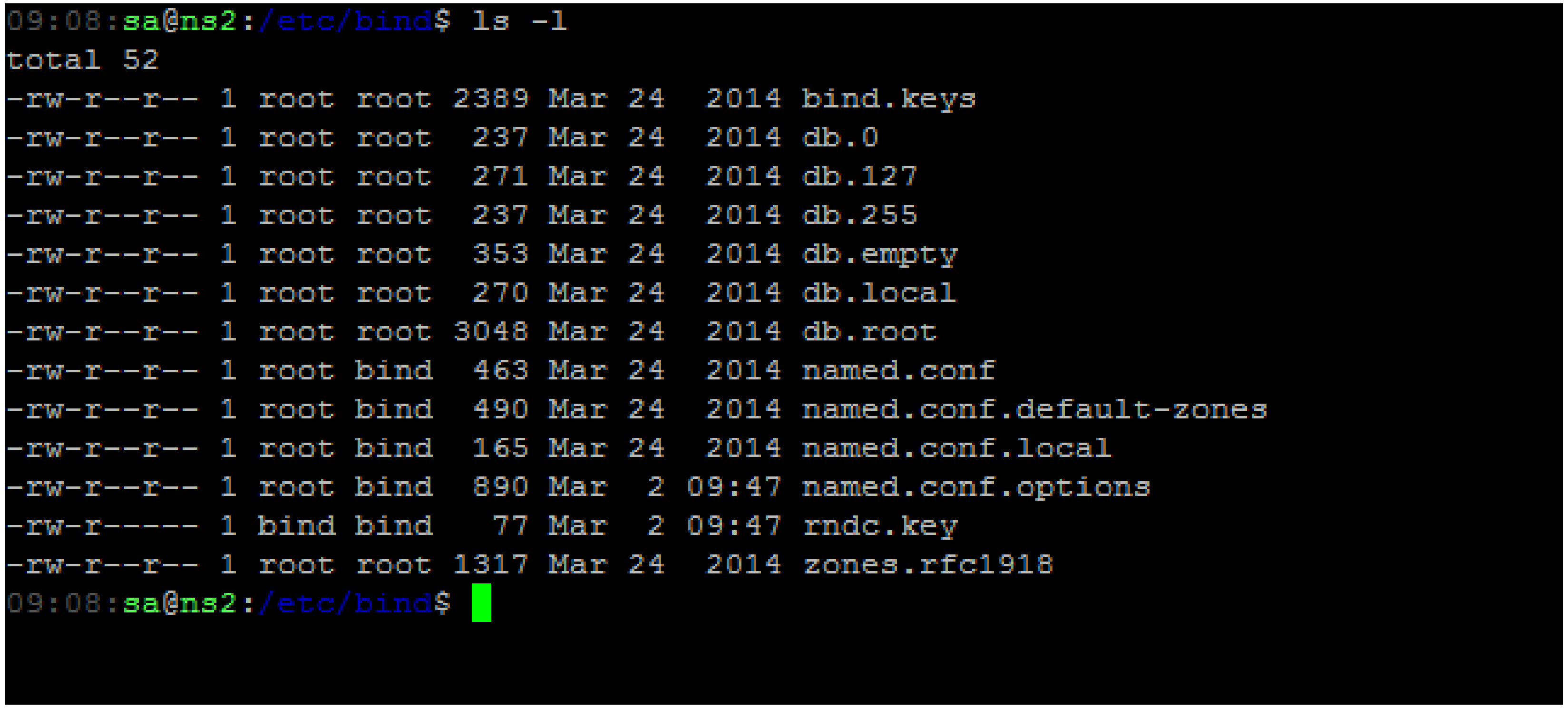
BIND

Domain Name System server

Lots of choices, Microsoft, Appliances, etc.

Internet Systems Consortium

BIND SAMPLE WORKFLOW



```
09:08:sa@ns2:/etc/bind$ ls -l
total 52
-rw-r--r-- 1 root root 2389 Mar 24  2014 bind.keys
-rw-r--r-- 1 root root  237 Mar 24  2014 db.0
-rw-r--r-- 1 root root  271 Mar 24  2014 db.127
-rw-r--r-- 1 root root  237 Mar 24  2014 db.255
-rw-r--r-- 1 root root  353 Mar 24  2014 db.empty
-rw-r--r-- 1 root root  270 Mar 24  2014 db.local
-rw-r--r-- 1 root root 3048 Mar 24  2014 db.root
-rw-r--r-- 1 root bind  463 Mar 24  2014 named.conf
-rw-r--r-- 1 root bind  490 Mar 24  2014 named.conf.default-zones
-rw-r--r-- 1 root bind  165 Mar 24  2014 named.conf.local
-rw-r--r-- 1 root bind  890 Mar  2 09:47 named.conf.options
-rw-r----- 1 bind bind   77 Mar  2 09:47 rndc.key
-rw-r--r-- 1 root root 1317 Mar 24  2014 zones.rfc1918
09:08:sa@ns2:/etc/bind$
```

BIND SAMPLE WORKFLOW

When(request)

- edit named.conf
- edit zonefile(s)
- restart bind9

Celebrate with a walk to the coffee shop

BIND SAMPLE WORKFLOW

When(request)

- edit named.conf
- edit zonefile(s)
- restart bind9

~~Celebrate with a walk to the coffee shop~~

Receive call while walking to the coffee shop

BIND SAMPLE WORKFLOW

When(request)

- edit named.conf
- edit zonefile(s)
- run named-checkconf
- run named-checkzone
- restart bind9
- test name server availability locally and from remote shell
- check secondary servers updated

Celebrate with a walk to the coffee shop

BIND SAMPLE WORKFLOW

Service Owner: Can you change that back? My servers weren’t ready.

Application Owner: You changed the wrong name, change it back right away.

IT Director: What server were we using for mail 27 months ago?

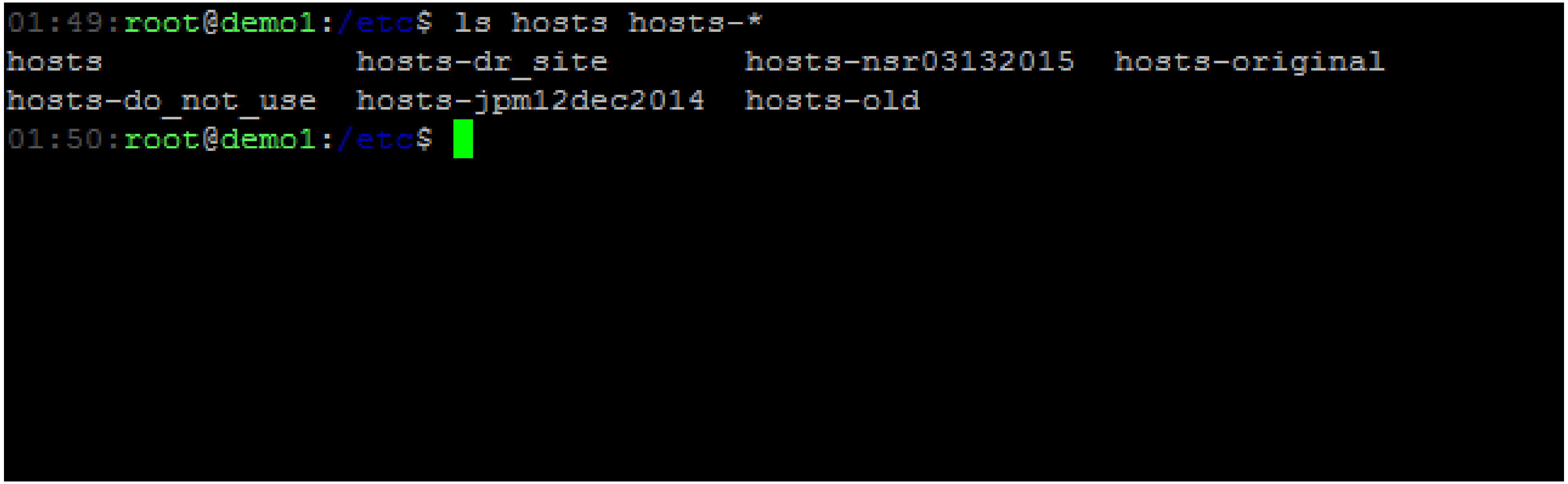
REVISION CONTROL SYSTEMS

Keeping files safe

REVISION CONTROL SYSTEMS

In computer software engineering, revision control is any kind of practice that tracks and provides control over changes to source code.

REVISION CONTROL SYSTEMS



```
01:49:root@demo1:/etc$ ls hosts hosts-*
hosts                 hosts-dr_site        hosts-nsr03132015  hosts-original
hosts-do_not_use      hosts-jpm12dec2014   hosts-old
01:50:root@demo1:/etc$
```

REVISION CONTROL SYSTEMS

SCCS

RCS

CVS

Perforce

SVN

BZR

Git

TFS

Mecurial

Bitbucket

CodePlex

GitHub

Google Code

SourceForge

REVISION CONTROL SYSTEMS

SCCS, 1972

RCS, 1982

CVS, 1990

Perforce, 1995

SVN, 2000

BZR, 2005

Git, 2005

TFS, 2005

Mecurial, 2005

Bitbucket

CodePlex

GitHub

Google Code

SourceForge

GIT

Git is a distributed revision control system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. Git was initially designed and developed by Linus Torvalds for Linux kernel development in 2005, and has since become the most widely adopted version control system for software development.

HTTP://EN.WIKIPEDIA.ORG/WIKI/GIT_(SOFTWARE)

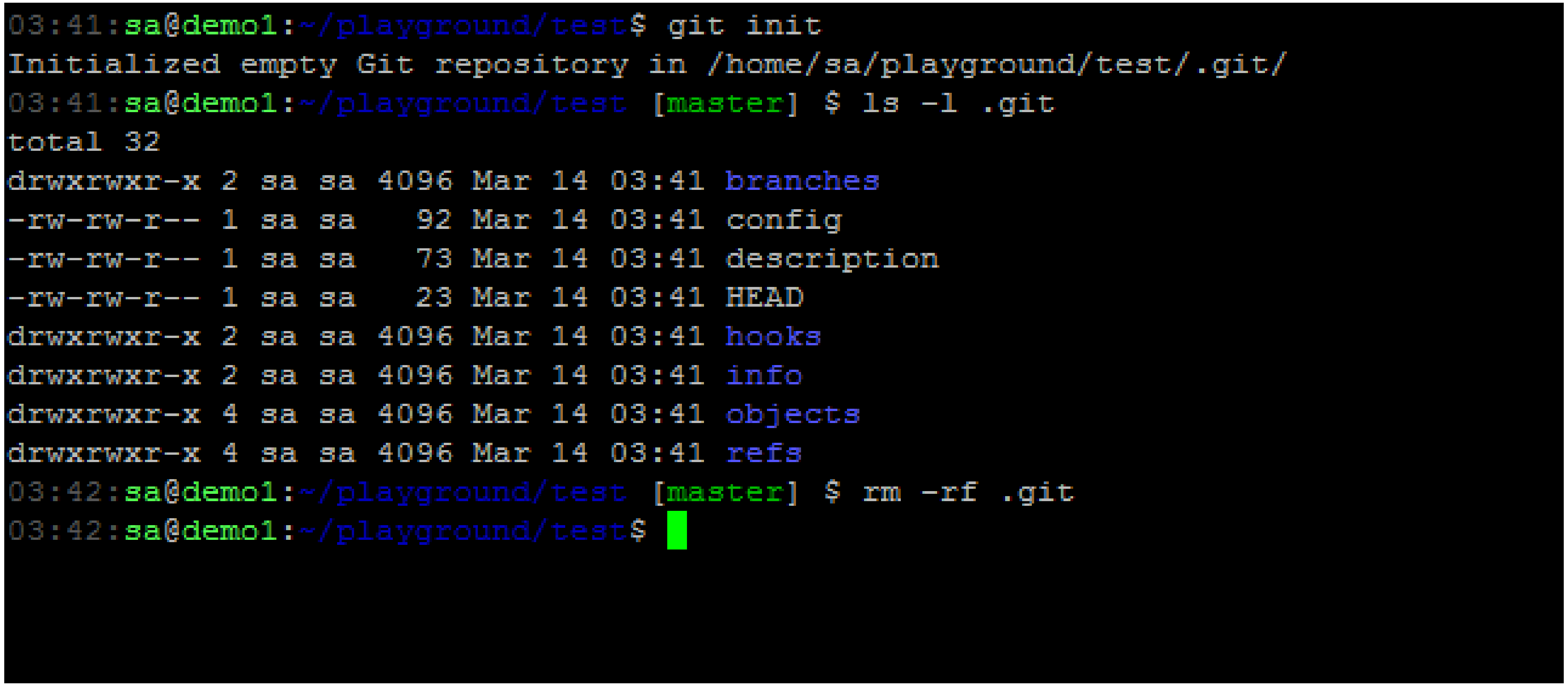
GIT

Easy to create a repository out of an existing directory

- git init
- git add <file>
- git commit –m ‘Initial Commit’

Distributed, anyone who clones the repository is a backup of the entire repository.

GIT



```
03:41:sa@demo1:~/playground/test$ git init
Initialized empty Git repository in /home/sa/playground/test/.git/
03:41:sa@demo1:~/playground/test [master] $ ls -l .git
total 32
drwxrwxr-x 2 sa sa 4096 Mar 14 03:41 branches
-rw-rw-r-- 1 sa sa   92 Mar 14 03:41 config
-rw-rw-r-- 1 sa sa   73 Mar 14 03:41 description
-rw-rw-r-- 1 sa sa   23 Mar 14 03:41 HEAD
drwxrwxr-x 2 sa sa 4096 Mar 14 03:41 hooks
drwxrwxr-x 2 sa sa 4096 Mar 14 03:41 info
drwxrwxr-x 4 sa sa 4096 Mar 14 03:41 objects
drwxrwxr-x 4 sa sa 4096 Mar 14 03:41 refs
03:42:sa@demo1:~/playground/test [master] $ rm -rf .git
03:42:sa@demo1:~/playground/test$
```

BIND + GIT

Sample Workflow

BIND + GIT SETUP

```
# cd /etc/bind
# git init
# git add *
# git commit –m ‘WOOHOO, finally have version control!’
```

BIND + GIT WORKFLOW

When(request)

- edit named.conf
- edit zonefile(s)
- run named-checkconf
- run named-checkzone
- git add <changed files>
- git commit
- restart bind9
- test name server availability locally and from remote shell
- check secondary servers updated

Celebrate with a walk to the coffee shop

BIND + GIT WORKFLOW

When(request)

- edit named.conf
- edit zonefile(s)
- run named-checkconf
- run named-checkzone
- git add <changed files>
- git commit
- restart bind9
- test name server availability locally and from remote shell
- check secondary servers updated

Celebrate with a walk to the coffee shop

BIND + GIT WORKFLOW

Service Owner: Can you change that back? My servers weren't ready.

Application Owner: You changed the wrong name, change it back right away.

- git revert HEAD
- update serial (maybe change the correct RR)
- restart bind9

IT Director: What server were we using for mail 27 months ago?

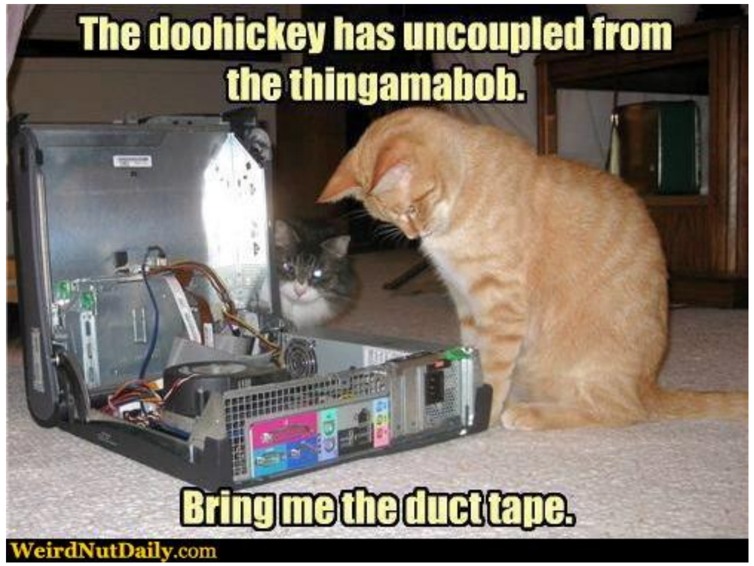
- git log
- git checkout <commit from 30 months ago>
- git checkout masster

BIND + GIT WORKFLOW DOWNFALLS

Easy to fall back to no revision control

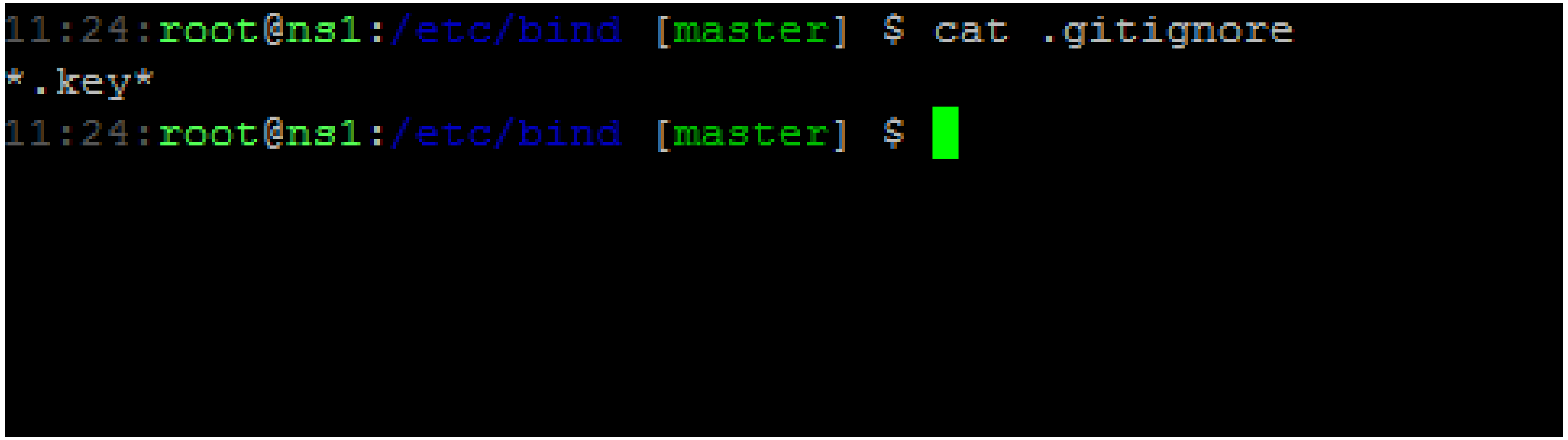
Using /etc/bind directory to browse history is dangerous

Checked in secrets



.GITIGNORE

.gitignore file tells Git to ignore files based on a list of patterns



```
11:24:root@ns1:/etc/bind [master] $ cat .gitignore
*.key*
11:24:root@ns1:/etc/bind [master] $ 
```

GIT HOOKS

Workflow helpers

GIT HOOKS

Git hooks are scripts that run automatically every time a particular event occurs in a Git repository. They let you customize Git's internal behavior and trigger customizable actions at key points in the development life cycle.

HTTPS://WWW.ATLASSIAN.COM/GIT/TUTORIALS/GIT-HOOKS/

GIT HOOKS

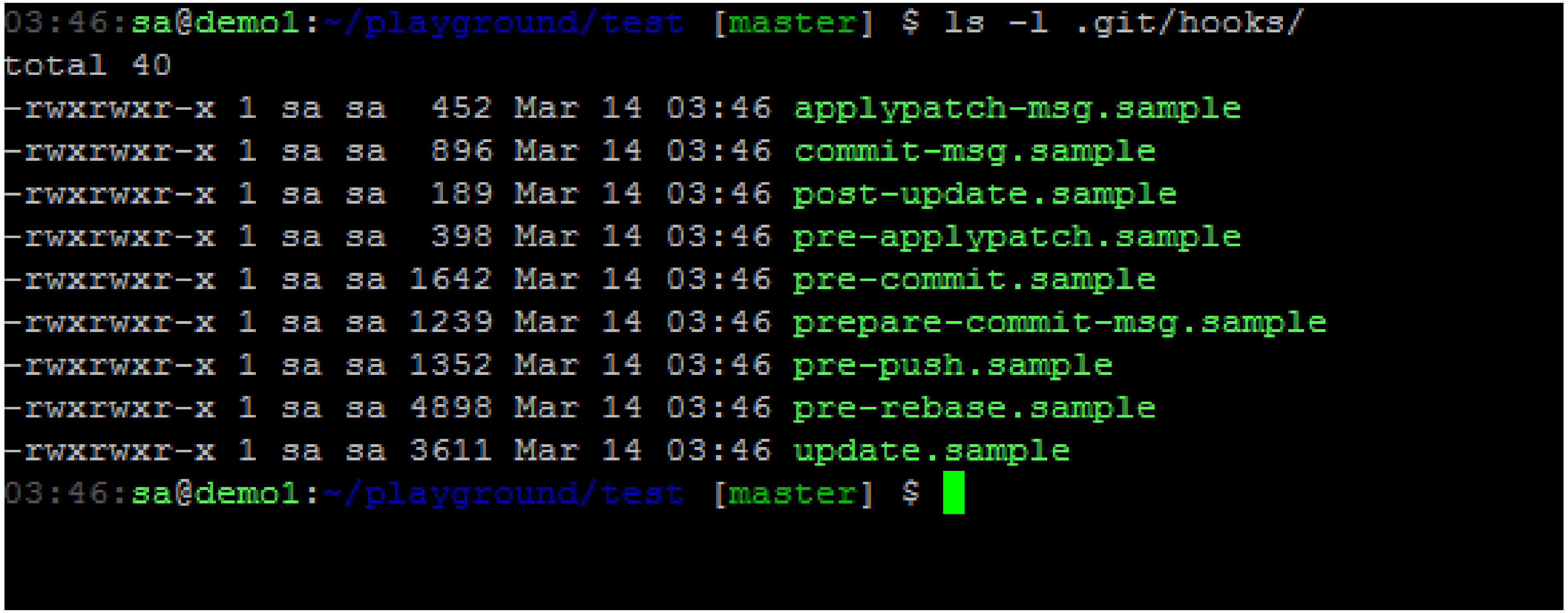
Hooks are scripts that live in .git/hooks
- do not get cloned when cloning the repo

Samples (*.sample)

pre-commit
- Check that my files are valid

post-commit
- Reload bind

GIT HOOKS



```
03:46:sa@demo1:~/playground/test [master] $ ls -l .git/hooks/
total 40
-rwxrwxr-x 1 sa sa  452 Mar 14 03:46 applypatch-msg.sample
-rwxrwxr-x 1 sa sa  896 Mar 14 03:46 commit-msg.sample
-rwxrwxr-x 1 sa sa  189 Mar 14 03:46 post-update.sample
-rwxrwxr-x 1 sa sa  398 Mar 14 03:46 pre-applypatch.sample
-rwxrwxr-x 1 sa sa 1642 Mar 14 03:46 pre-commit.sample
-rwxrwxr-x 1 sa sa 1239 Mar 14 03:46 prepare-commit-msg.sample
-rwxrwxr-x 1 sa sa 1352 Mar 14 03:46 pre-push.sample
-rwxrwxr-x 1 sa sa 4898 Mar 14 03:46 pre-rebase.sample
-rwxrwxr-x 1 sa sa 3611 Mar 14 03:46 update.sample
03:46:sa@demo1:~/playground/test [master] $
```

http://www.git-scm.com/book/en/v2/Customizing-Git-Git-Hooks

GIT HOOKS PRE-COMMIT

```
#!/bin/sh
#
# Redirect output to stderr.
exec 1>&2

named-checkconf -z named.conf > /dev/null

if [ $? -ne 0 ]; then
  named-checkconf -z named.conf
  echo "\n=================================================================="
  echo "Configuration or zonefiles don't pass named-check. Commit Canceled"
  echo "=================================================================="
  exit 1
fi
```

GIT HOOKS POST-COMMIT

```
#!/bin/sh
#
# Redirect output to stderr.
exec 1>&2

service bind9 reload

if [ $? -eq 0 ]; then
  echo "\n=================================================================="
  echo "=================================================================="
  echo "                  New Configuration Loaded"
  echo "=================================================================="
  echo "=================================================================="
  exit 0
fi

exit 1
```

BIND + GIT + HOOKS WORKFLOW

When(request)

- edit named.conf
- edit zonefile(s)
- ~~run named-checkconf~~
- ~~run named-checkzone~~
- git add <changed files>
- git commit
- ~~restart bind9~~
- test name server availability locally and from remote shell
- check secondary servers updated

Celebrate with a walk to the coffee shop

BIND + GIT + HOOKS WORKFLOW

When(request)

- edit named.conf
- edit zonefile(s)
- ~~run named-checkconf~~
- ~~run named-checkzone~~
- git add <changed files>
- git commit
- ~~restart bind9~~
- test name server availability locally and from remote shell
- check secondary servers updated

Celebrate with a walk to the coffee shop

NEXT STEPS

Now that we have a good workflow, what else can we do?

- More strict checking, running checkconf doesn’t stop on zone warnings
- Check for updated serial number in changed zones
- Remote Checkout
- Store Secondary configs, update secondary server configs automatically
- Email the team
- Slack/HipChat/IRC updates

RESOURCES

etckeeper

http://etckeeper.branchable.com/

GitZone

https://www.dyne.org/software/gitzone/

RESOURCES

http://en.wikipedia.org/wiki/Revision_control

http://en.wikipedia.org/wiki/Git_(software)

http://www.git-scm.com/book/en/v2

https://www.atlassian.com/git/tutorials/

https://help.github.com/

J.P. MCGLINN

github.com/jpbot

@pilotlamp