Course 1 - Devops Certification Training

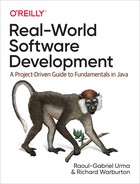
# Books / References / Links

**Books**

* **Effective DevOps** by Jennifer Davis, Ryn Daniels  
  [O'reilly learning link](https://learning.oreilly.com/library/view/effective-devops/9781491926291/)
* **Real world software development** by Raoul-Gabriel Urma, Richard Warburton

[O'reilly learning link](https://learning.oreilly.com/library/view/real-world-software-development/9781491967164/)





**DevOps, how it all started**

* <https://blog.newrelic.com/engineering/devops-name/>
* Original flickr presentation in 2007 : <https://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr>

**Keep up with Devops**

* [2020 status of devops report](https://puppet.com/resources/report/2020-state-of-devops-report/)

**Devops Use cases**

* [16 Popular DevOps Use Cases & Real Life Applications [2021]](https://www.upgrad.com/blog/devops-use-cases-applications/)
* <https://dzone.com/articles/devops-use-cases>
* [The Most Logical Federal Use Cases for DevOps](https://fedtechmagazine.com/article/2020/03/most-logical-federal-use-cases-devops)
* How amazon embraced devops
  + <https://api-university.com/blog/the-api-mandate/> - Bezos memo

**DevSecOps**

* [What is devsecops](https://www.redhat.com/en/topics/devops/what-is-devsecops)

## Handy Links

**Cheat Sheets**

Linux

* Simplilearn linux cheatsheet (LMS → Self paced learning → at the bottom)
* <https://www.guru99.com/linux-commands-cheat-sheet.html>
* <https://phoenixnap.com/kb/linux-commands-cheat-sheet>

Git

* [Git cheat sheet](https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet)

# DevOps practitioners' guide



# How Software is Delivered



# Tools Being Used

* JIRA
* GIT
* Jenkins
* SonarQube
* Mercurial
* Bitbucket / github
* Confluence
* Docker
* Kubernetes
* NExus
* Ansible
* VS Code
* Coverity
* SVN
* NET core

# Incorporating Security as part of Dev Cycle

Web security scanner : <https://www.zaproxy.org/> (ZED)

Check for SQL injection vulnerabilities



**DevSecOps in Logging**

Sensitive data can leak in logging

Printing out the response

User: teddy123 logged in with password: iLikeHugs

# Version Control System Architectures

**Centralized Server - Clients**



**Distributed System**



# Git

[Linus explains how it all started](https://www.youtube.com/watch?v=4XpnKHJAok8) - Google talk video

**Git Architecture**

Git tracks **content** (collection of files).. not individual files.

It checksums the content



## Git clients

* Git command line
* Mac
  + Gitx
  + Gitkraken
  + Git desktop
* Windows
  + Git tree
  + Git shell / git bash
  + Git desktop
* Other
  + Tortoise git

**Diff tools**

Delta walker (win / mac / linux)

* Can diff office files

Mac

* Diff merge 3
* Source tree

Ubuntu

* Meld

**Github account setup**



## Lab git-1 : checking into github

**Step-1 : Sign into github and create a project**

You can keep the project public for now.

[x] Add a 'README.md' file

If using btibucket…

* Sign in
* Create a project

**Step 2 - on our ubuntu linux virtual machine**

Configuring Git, on a terminal

$ git config --global user.email "sujee@elephantscale.com"

$ git config --global user.username "Sujee Name"

$ cat ~/.gitconfig # to very

**Step 3 - Clone the repo (linux desktop)**

$ git clone <https://github.com/sujee/devops-test>

**Step 4 - add some files**

$ ls

$ cd first-project

$ nano a.txt

# add some text content to a.txt

**Step 5 - add file to local repo**

$ git status

$ git add a.txt

$ git commit -m 'adding a file' a.txt

**Step 6 - push/publish (local repo → github)**

$ git push

It will ask you for password

**Step 7 - verify on github**

Go to github.com and see if your file showed up

# SSH Keys



**Step 7:** Creating SSH Key and adding it to GitHub

● Switch the current directory to ssh by executing below command:

**cd ~/.ssh**

● Generate an RSA key for the registered email Id (An example is available below)

**~~ssh-keygen -t rsa -C <YourEmailId>~~**

**ssh-keygen**

**~~vi id\_rsa.pub~~**

**cat id\_rsa.pub**

You can skip 'add origin' command

Install tree command

sudo apt install tree

Lock key = Priv + pub keys

Once you added ssh keys, re-clone using SSH URL

git clone [git@github.com](mailto:git@github.com):sujee/test1.git test1b

**How to specify different key than id\_rsa ?**

Create a file **~/.ssh/config** with the following

Host github.com

HostName github.com

PreferredAuthentications publickey

IdentityFile ~/.ssh/deploy-key1.pem

**Git UI on Linux**

sudo apt install gitk

gitk

# Git Branching



**Branching strategy**

Good reads

* <https://medium.com/@patrickporto/4-branching-workflows-for-git-30d0aaee7bf> - good comparison of all models
* <https://stackoverflow.com/questions/35035843/git-branching-strategies-for-cicd>
* <https://nvie.com/posts/a-successful-git-branching-model/>

Gitflow based

* <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

\*\*Main line / trunk based

<https://wiki.faa.gov/display/AH/Branching+Standards>

Read more :

* <https://docs.microsoft.com/en-us/azure/devops/repos/git/git-branching-guidance?view=azure-devops>

**Git hosting**

https://gitolite.com/gitolite/index.html

# End day 1

**Lab : bitbucket**

Setup a repo on bitbucket

Pull

Commit

push

**BONUS Lab : reverting a change**

commit 1 : added a couple of lines

git commit

git push

commit 2 : you deleted these

git commit

git push

commit 3 : go back to commit-1

git log

# copy the SHA

git checkout SHA a.txt

git commit

git push

[End of day - 1 survey](https://www.surveymonkey.com/r/NBKJBF7?trainer=akshatha&company=XanderTalent&milestone_id=00060315_01&course=tableau%20training&geo=ROW&type=ELVC&Webex_Session_ID=1849146527)

**Sujee's ~/.gitconfig**

[user]

name = Sujee Maniyam

email = [sujee@sujee.net](mailto:sujee@sujee.net)

[color]

status = auto

diff = auto

#pager=true

branch = auto

interactive = auto

[alias]

st = status

ci = commit

co = checkout

br = branch

pr = pull --rebase

[merge]

tool = opendiff

[push]

default = current

[pull]

#rebase = true

[core]

pager = "less -F -X -M -R"

excludesfile = /Users/sujee/.gitignore\_global

-----