# DevOps Demo

## Why devops?

* DevOps is a combination of set of tools, principles, philosophies where organizations want to achieve rapid and frequent release.

How applications are developed and released without devops.

* Waterfall model

1. Requirement gathering
2. Analysis
3. Design
4. Code/Implementation
5. Testing
6. Deployment → Prod

Note : 3 months, 6moths, 1 year release

Problems:

1. Longer SDLC cycles
2. Takes lots of time for integration and fixing bugs
3. Release are not reliable and buggy.
4. Less productivity
5. There was no automation so lots of waiting periods between phases.
6. No collaboration between dev and ops teams.

What we do in DevOps?

* Keeps SDLC cycles short(Time to the market is fast)
* Integrate code continuously (CI)
  + Increases productivity
* Collaboration between dev,testing and ops teams.
* Automate everything
* Centralized logging with alerts (Splunk, ELk, GrayLogs, etc…)
* Monitoring CPU, RAM, Disk, IO etc…(Nagios, Zabbix, Geneos, Grafana, Splunk, ELK)
* Configuration management (Ansible, Chef, Puppet etc)
* Microservices and docker with kubernetes orchestration tool

1. Git,
2. Build tool - Maven
3. Tomcat - Web server
4. **Jenkins - groovy scripting**
5. Sonatype Nexus - private artifactory store (docker images, jar, war, etc..)
6. SonarQube - static code analysis
7. **Ansible** - configuration management tool
8. **Docker and Kubernetes and Microservices**
9. Linux essentials for DevOps
10. Nagios monitoring
11. ELK - log management - (Optional)

# SCM (Source Code Management) Tool

## Why to use SCM tool?

1. Allows multiple developers to collaborate and work on a single project.
2. SCM tools keep track of all changes happening to your files.
   1. In case of defects we can fallback to previous version.
   2. It helps in troubleshooting errors by tracing recent changes made to code.
3. SCM servers a backup of your source code.
4. It secures source code be having authentication and authorization.
5. It acts as central code base for your project

## Who should use SCM ?

* Everyone

## SCM tools in the market

1. Git
2. SVN
3. Clearcase
4. TFS (Team Foundation Server) Microsoft
5. Etc..

## Which SCM is trending now?

1. Git

## What are the features of Git?

1. Git is fast and distributed Version Control System
2. Git platform independent
3. Lots of features which can solve complex problems easily.

## Setup Git for our practice

Git remote servers can be setup in two ways

1. Hosted git (Installing and configuring Git on your own server)
2. Cloud based Git, third party companies provide Git as as service
   1. The companies providing git as a service
      1. GitHub (Microsoft)
      2. Bitbucket
      3. Gitlab
      4. Cloud Providers (AWS, Azure, GCP, etc…)

## Create account in GitHub (Remote Repository)

1. <https://github.com/>

## Create Git repository

* One repository represents a project

## Install Git client

* There are several git clients in the market
  + GUI(Graphical User Interface) clients
  + CLI (Command Line Interface) clients
* Install Gitbash on your laptop
  + While installing gitbash (optionally choose your own editor editor/ default is vim)

## Adding our content to repository

Git Clone: git clones remote repository and keeps it in your local machine

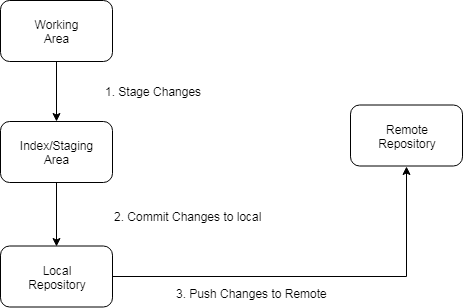
**Note:** clone is ontime activity, ones you have copy locally next time onwards instead of clone you will use pull.

* Open Gitbash (right click and onen where you want to clone)

git clone <https://github.com/javahomehari/devops-we-aug-2019>

Change your directory to repository

cd [devops-we-aug-2019](https://github.com/javahomehari/devops-we-aug-2019)



**Working Area:**

Git keep all your modifications(create/updating/deleting) to local repository in working area

**Index/Staging Area**:

Git uses this for the files that should be committed to local repository

* git add \*
* git add info.txt
* git add \*.txt

**Commit:**

Before we commit we have to configure user name and email using our git client

git config --global user.name "Hari kammana"

git config --global user.email "[hari.kammana@gmail.com](mailto:hari.kammana@gmail.com)"

The above information is stored in $USER\_HOME/.gitconfig

All files are picked form **index/staging** and commit to local repository

* git commit -m ‘MYAPP-12678 commit message’

Use git status at anypoint of time to check your working/staging area.

* git status

Check git history

* git log
* git log --oneline
* git log --oneline -3 (get recent 3 commits)

Display the files committed in specific commit id

* git show 6f1eb73 --name-only --pretty=""

Pushing commits in local repository to remote

* git push origin master

Git Pull

* Pull gets new commits from remote and **integrates/merge** with local
* git pull origin master

Git Fetch

* Fetch gets new commits from remote and will not **integrate/merge**
* git fetch
* git merge