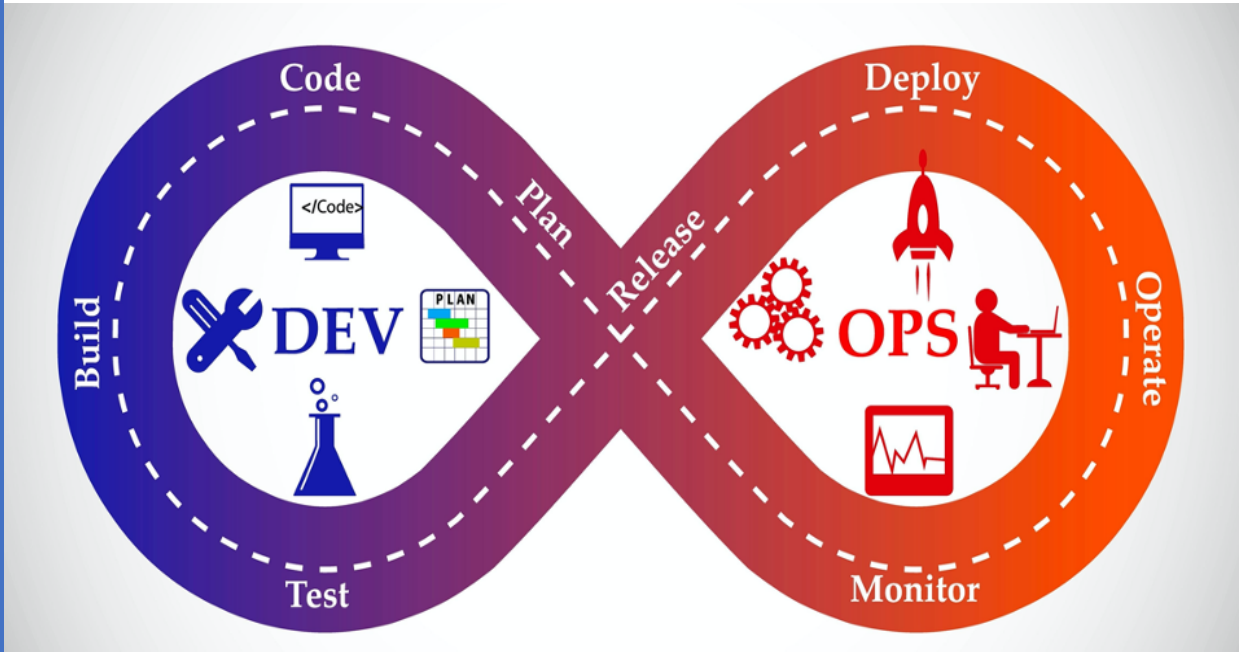
DevOps

COMP 3104

Lecture 01



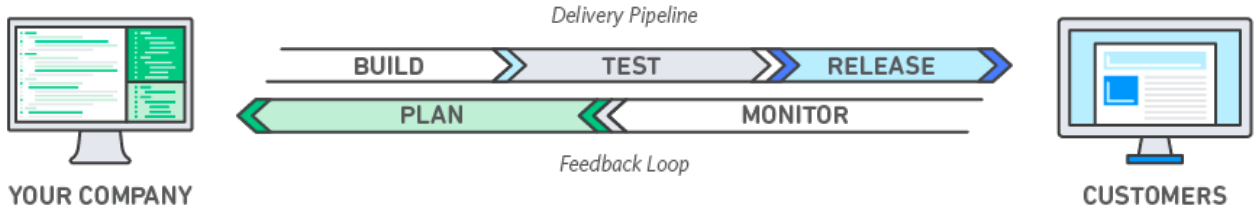
- Why DevOps?
- Future of DevOps
- What is a build pipeline?
- BASH Commands
- Installing Course Dependencies

What is DevOps ?

- DevOps is a new term emerging from the collision of two major related trends. The first was also called "**agile infrastructure**" or "**agile operations**"; it sprang from applying Agile and Lean approaches to operations work.

- DevOps is "*a cross-disciplinary community of practice dedicated to the study of building, evolving and operating rapidly-changing resilient systems at scale.*" - Jez Humble

- DevOps is a combination of software development and information technology operations that enables businesses to deliver applications at a **faster** pace. It brings together development and operations teams so there are fewer **redundancies** in the software development process.



Knows buzz words

- Waterfall
- Agile
- Sprint
- SDLC
- Scrum
- Scrum Master

Why DevOps adoption?

DevOps culture brings the cohesion of autonomous teams to fast paced, high-demanding environments.

It has a set of processes that when applied correctly can:

- *Drastically reduce IT overhead*
- *Remove manual tasks both of a mundane or complex nature*
- *Enforce best security practices*
- *Expedite product delivery*
- *Improve communication and collaboration across teams*

What is a build pipeline?

A **build pipeline** is the entity through which you define your automated **build pipeline**.

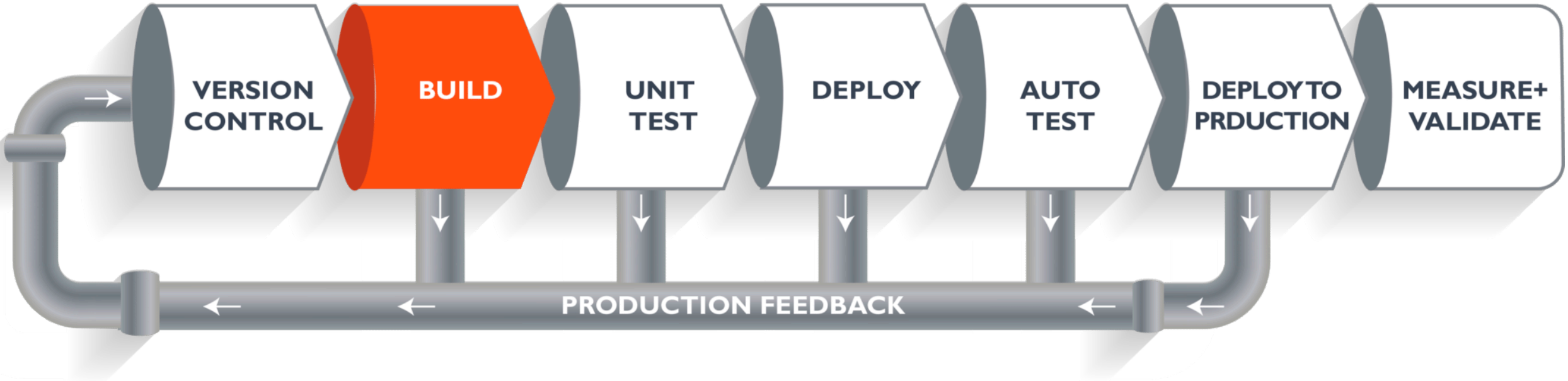
In the **build pipeline**, you compose a set of tasks, each of which perform a step in your **build**.

CI/CD Pipeline ?

A CI/CD Pipeline implementation, or **Continuous Integration/Continuous Delivery or Continuous Deployment**, is the backbone of the modern DevOps environment.

It bridges the gap between development and operations teams by automating the building, testing, and deployment of applications.

CI/CD Pipeline



Categories of Tools in Pipeline

A pipeline generally consists of a set of tools which are normally broken down into the following categories;

- Source Control
- Build tools
- Containerisation
- Configuration Management
- Monitoring/Feedback

Continuous Integration (CI) ?

Continuous Integration (CI) is a practice in which developers will check their code into a version-controlled repository several times per day. Automated build pipelines are triggered by these check ins which allow for fast and easy to locate error detection.

The key benefits of CI are:

- Smaller changes are easier to integrate into larger code bases.
- Easier for other team members to see what you have been working on
- Bugs in larger pieces of work are identified early making them easier to fix resulting in less debugging work
- Consistent code compile/build testing
- Fewer integration issues allowing rapid code delivery

Continuous Delivery (CD) ?

Continuous Delivery (CD) is the process which allows developers and operations engineers to deliver bug fixes, features and configuration changes into production reliably, quickly and sustainably. Continuous delivery offers the benefit of code delivery pipelines that are routinely carried out that can be performed on demand with confidence.

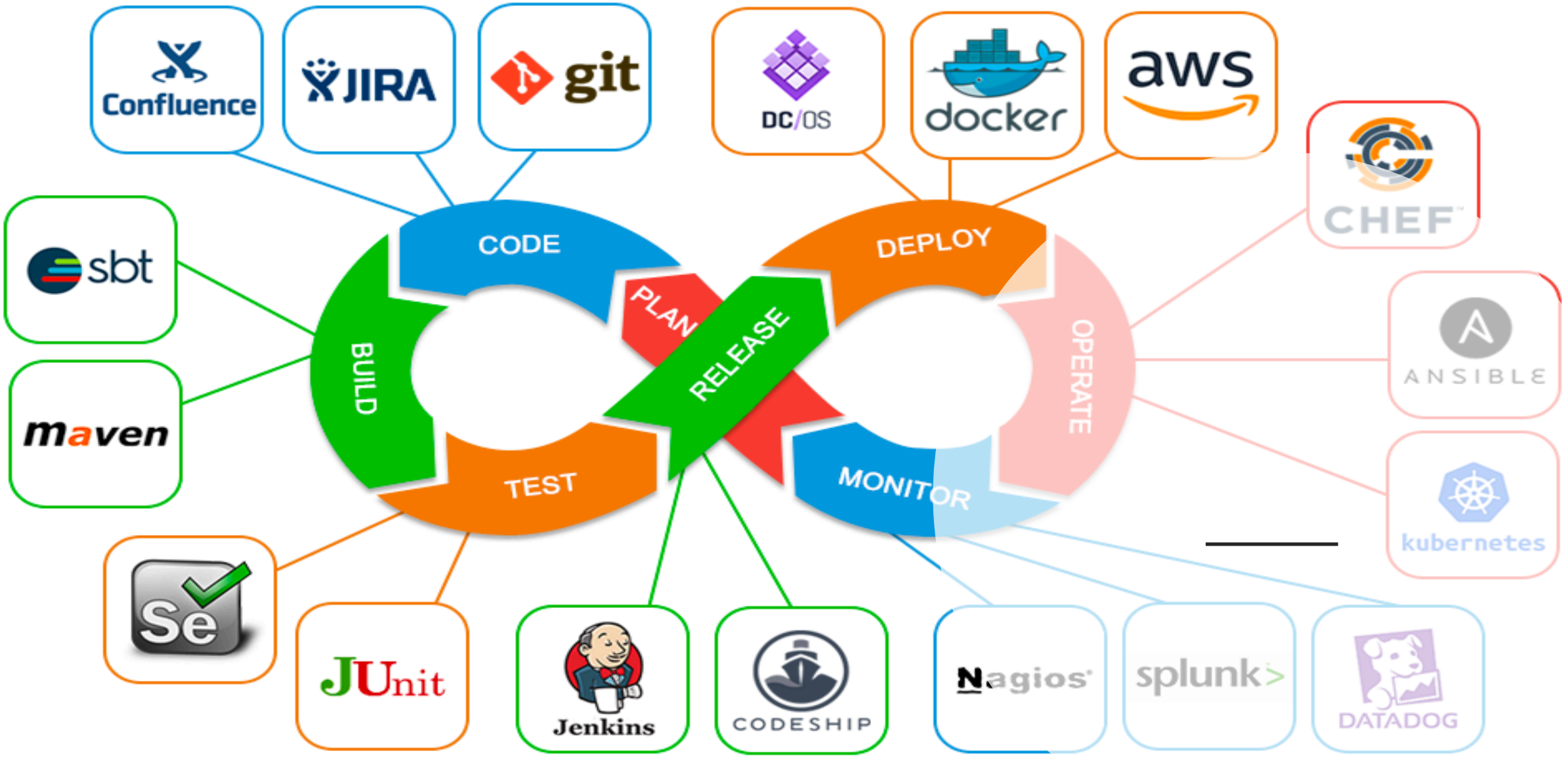
The benefits of CD are:

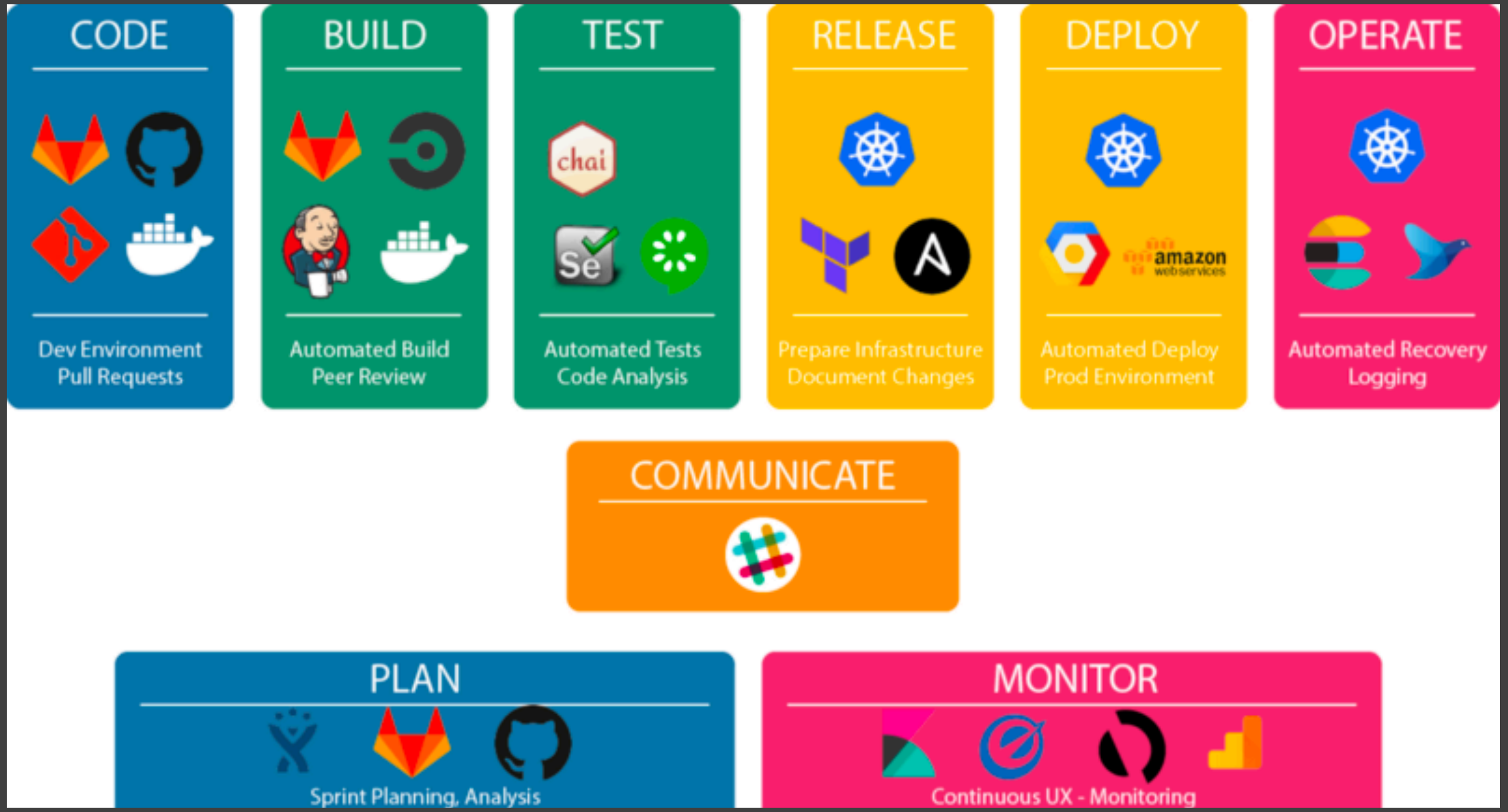
- **Lower Risk Releases** – Blue/Green deployments and canary releases allow for zero downtime deployments which are not detectable by users and make rolling back to a previous release relatively pain free.

- **Faster Bug Fixes & Feature delivery** – when features or bug fixes are finished and have passed the acceptance and integration tests – a CD pipeline allows these to be quickly delivered into production.

- **Cost savings** – Continuous Delivery allows teams to work on features and bug fixes in small batches which means user feedback is received much quicker.

Tools Involves in DevOps




CODE
Dev Environment
Pull Requests
BUILD
Automated Build
Peer Review
TEST
chai
Se
Automated Tests
Code Analysis
RELEASE
Prepare Infrastructure
Document Changes
DEPLOY
amazon
web services
Automated Deploy
Prod Environment
OPERATE
Automated Recovery
Logging
COMMUNICATE
PLAN
Sprint Planning, Analysis
MONITOR
Continuous UX - Monitoring

Case Study – DevOps

- **Netflix**

  https://www.youtube.com/watch?v=UTKIT6STSVM

- **Uber**

  https://www.youtube.com/watch?v=WDLPytojmck

BASH Command

- mkdir
- echo
- cd
- pwd
- man
- Create hidden file
- ls
- touch
- cat
- open
- vi
- which

BASH Command Reference

Command	What It Does
`pwd`	Prints the 'present working directory,' letting you know where you are.
`ls`	Lists the files in the current directory
`man *`	Lists the manual for the command, substituted for the `*`
`cd *`	Changes the current directory to `*`
`mkdir *`	Makes a directory named `*`
`open` or `explorer`	On OS X, `open` followed by a file opens it; in Windows, the command `exp` followed by a file name does the same thing.
`cat *`	`cat` is a versatile command. It will read a file to you if you substitute a file but can also be used to combine files.
`head *`	Displays the first ten lines of `*`
`tail *`	Displays the last ten lines of `*`
`mv`	Moves a file
`cp`	Copies a file
`rm`	Deletes a file
`vim`	Opens up the `vim` document editor.

More Reference: https://ss64.com/bash/

Installing Course Dependencies

Software	Installation Links
NodeJS	https://nodejs.org/en/download/
Git	https://git-scm.com/downloads
GitHub Desktop	https://desktop.github.com/
SourceTree	https://www.sourcetreeapp.com/
Jenkins	https://www.jenkins.io/download/
SonarQube	https://www.sonarqube.org/downloads/
Docker	https://www.docker.com/products/docker-desktop

Other required tools will be downloaded and setup whenever required

Working with git and GitHub

- Create account on GitHub - https://github.com/
- Setup Git on your local machine
- Working with Git CLI

  - *git init*
  - *git status*
  - *git add*
  - *git commit*
  - *git log*

***Next Lecture**: https://guides.github.com/introduction/git-handbook/*

Thank You