**Content**

[1. Concepts 2](#_Toc12870828)

[1. Continuous Integration 2](#_Toc12870829)

[2. Jenkins 5](#_Toc12870830)

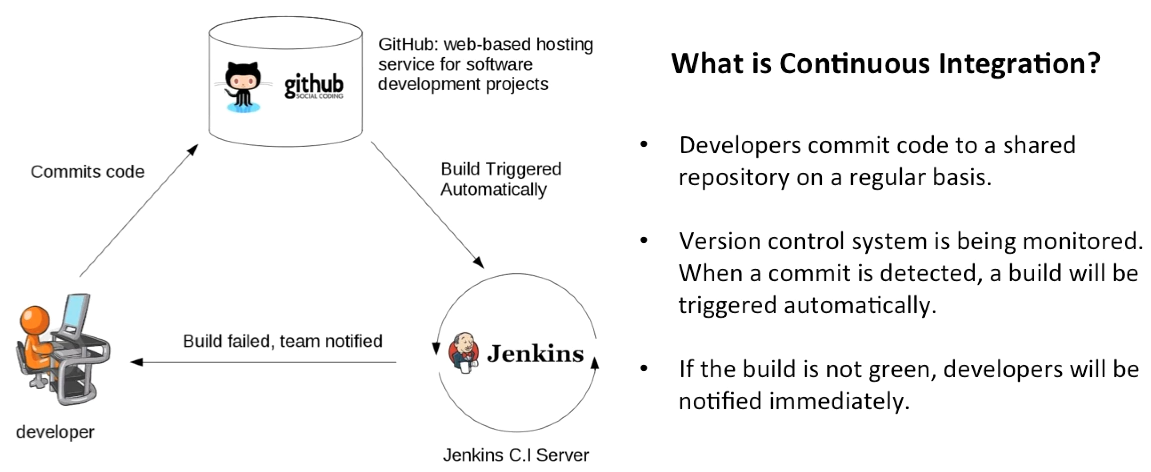
[1. Installation 5](#_Toc12870831)

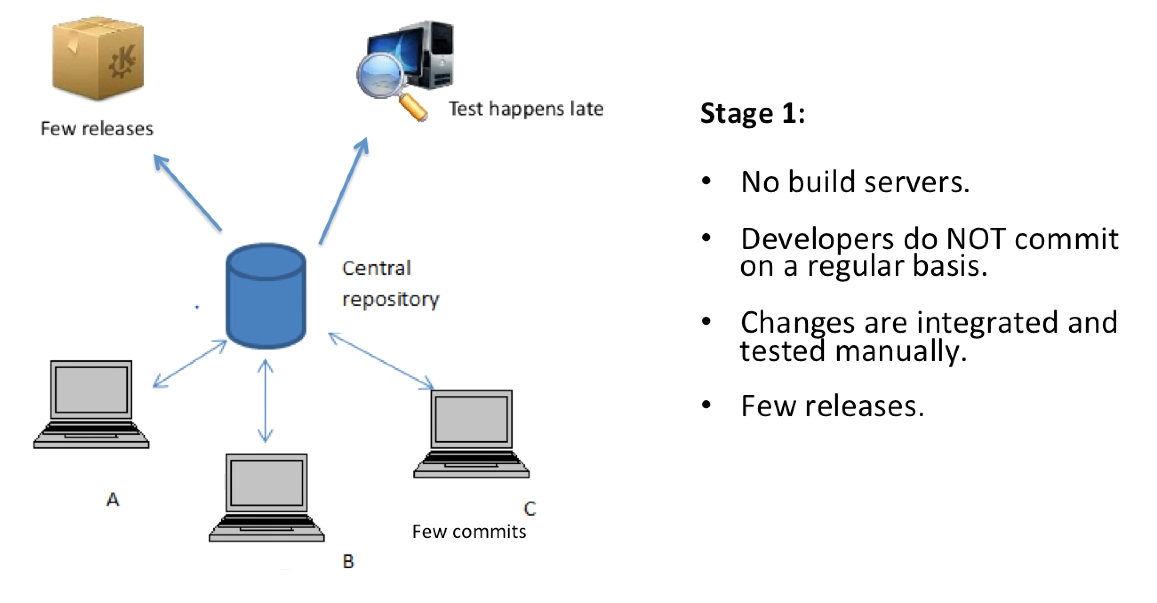
[2. Master and Slave Architecture 5](#_Toc12870832)

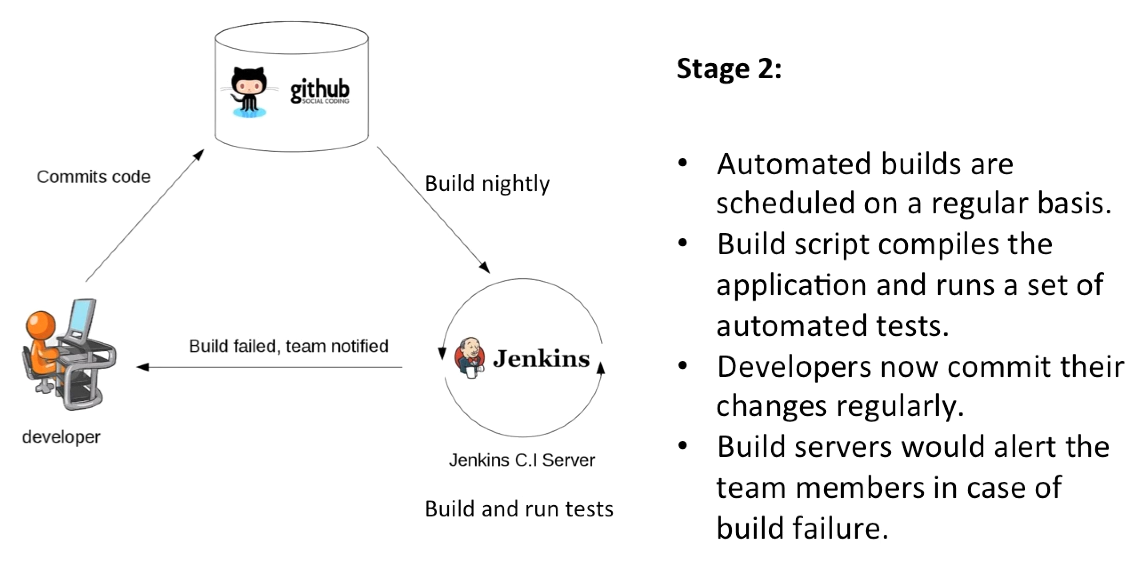
[3. Terminology 5](#_Toc12870833)

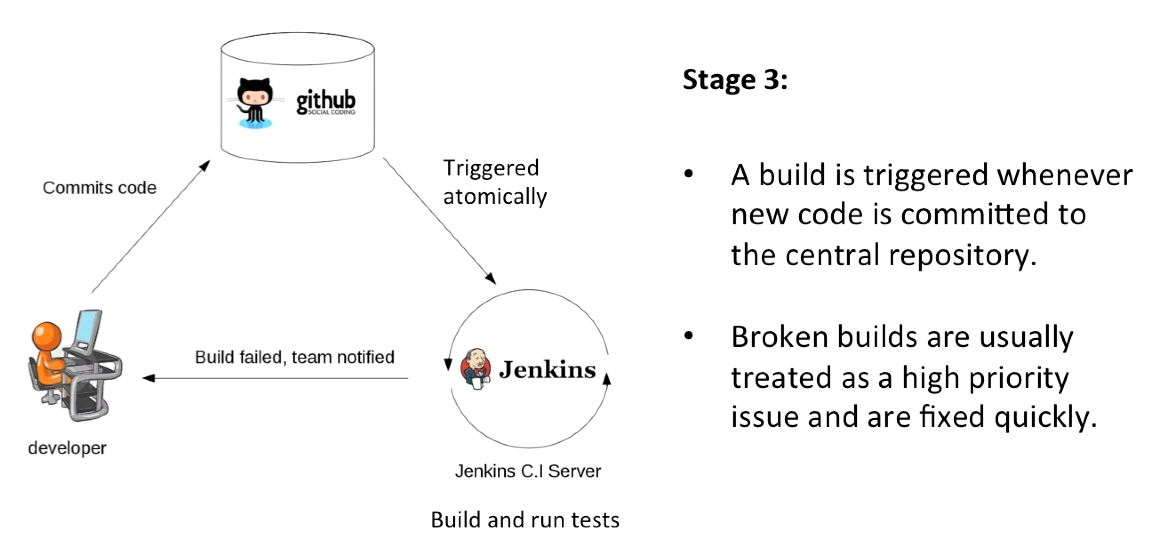
[3. Automating your Delivery Pipeline from GitHub to Amazon EC2 instance using Jenkins 7](#_Toc12870834)

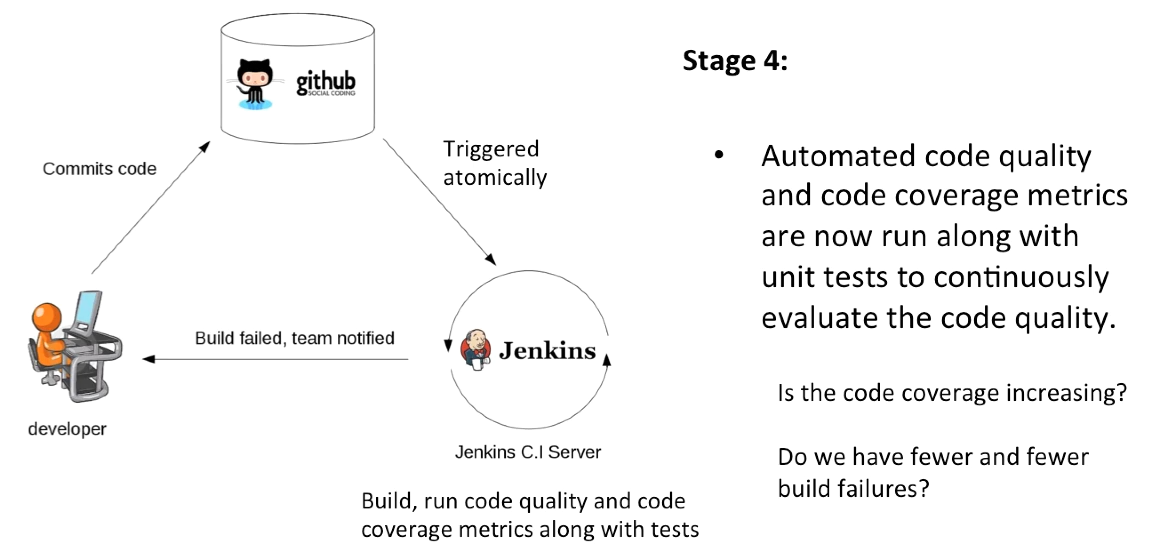
1. Concepts
   1. Continuous Integration

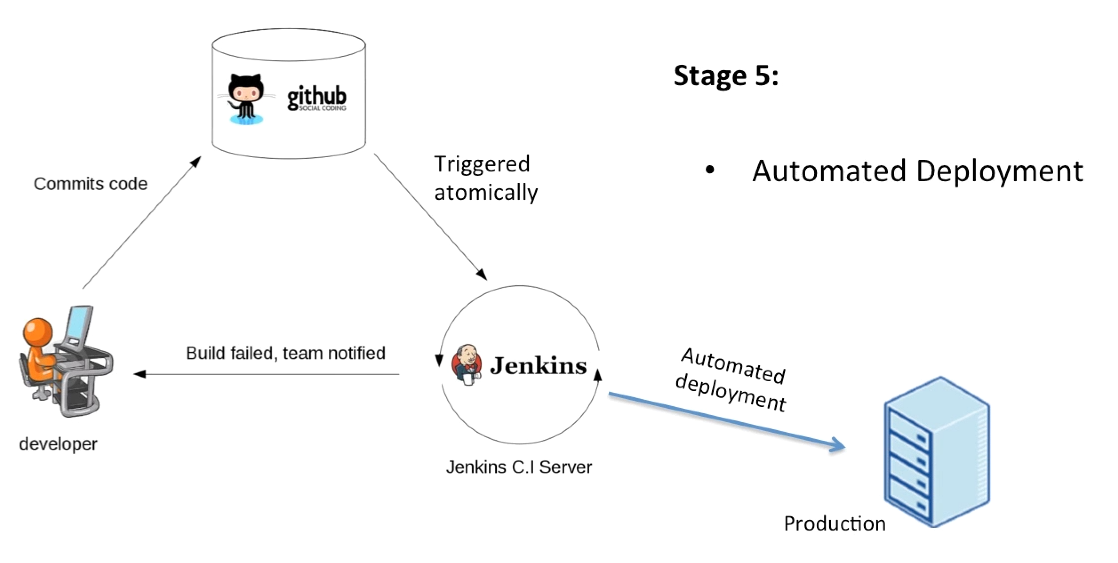












**Continuous Integration**

The practice of merging development work with the main branch constatntly.

**Continuous Delivery**

Continual delivery of code to an environment once the code is ready to ship. This could be staging or production. The idea is the product is delivered to a user base, which can be a QUs or customers for review and inspection.

**Continuous Deployment**

The deployment or release of code to production as soon as it is ready.

Non-hosted solutions: Jenkins

Hosted solutions: circleci

Others:

* Buddy
* TeamCity
* TravisCI
* Bamboo
* GitLab CI
* Codeship
* codefresh

1. Jenkins

* CI and build server
* Used to manually, periodically, or automatically build software development projects
* It is an open source CI tool written in Java
* Language support : Groovy, PHP, .NET, Ruby, C/C++, Android, Java, Python
* Easy to use, lots of plugins
  1. Installation

JAVA\_HOME env var points to the installation path for the JDK

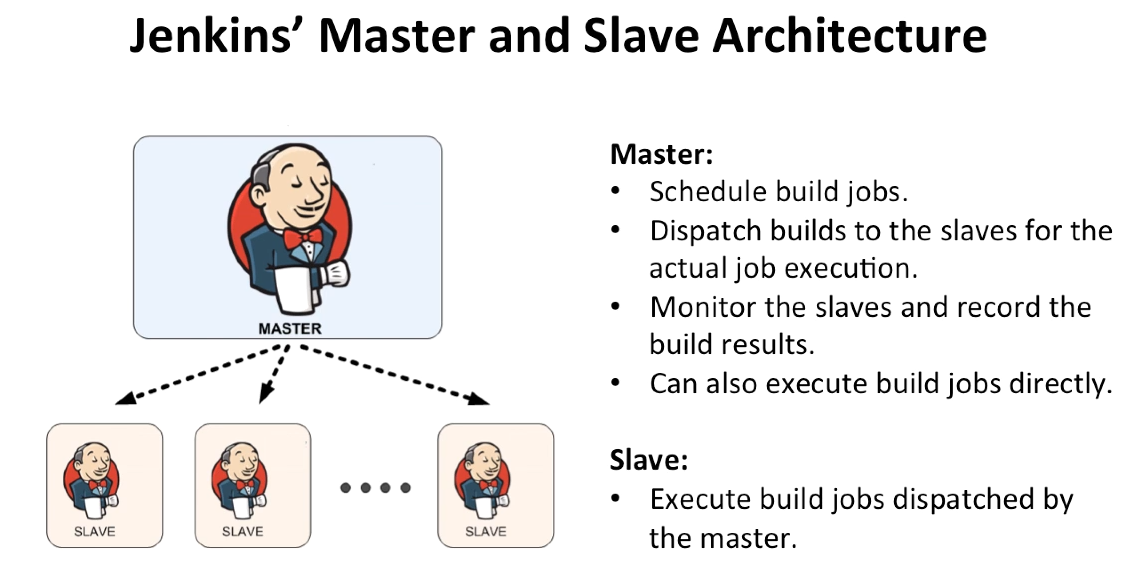
Address: <http://localhost:8080>

echo %JAVA\_HOME%

cd %JENKINS\_HOME%

jenkins [stop | start]

* 1. Master and Slave Architecture



Slave is a small java program that listens to the request

* 1. Terminology

**Job / Project**

Those 2 terms are used interchangeably. They all refer to runnable tasks that are controlled / monitored by Jenkins.

**Slave / Node**

Slaves are computers that are set u to build projects for a master. Jenkins runs a separate program called “slave agent” on slaves. When slaves are registered to a master, a master starts distributing loads to slaves. Node is used to refer to all machines that are part of Jenkins grid, slaves and master.

**Executor**

A separate stream of builds to be run on a node in parallel. A node can have 1 or more executors.

**Build**

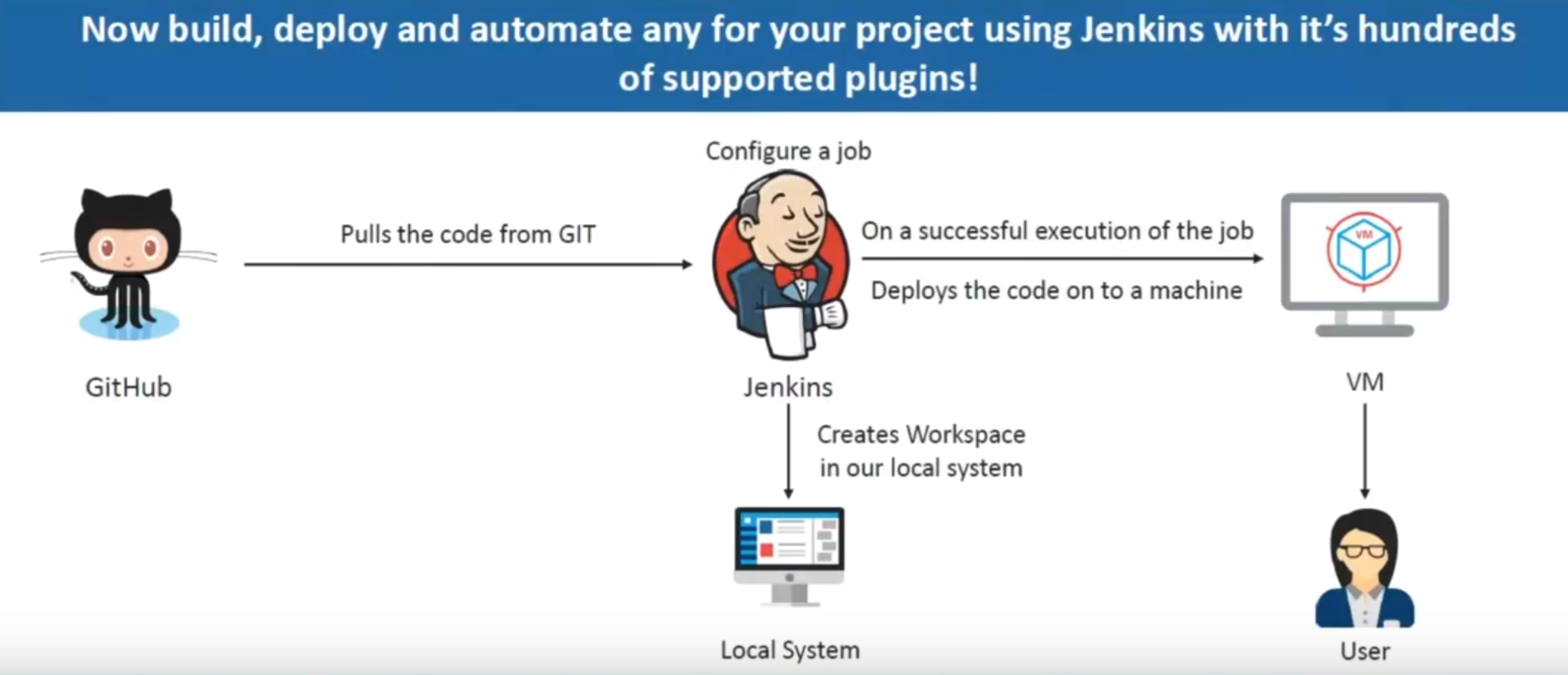
Is a result of one of the projects.

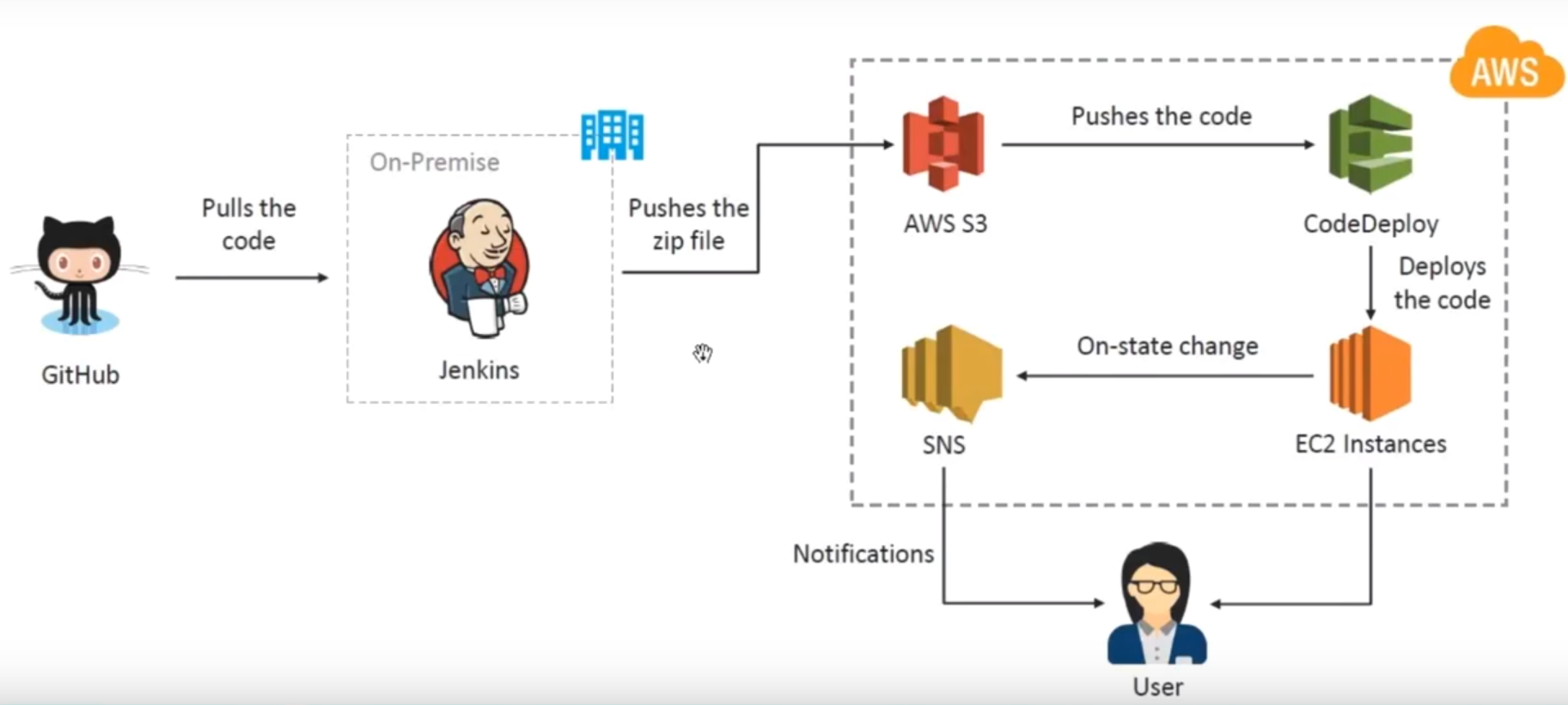
**Plugin**

A Plugin, like plugins on any other system, is a piece of software that extends the core functionality of the core Jenkins server.

1. Automating your Delivery Pipeline from GitHub to Amazon EC2 instance using Jenkins

Jenkins AWS plugin???

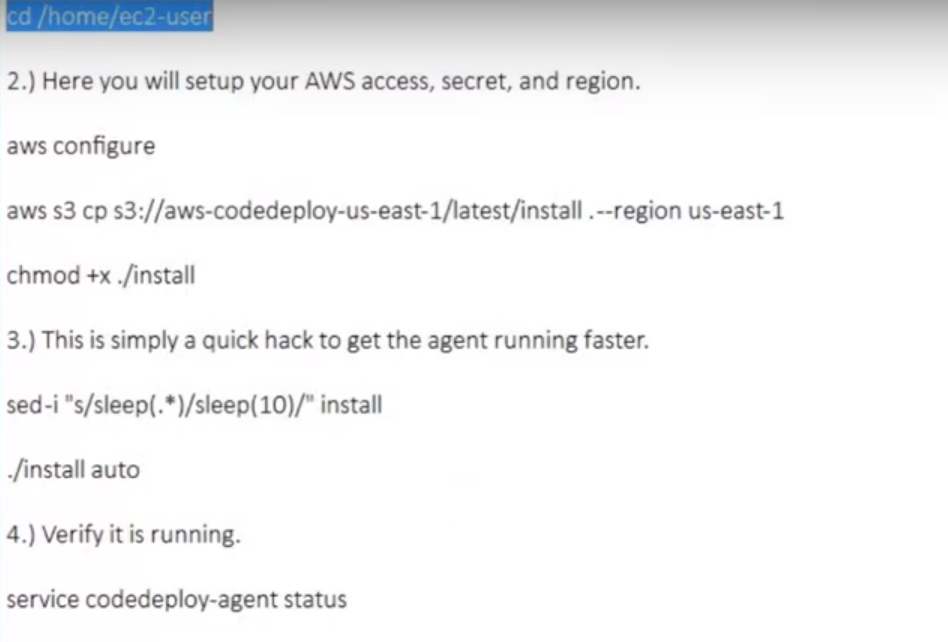




* Jenkins is on my local machine

Steps:

1. Ec2 needs to communicate with S3 (create role)
2. Install code deploy agent on Ec2 (code deploy agent???))



1. Code Deploy:
   * Custom deployment
   * Crate application (app neme, deployment group name)
   * Choose Auto Scaling group or Ec2 instances (in this we proceed w Ec2 instnaces)
   * Deployment configuration (CodeDeployDefault . OneAtTime – for this tutorial)
   * + Code Deploy needs a role (god knows how, ec2 is needed for sure)
2. Create Deployment
   * Goto Deplyoment groups -> Actions -> Deploy new revision
   * Configure your git repo (github, not gitlab w mfa )
   * Content options – override the content
3. Jenkins
   * Pust-build Action – where it is specified to deploy the application onto the AWS code deploy.
   * Region, bucket, access key, secret access key … and aws shits

More info: <https://www.youtube.com/watch?v=LFkGtg-ZTko>

1. AWS CodePipeline Integration With CodeDeploy and GitHub