CICD

LEARNING OBJECTIVES

- Differentiate between automated testing, test automation, continuous integration, delivery and deployment and DevOps principles
- Run different tests separately using test suites and build goals, plugins, profiles and properties
- Test locally and remotely using a virtual machine image
- Deploy remotely using to a virtual machine image
- Set up an automation server capable of building, testing and deploying depending on test results

CONTINUOUS PRINCIPLES

AUTOMATED TESTING

Automated testing is the act of automatically conducting execution of various test cases, based on some test scripts and by using testing frameworks

TEST AUTOMATION

Test automation refers to automating the process of managing and tracking tests by using automation tools

CONTINUOUS INTEGRATION

Continuous integration is the practice of routinely merging all developer work and continually integrating code changes together into a shared mainline

CONTINUOUS DELIVERY

Continuous delivery involves having an automated release process and the ability of easily releasing code changes at any time

CONTINUOUS DEPLOYMENT

Continuous deployment consist of having every code change automatically tested and deployed when tests are passed

DEVOPS

Set of practices that automates the processes between software development teams (Dev) and information technology operations teams (Ops), in order to build, test, and release software faster and more reliably

RELEASE PIPELINE

A build is generated based on code changes and automated tests are used to validate the build before releasing build

Software is continuously in development and always deployment ready, while testing is done as early, as often and as much as possible

CONTINUOUS STEPS

Locally

Developing / Building / Testing / Deploying

- Execute different types of tests (Unit / Integration)
- Build application
- Test using virtual machine

Remotely

Adding / Commiting / Pushing / Triggering / Building / Testing / Deploying

- Add, commit and push code to github
- Trigger Travis
- Execute different types of tests with test suites (Unit / Integration)
- Build application
- Deploy application to test server
- Deploy application to host server depending on test results

CONTINUOUS CHALLENGES

- Running different test suites
- Merging branches depending on test results
- Updating droplet image
- Keeping server active while redeploying
- Generating reports and mails
- Checking status of build process
- Managing SSH keys / passwords between local and virtual machine, github repository, test and deployment servers

VIRTUAL MACHINE

Emulation of a computer system / Sandboxed operating system

Virtual machine can be used to simulate production environment

VIRTUALBOX

Free open source hosted hypervisor supporting the creation, execution and management of virtual machines

Download and install VirtualBox

https://www.virtualbox.org/

VAGRANT

Open source product for building and maintaining portable virtual software development environments

Download and install Vagrant

https://www.vagrantup.com/downloads.html

Check Vagrant version

vagrant --version

Create Virtual Machine root folder with image files

Vagrantfile	Base configuration
Install.sh	Bash script installation

Start up Virtual Machine

vagrant up

Log in to Virtual Machine

vagrant ssh

VIRTUAL MACHINE IMAGE

Copy of the entire state of a computer system stored in a file
Virtual machine image requirements…

- Ubuntu 18.10
- OpenJDK 11.0.1
- Mysql 8.12
- Tomcat 9.0.16

Vagrantfile	->	Set up operating system / Set up synced folder / Set up ip and port
Install.sh	->	Install openjdk, mysql, tomcat / Set up root and sudo users

Use virtual machine image locally for test server

MAVEN

Build tool for managing Java based projects, their dependencies, their plugins and their build process
Build life cycle is made up of phases that can be controlled with goals, profiles and properties

Phases…
clean / validate / compile / test / package / verify / install / deploy

Plugins…

SureFire Plugin	Unit tests
FailSafe PlugIn	Integration tests

Current version of Surefire plugin has problems with Netbeans integration, execution of JUnit5 test suites and generation of JaCoCo reports

For now:	Do not add SureFire and FailSafe plugins to main build configuration
At some point:	Add SureFire and FailSafe plugins to main build configuration, when problems are fixed

ExecMaven Plugin	Execute bash script

Goals…

mvn test	Executes unit tests
mvn verify	Executes unit and integration tests

Naming convention
Default naming convention for JUnit unit tests classes are ”Test*”, “*Test” and “*TestCase” for unit tests
Default naming convention for JUnit integration tests classes are “IT*” and “*IT” for integration tests

Do not name tests and test suites with default prefixes or postfixes to disable execution of tests when building
Name test classes “Integration*”, “Unit*” or “I_T_E_S_T_*”, “U_T_E_S_T_*” or other conventions instead

Use regular expressions to exclude and include class name patterns in JUnit test suites and maven plugins

*^.*Unit.**	All files named Unit*
*^.*Integration.**	All files named Integration*
*^.**	All file names

Profiles…

mvn verify -P verifyLocal

Properties…

mvn test -D test=tests.unit.UT_Basic
mvn test -D test=tests.suites.Suite_UT

NETBEANS GOALS

By default Netbeans Maven goals are used to build, clean and test when working in a Maven project

Additionally more specific Maven goals can be set up and executed in Netbeans by right clicking the project and under "Run Maven" setting up the goals, profiles and properties before executing them

DEPLOYMENT

Tomcat7 plugin…
Deployment to Tomcat server can be done by using Tomcat7 plugin and with "mvn clean tomcat7:deploy" instead of manually using Tomcat manager

Bash script…
A more flexible approach would be to use ExecMaven plugin and execute a bash script on virtual machine…

DIGITAL OCEAN

Cloud infrastructure provider / Hosting of virtual machine images

Droplet <-> Custom image <-> Virtual machine

Use virtual machine image remotely for test and deployment server

UPLOADING CUSTOM IMAGES

Upload virtual machine image files as custom images to Digital Ocean
C:\Users\username\VirtualBox VMs\VirtualMachineFolder\VirtualMachineFile.vmdk

Import via url
Provide link to vmdk file uploaded online

Create droplet based on custom image
Select droplet size same size or larger size than custom image or scale down custom image size
Add public SSH key

TRAVIS

Travis CI is a hosted, distributed continuous integration service written in Ruby used to build and test software projects hosted at GitHub
Online server for building, testing and deploying projects from GitHub repositories, triggered by activity and based on a .travis.yml file in the GitHub repository

https://travis-ci.com/
https://travis-ci.org/ DEPRECATED

1. Sign up with GitHub account
2. Authorize GitHub account
3. Select GitHub repository
4. Create .travis.yml file in GitHub repository

.travis.yml

Language	Project language
jdk	Project jdk
Cache	Build tool caching

Job build phases…

1. Install apt addons — OPTIONAL
2. Install cache components — OPTIONAL
3. before_install
4. install
5. before_script
6. script
7. before_cache — OPTIONAL
8. after_success / after_failure
9. before_deploy — OPTIONAL
10. deploy — OPTIONAL
11. after_deploy — OPTIONAL
12. after_script

Linux commands…
Linux commands such as echo, bash, openssl, scp, ssh and others can be used during different job build life cycle phases

Maven commands…
Maven commands can be used during different job build life cycle phases
If any tests fail it can be handled in after_failure phase and if not tests fail it can be handled in after_success phase

RUBY

Ruby is a dynamic, interpreted, reflective, object-oriented, general-purpose programming language
Required to install the Travis command line client

https://www.ruby-lang.org/en/
https://rubyinstaller.org/

WINDOWS:

Download RubyInstaller
Install Ruby
Install MSYS2 base component

LINUX:

sudo apt install ruby ruby-dev
sudo gem install travis
snap install travis

TRAVIS COMMAND LINE CLIENT

Used for encrypting data / files and interacting with Travis online

https://github.com/travis-ci/travis.rb#command-line-client

Use Travis command line client in command prompt...

Install

Install Travis command line client
Windows:

gem install travis -v 1.8.9

Linux:

snap install travis

Check Travis installation

travis --version

Login

Login to Travis with Github username and password

Command prompt:

travis login --pro

Git bash:

winpty ruby `which travis` login

Environment variables…

Environment variables can be created on Travis under settings for a repository

Encrypt environment variable

travis encrypt --com SOMEVAR="secretvalue"

Encrypt environment variable and add it to .travis.yml file

travis encrypt --com SOMEVAR="secretvalue" --add

Encrypt file

travis encrypt-file ./sshkey_openssh --add

travis encrypt-file ./sshkey_openssh –r owner/repository

Make sure to add sshkey_openssh.enc to the git repository

Make sure not to add ./sshkey_openssh to the git repository

ALTERNATIVES

DOCKER

Open source computer program that performs operating-system-level virtualization

Used to run software packages called containers

Containers are isolated from each other and bundle their own application tools, libraries and configuration files

Designed to make it easier to create, deploy, and run applications by using containers

Containers allows packaging an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package

Docker is a bit like a virtual machine, but unlike a virtual machine, rather than creating a whole virtual operating system, Docker allows applications to use the same Linux kernel as the system that they're running on and only requires applications be shipped with things not already running on the host computer. This gives a significant performance boost and reduces the size of the application compared to Vagrant.

JENKINS

Leading open source automation server written in Java

Offers a simple way to set up a CI / CD environment for almost any combination of languages and source code repositories

Available as Java WAR archive and installer packages for the major operating systems, as Homebrew package, as Docker image, and as source code

Run the Jenkins WAR standalone or as a servlet in a Java application server such as Tomcat, in either case, it produces a web user interface and accepts calls to its REST API

RESOURCES

https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment
https://www.taniarascia.com/what-are-vagrant-and-virtualbox-and-how-do-i-use-them/
https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html
http://wiki.netbeans.org/MavenBestPractices
https://travis-ci.com/
https://travis-ci.org/getting_started
https://docs.travis-ci.com/
https://docs.travis-ci.com/user/encrypting-files/
https://github.com/travis-ci/travis.rb
https://github.com/travis-ci/travis.rb#command-line-client
https://www.ruby-lang.org/en/
https://rubyinstaller.org/
https://www.docker.com/
https://jenkins.io/