Github Repo: <https://github.com/bharaththippireddy/devopsandawsformicroservicedevelopers>

Source Build Test Deploy Monitor

Git Maven War/jar War/Jar Prometheus/Cloud Watch

Docker War/Jar Launch Container Launch Container on preprrod

(dockerize app) Docker Image Deploy apps on cluster (k8s/ aws)

K8s for container orchestration

(spawn pods)

Deploy on EBS

🡨------------------------------------------------------Jenkins ---------------------------------------------------🡪

Commit Compile Integration test Pre-Prod HealthChecks

Code Review Unit Test Load test Prod

Program War/Jar System test

Docker Image UI test

DevOps

1. Git

2. Docker

3. Maven

4. Kubernetes

5. Cloud Watch

6. Jenkins

7. EBS (Elastic Bean Stalk)

8. RDS (relational database service)

9. S3 (simple storage service)

10. IAM (security)

11. Auto Scale

12. EC2 (elastic compute cloud) provides virtual machine to run our application

13.AMI (Amazon machine image)

AMI combination of Software (Docker + Java + Python + MySQL)

Public DNS and Public IP allocated

Launch an EC2 instance/VM

AWS EC2 uses SSH for communication

After launching a Linux instance you can use SSH to communicate with the machine

Port 22

Encryption/Decryotion

Public Key -> Remote machine

Private Key -> Client Machine

Generate keys

ssh-keygen –t rsa

Connect to AMI

1 Go to folder where copied awskeys.pem

2 open putty

3 pushkarchauhan91@pushkarchauhan91s-MacBook-Air linux % ssh -i "awsmykeys.pem" ec2-user@ec2-13-233-197-187.ap-south-1.compute.amazonaws.com

The authenticity of host 'ec2-13-233-197-187.ap-south-1.compute.amazonaws.com (13.233.197.187)' can't be established.

ECDSA key fingerprint is SHA256:+MARWjZKL9447QBErPHrF8FIzhP8fy+Skesn+TpFI8s.

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

Warning: Permanently added 'ec2-13-233-197-187.ap-south-1.compute.amazonaws.com,13.233.197.187' (ECDSA) to the list of known hosts.

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

@ WARNING: UNPROTECTED PRIVATE KEY FILE! @

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Permissions 0777 for 'awsmykeys.pem' are too open.

It is required that your private key files are NOT accessible by others.

This private key will be ignored.

Load key "awsmykeys.pem": bad permissions

ec2-user@ec2-13-233-197-187.ap-south-1.compute.amazonaws.com: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).

pushkarchauhan91@pushkarchauhan91s-MacBook-Air linux % sudo -i

Password:

pushkarchauhan91s-MacBook-Air:~ root#

EC2 Tips

1 Terminate instance whenever not required

2

yum install httpd

service httpd start

service httpd stop

yum remove httpd

[root@ip-172-31-44-43 ~]# tail -5 /var/log/yum.log

yum history

yum search java

yum search httpd

Maven

Maven uses “Convention over Configuration”

1. Particular Maven project structure

src/main/java

src/main/resources

src/test/java

src/test/resources

Command to build:

Mvn install (compile, run and package, deploy on server)

1. Maven Provides archetypes/templates which create a standard template with readymade architecture

Project Management Tool

1. Building the project
2. Compile source code
3. Run the unit tests and integration tests
4. Package code into jar file
5. Bundle jar files into war file
6. Deploy war files to servers
7. Dependency Management Tool
8. Maven uses public Repositories
9. Lightweight
10. Plugin model(surefire, compiler, wsimport plugins)

Create a maven project from command line

mvn archetype:generate –DgroupId=com.bharth –DartifactId=hellomaven –DarchetypeArtifactId=maven-archetype-quickstart –DinteractiveMode=false

Maven Plugins and Goals

archetype:generate(archetype is plugin and generate is goal)

install:install(install is plugin and install is goal)

goal eg: clean test package

eg of plugin : compiler, surefire, wsimport

Maven Phases and Goals

Process-resources – compile – test – package

Package will run all phases before it

Resources:resources -> compiler:compiler -> surefire:test -> jar:jar

AWS

productService : 9090

couponService : 9091

Manual Deployment to AWS using EC2 using S3

S3 -> Simple Storage Service, similar to google drive

1. Create S3 Buckets
2. Provide public access to jar files, change security rules and port
3. Deploy apps to EC2 (elastic compute cloud)
4. Create an EC2 instance
5. Install MySQl, Java -> start the services
6. Deploy jar on AWS using EC2 using S3 and Test
7. Launch spring boot application on EC2 instance