**Basic DevOps Tools & Linux Commands:**

1. SCM (Source Code mangament)

Tools : Git, GitHub, SVN, Bitbucket etc…

1. Build (Compile & Package)

Tools : Ant, Maven, Gradle, Fastlane ….

1. CI/CD (Contineous integration & Conitneous Deployment & Delivery)

Tools: Jenkins, Bamboo, Circleci, Teamcity etc…

1. CodeQuality (Validate Code syntax & Rules)

Tools: Sonarqube

1. Artifactory (Storing Artifacts Backp)

Tools : jfrog, Nexus

1. Deployment (Deployment & Configuration management)

Tools: Ansible, Chef, Saltstack,Puppet,Appcenter etc…

1. Cloud

Providers: AWS, GCP, Azure etc….

1. Docker (Containerization tool)
2. Kubernetes (Orchestraton tool)
3. Terraform (Infrastructure management tool)
4. Linux Commands \* basic Networking commands
5. XML, JSON, YAML, SHELL etc….

**Basic Linux Commands:**

1. **pwd = present working directory (folder)**
2. **mkdir = make directory (create folder) (EX: mkdir demo)**
3. **touch = create the files (EX: touch filename)**
4. **cd = change directory (Ex: cd movies)**
5. **ls = list of files**
6. **ls -a = list of hidden files**
7. **ls -ll = long list of all files**
8. **cp source destination (EX: cp abcd.txt 123.txt)**
9. **mv source destination (By using this we can rename and move the files)**

**Ex: mv abcd.txt myfile.txt or. mv abcd.txt c/user/username/foldename**

1. **rm filename (we can delete the file)**
2. **rm -rf foldername (we can delete files & Folder forcefully)**
3. **ipconfig = Windows**
4. **ifconfig = linux & Mac**
5. **ipaddr = find the ipaddress**
6. **ping ipaddress (to check the server is up or down)**
7. **chmod (by using this we can give access permissions to the files & Folders)**
8. **grep**
9. **find**
10. **top**
11. **kill**
12. **ps -ef**
13. **wget. :** **by using this we can download packages from internet.**
14. **Curl : by using this we can download & Upload packages to internet.**
15. **how to edit files and add the content in linux:**

**create a file = touch filename**

**vi filename**

**I = insert**

**Write content**

**Press esc**

**:wq! = save the data & quit.**

**25: . (dot) = present place or directory or folder**

**Day2:**

SCM = Source code management

**Repositories : (**Storage )

Online repo or central repo or remote repo

Local repository

Pull or Clone = taking code from online repo to local repo (laptop)

Do the required changes in code and do the commit

Push to online repository

Tools : Git,Github, Bitbucket etc….

Gitbash (localy install). ----------------🡪 Github (online repo)

Steps:

Create Github account in online?

Create repository in github?

Install git bash in local (laptop)?

Clone the code from github repository to local?

Do the required changes and add commit?

Push the latest code chabges to github?

Create Github account in online?

Go to google 🡪 use the below link 🡪 <https://github.com/join?source=login>

🡪 Username 🡪 Emailid 🡪 Password 🡪 Verify the account 🡪 create account 🡪 select student 🡪 learn to code 🡪 interested in DevOps 🡪 create account 🡪 verify email address 🡪login into github account.

Create repository 🡪 repo name 🡪 description 🡪 public 🡪 initialize with readme file 🡪 create repository.

Download git bash for windows & install:

<https://github.com/git-for-windows/git/releases/download/v2.30.1.windows.1/Git-2.30.1-64-bit.exe>

1. install the git bash in your computer.
2. Go online repo copy the URL link.
3. go to desktop create folder.
4. go to folder give right click and open git-bash.
5. git config - -global user.email “emilid”
6. git config - -global user.name “name”
7. git clone repoURL link
8. cd reponame
9. ls -ll
10. modify the files
11. git status (red colour)
12. git add .
13. git status
14. git commit -m “message”
15. git push (if required give the username and password)

Note: after push go to your git hub account refresh the page and check the latest changes.

Realtime:

We are not push the code directly to main branch, we have to create the pull request by using any feature/bugfix/release branches, once PR is approved then we can merge the code form our branch to main branch.

Branching strategy:

Develop branch

Bugfix branch

Release branch

Feature branch

Git Commands:

git config - -global user.email “emailid”

git config - -global user.name “username”

git clone https://github.com/devopsreddyprasad/DevOps.git

ls

cd DevOps/

ls

vi README.md

touch developercode

vi developercode

git status

git add .

git status

git commit -m "added data in readme file and create the new devcode"

git log

git push

history

**Note: What is ssh and how we are using this in realtime for security.**

1. **To creaete the ssh-key we have to use the below command: ssh-keygen**
2. **Go to /user/username/.ssh**
3. **Cat id\_rsa.pub = publickey id\_rsa = Provate key**
4. **Always privatekey have to connect with publickey**
5. **When we are creating new ssh key then we need add latest ssh keys with servers to perform operations.**

**How to add sshkeys with github:**

1. **Go to gitbash 🡪 run this command (sssh-keygen) 🡪 don’t add any phsaprase give enter upto 3-4 times to create the ssh-key.**
2. **Go to ssh location copy the publickey and add into github.**
3. **Go to github 🡪select global settings🡪select ssh keys 🡪add sshkey 🡪paste the ssh-key 🡪save it.**
4. **Then go to your repository 🡪 code 🡪 clone the code by selecting ssh 🡪 coming to desktop 🡪 create new folder 🡪do git clone with ssh url link 🡪 accept ‘yes’ 🡪 now code is cloned by uising the ssh.**

**BUILD TOOLS :**

**Tools : ant, maven ,gradel, fastlene.**

**Maven lifecycle : validate, clean, compile, test, package, deploy, install.**

**What is plugins & maven plugind advantages?**

**Maven repositories :**

1. **Local repo(.M2): by using this maven will store all dependencies in local, once if we have dependencies then it will use form here, otherwise it will connect to central repository and download the dependencies and storing in .M2 repo.**
2. **Central repo: this is the repo is maintained by apache maven repositories.**
3. **Remote repo or (Artifactory): this repo is maintaining by our project team; there we will keep our project dependencies and securely.**

**Maven Files:**

1. **Setting.xml : by using this we can add plugins and configure (integrate) with other tools.**
2. **POM.xml : ( project object model ) by using this we can define our project details.**

**Ex: scm, G = groupid , A = artifact id, V= versions, Distributions, Dependncies, package details etc…..**

**Installation of maven and working with maven:**

**Step: maven have JDK dependency.**

How to Configure Maven & JDK:

1. Oracle account to download the JDK file

Link : <https://profile.oracle.com/myprofile/account/create-account.jspx>

Note : add all required details and create the oracle account.

JDK 🡪 JRE .

1. Downloaded & install JDK file.

Link: <https://www.oracle.com/in/java/technologies/javase/javase-jdk8-downloads.html#license-lightbox>

1. Maven zip Download & Unzip .

Link: <https://mirrors.estointernet.in/apache/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-bin.zip>

1. Copy Java & Maven path up to bin:

Java : C:\Program Files\Java\jdk1.8.0\_181\bin

Maven : D:\DevOps\_Softwares\apache-maven-3.6.3\bin

1. Configure Java & Maven in Environment variables:

Go to this pc 🡪 Rightclick 🡪 Properties 🡪 Advanced System Settings 🡪 Environment Variables 🡪

**Uservariables** : (here create new and add java & Maven paths before bin

EX: **JAVA\_HOME** = C:\Program Files\Java\jdk1.8.0\_181 ,

**MAVEN\_HOME** = D:\DevOps\_Softwares\apache-maven-3.6.3

* **System Variables** :

**JAVA\_HOME**= C:\Program Files\Java\jdk1.8.0\_181\bin ,

**MAVEN\_HOME=** D:\DevOps\_Softwares\apache-maven-3.6.3\bin.

1. Go to desktop and open cmd or gitbash.
2. Search for java & maven versions

CMD: java –version , mvn - -version.

**How to create POM.xml from Scratch:**

1. Install and check the java & maven versions.
2. Go to c drive 🡪 Users 🡪 Username 🡪 open window and minimize.
3. Create folder (maven\_operations)in desktop 🡪 go to folder 🡪give right click 🡪open gitbash 🡪run the below connamds 🡪CMD: “mvn archetype:generate” 🡪 chose version 🡪 example choose 3rd version 🡪 give the group id (google.com) 🡪 artifactid (com.google) 🡪 version (google) 🡪 packagetype = war 🡪 give “Y” for yes operation 🡪 build is success.
4. Go to maven desktop folder and go to pom.xml location 🡪 run the maven commands and observer the changes.

**Commands:**  
153 ls  
154 cd com.google  
155 ls  
156 mvn clean =. Sucess  
157 mvn compile = success  
158 mvn test = sucess  
159 mvn package = success  
160 mvn deploy = fail  
161 mvn clean. = here observe what it is cleaning  
162 mvn package = here observer all stages of commands opeartions

**CI/CD TOOLS:**

Diagram

Description automatically generated

CI = continuous integration

CD = continuous deployment or delivery

1. **Installation of Jenkins :**

**Install Java 🡪 go to google 🡪 search Jenkins war download 🡪**

**Link :** **<https://updates.jenkins-ci.org/download/war/> 🡪 Here choose version 🡪 Ex: 2.244 🡪 War file download link :** [**https://updates.jenkins-ci.org/download/war/2.244/jenkins.war**](https://updates.jenkins-ci.org/download/war/2.244/jenkins.war)

**OR**

1. **Use the below link and hit the windows**

[**https://www.jenkins.io/download/**](https://www.jenkins.io/download/)

**--> move this file into devops softwares folder 🡪 🡪 open git-bash in Jenkins war location 🡪 Run the below command : java –jar Jenkins.war . here accept firewall settings 🡪**

**Once Jenkins is up and running 🡪 go to google and search for localhost:8080 🡪 go to the below user location and copy the password (**C:\Users\Reddy prasad\.jenkins\secrets\initialAdminPassword**) 🡪 Continue 🡪 install sugguested plugins 🡪 UN =admin 🡪 Password = admin 🡪 Confirm-password = admin 🡪 full name = admin -🡪 Email:** [**sbrpdevops@gmail.com**](mailto:sbrpdevops@gmail.com) **🡪 Save & Continue 🡪 Save & Finish 🡪Start using Jenkins.**

Install plugins:

Create user account :

Configure tools:

Credentials:

Manage jenkins = 1. Configure systems

2.Global tool configurations

3. Manage Plugins

4.Manage credentials

5.Manage Users

6.Manage nodes

7.system properties etc…..

Job Types:

1. Freestyle job

2.Maven Job

3. Multi job

4. Pipeline job..

How to map jenkins portnumber: java -jar jenkins.war --httpPort=8081

How to install Plugins:

**New-Day:**

**How to install Plugins: By installing this we can get new features**

**Go to Jenkins 🡪 Manage Jenkins 🡪 manage plugins 🡪 choose available 🡪 search for plugin 🡪 select the plugins 🡪 install without restart or install after restart.**

**Ex: git , maven integration, gradle, sonar, artifactory,docker, job config history etc…**

Create Freestyle Job:

Go Jenkins 🡪 select New item 🡪 give job name 🡪 select freestyle job 🡪 ok.

Go to Job Configure:

Add description 🡪 add parameters 🡪 use git SCm to clone the code 🡪 add URL link 🡪 add credentials 🡪add branch names 🡪 if required enable the cron tabs 🡪 Add Execute Shell & save job.

Build the job.

**NewDay:**

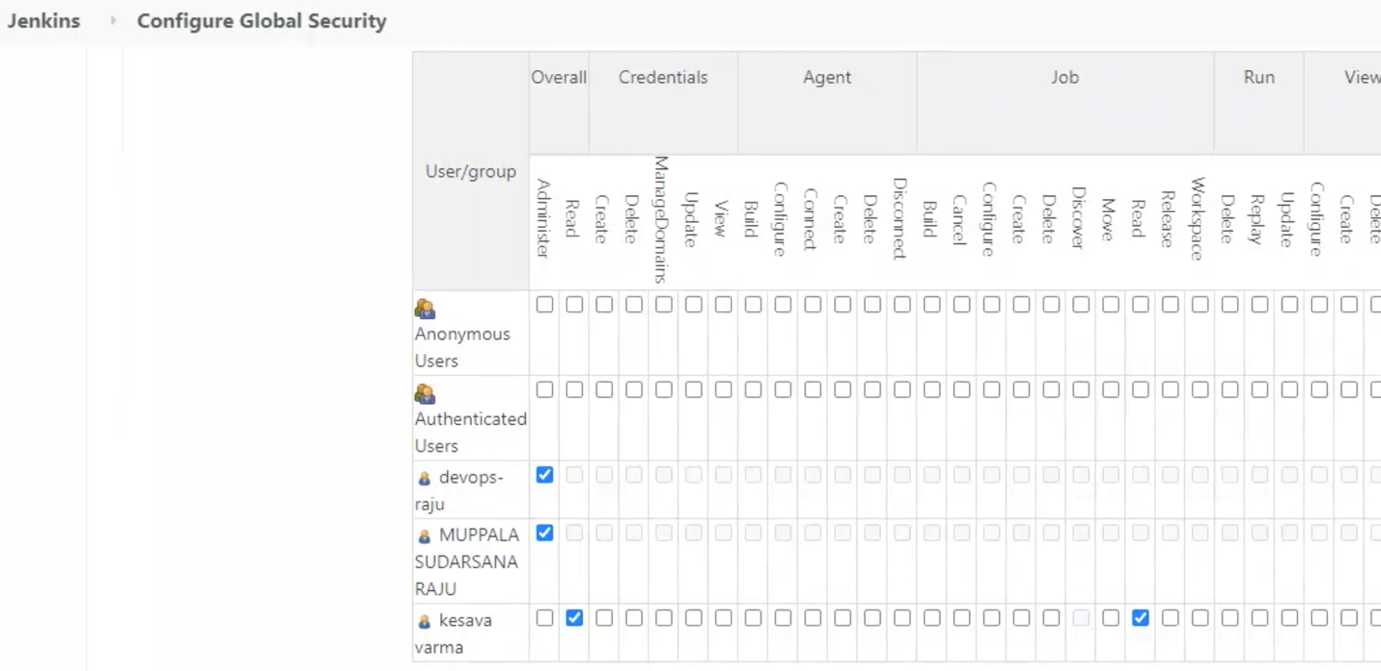
1. How to create Users and provide access ?

Jenkins ✍ manage Jenkins ✍ Manage Users ✍ create User ✍ Username, Email, Password, ConfirmPassword ✍ create User.

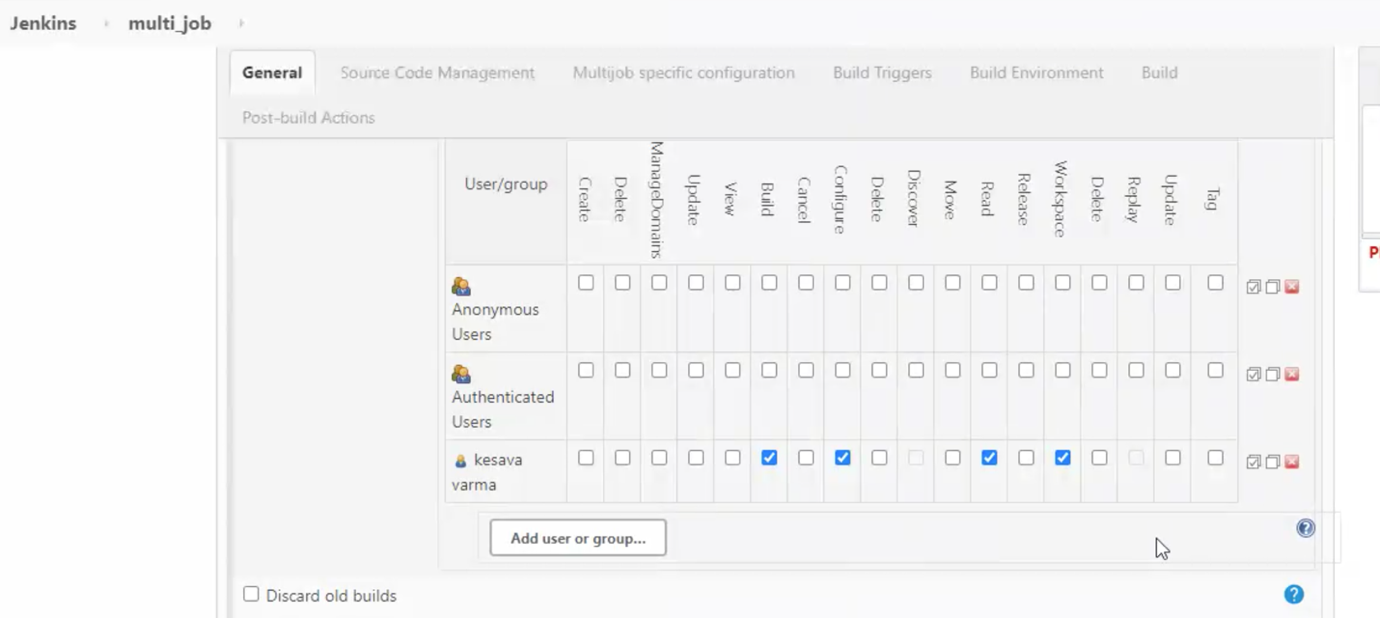
1. How to add Users in Global Security :

Jenkins ✍ manage Jenkins ✍Configure global tool security✍ Enable “Project-based authorization strategy” ✍ **Note: add your current admin user and provide administration access to admin-user, then add created user and provide admin access to them.**

✍ Create user and for this user we can provide limited access, for this see the below user “Kesava Varma” permissions. ✍ Save & Apply.



1. Go to Required Job✍ Select Configure ✍ Enable Project based security ✍ add user ✍ Give required permissions ✍ save & Apply.



**CI/CD= PIPELINES ?**

pipeline {

agent {

node {

label 'master'

}

}

stages {

stage('scm') {

steps {

git credentialsId: '16348fbc-5f20-4288-8a2f-79eab090c6bb', url: 'https://github.com/redprasa/google.git'

}

}

stage('build') {

steps {

sh '''

echo "this is the build stage"

mvn clean compile

'''

}

}

stage('test') {

steps {

sh '''

echo "this is testcases stage"

mvn test

'''

}

}

stage('sonar') {

steps {

sh '''

echo "this is Sonar codecoverage stage"

mvn sonar:sonar

'''

}

}

stage('package') {

steps {

sh '''

echo "this is Sonar package stage"

mvn package

'''

}

}

stage('JFROG') {

steps{

script{

if(env.BRANCH\_NAME == "stage") {

echo "PUSH Artifactory to JFROG"

sh ''' curl -u prasad:prasadprasad -T "/Users/redprasa/.jenkins/workspace/DevOps\_Class/batch3job/target/google.code-google.war" "http://localhost:8081/artifactory/devops123/" '''

}

}

}

}

}

}

**MASTER/SLAVE ?**

**JFROG (Artifactory) :**

1. Jfrog is enterprise one, we have 30 days free trail.
2. It will support for multiple repositories systems
   1. Local repo
   2. Central repo
   3. Remote Repo
   4. Distribution repo
   5. Tools wise repository types we have

EX: Maven, Gradle, Docker, Chef, Generic ………….

1. Go to jfrog website and do the registration login, here we can get the License-Key email, for our 30 days free trail.
2. Download the required package and install the jfrog.
3. Jfrog default port number is 8081.
4. Here we can provide the access to other users also.

Ex: Dev Team, DevOps team, QA team etc…….

1. Here we have to check the license key and validation timings also.

**JFROG (Artifactory):**

**Download Links: 1.** <https://bintray.com/jfrog/artifactory-pro/jfrog-artifactory-pro-zip/6.21.02>.

<https://bintray.com/jfrog/artifactory-pro/jfrog-artifactory-pro-zip/6.23.7>

How to get 30 days free- trail License-Key: <https://jfrog.com/artifactory/start-free/>

Installation:

1. Download artifactory 🡪 Unzip
2. Goto artifactory 🡪 upto bin 🡪 based on os run the file (if windows system give right click and run as administrator)
3. Go to browser 🡪 localhost:8081
4. Login UserName – admin, Password = password.
5. Click on next 🡪 Enter the license-Key 🡪 Next 🡪 Configure password 🡪 Skip 🡪Finish..

Graphical user interface, text, application, email

Description automatically generated

**What are the Types of Repositories We have and advantages?**

**Learn few details about repository Types.**

**How to create repo in Artifactory:**

**Goto Admin 🡪 Select Repo Type 🡪 +New 🡪 select package type 🡪 Generic 🡪 Repo Key 🡪 Public Description 🡪 Save & Finish.**

**How to upload files Manually & Download & Check the uploaded files details:**

**Artifacts 🡪 Select Repository 🡪 Deploy 🡪 Choose files 🡪 give the required directory path 🡪 Deploy.**

**Select artifacts** 🡪 **select repository** 🡪 **select actions** 🡪 **Native browser** 🡪 **here we can se the files uploaded details & We can download the files.**

**How to Add Users and provide permissions:**

**Select admin 🡪 security 🡪 Users 🡺 + New 🡪 UserName 🡪 Select Required Permissions 🡪 Password 🡪 create User.**

**SONARQUBE :**

**Topics: SONAR:**

1. **Projects**
2. **QualityProfiles**
3. **Rules**
4. **Qualitygate : q) blocker,major,critical, etc…**
5. **Administration , API,Users,Creating projects, install plugins etc…**
6. **Plugins : EX: jacocoa , Cobertura.**
7. **Integrate sonar with maven.**
8. **Sonar-project.propertiesfile**

**SonarQube:**

1. **Code quality Analysis 2. Quality profiles & Rules**
2. **Installation in windows, Linux, MacOS**
3. **Sonar-scanner**
4. **Login Admin/Admin**
5. **Projects**
6. **Quality gates & Quality profiles**
7. **Administration: Users Add/Delete, creating projects, add plugins, Api tokens, etc…**
8. **Sonar-Properties file.**

**Download Version 6.6:** [**https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-6.6.zip**](https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-6.6.zip)

**Extract the file 🡪** /Sonar-Qube/sonarqube-7.4/bin/Select windows (windows-64/32) 🡪 Run sonar.bat file with administration 🡪 go to Browser 🡪 localhost:9000 🡪 skip 🡪 login 🡪 UN = admin 🡪 PW= admin.

**How to Provide User access:**

**Administration 🡪 Security 🡪 Users 🡪 Create User 🡪 UN 🡪Full-Name 🡪 Email-id 🡪 Password 🡪 Create.**

**How to Modify User access: select user groups 🡪 enable permissions 🡪 Done.**

**How to reset user Password 🡪 User settings 🡪change password.**

**How Create API-Token & Reset your Current User Password:**

**Go to User account 🡪 Security 🡪 Change Password.**

**Go to User account 🡪 Security 🡪 Give Api Token Key Name 🡪 Generate Api Token.**

**How to Create Quality-Gates:**

**By using this we can pass the Conditions based on issues, if the qulitygate is not passed the condition then Jenkins build will be break or stop.**

**Select QualityGates 🡪 create 🡪 qualitygate- name 🡪 create 🡪 add conditions.**

**Graphical user interface, application, Teams

Description automatically generated**

**\*\*\* How to create projects & Assign the Qualitygates and qulity profies & Check the Code Quality Errors: \*\*\***

**How to create Project: Administration 🡪 Projects 🡪 Management 🡪 create new project 🡪 Project Name 🡪 Project Key 🡪 Create.**

**How to assign the Quality gates and profiles to project:**

**Select Project 🡪 Choose project administration 🡪 Select quality gate or Quality profile 🡪 Choose required quality gate or Profile 🡪 it will update automatically.**

**How to find project Issue & Types:**

**Choose required project 🡪 project wise issues 🡪 Seviarity 🡪 here we can see the Blocker, Critical, Major, Minor and Info issues with Developers name also.**

**AWS**

**Ec2-Launch: Linux :**

**Step1:**

**Go to services ✍ Ec2 ✍ launch instance ✍ select free-tier LINUX instance ✍ select instance type ✍ configure instance details ✍ add storage ✍ select security group ✍ add rule ✍ select all traffic ✍ here we can choose anywhere option ✍ add storage ✍ add tags ✍ download keypair ✍ launch ✍ view instance.**

**Note: after few minutes we can able to see our instance and give the required name.**

**Step2: How to connect instance.**

**By using git bash, putty and other third-party tools we can use to connect the Ec2 instance.**

**Select required instance ✍connect ✍select SSH client ✍ goto pem download location ✍ run the below command**  chmod 400 class instance.pem ✍ copy example and use the command to connect the instance ✍ now we are connected.



**Go to root user and install below plackages:**

sudo su

yum update -y

yum install java maven git -y

java -version

mvn --version

git -version

git --version

history

Topics:

1. Launch windows instance
2. Connect windows instance
3. Attach volume to windows instance
4. Create and attach volume.
5. Enable configuration in windows server to get the volume storage.

**How to Configure JDK in Linux:**

Step1: Install JDK:

1. yum --showduplicates list java-1.8.0-openjdk-devel

2. sudo yum install java-1.8.0-openjdk-devel.x86\_64

3. java -version

4. export JAVA\_HOME="/usr/lib/jvm/jdk-1.8.0-openjdk.x86\_64"

5. PATH=$JAVA\_HOME/bin:$PATH

6. source .bashrc

7. export PATH=$PATH:$JAVA\_HOME/bin

8. echo $JAVA\_HOME

9. echo $PATH

Note & Task: Learn advantages of Linux file system.

Linux File Syatem : bin boot dev etc home lib lib64 local media mnt opt proc root run sbin srv sys tmp usr var

**Ec2-Launch: WINDOWS:**

**Step1:**

**Go to services 🡪 Ec2 🡪 launch instance 🡪 select freetier Windows instance 🡪 select instance type 🡪 configure instance details 🡪 select security group 🡪 add rule 🡪 select all traffic 🡪 here we can choose anywhere or MYIP option 🡪 add storage 🡪 add tags 🡪 download keypair 🡪 launch 🡪 view instance.**

**Note: after few minutes we can able to see our instance and give the required name.**

**Step2: How to connect instance.**

**Select instance 🡪 Connect 🡪 Download RDP client 🡪 Get Password 🡪 Browse & Upload the Windows instance Keypair 🡪 Decrypt the password 🡪 Copy the Password 🡪 run or Install the downloaded RDP file 🡪 accept all firewalls 🡪add password 🡪 continue 🡪 now we are able to see the Windows server launch.**

**How to attach volumes to Ec2 Servers:**

1. **Attach volume to Ec2 -Linix Server:**

**Step1:**

**Go to ec2 🡪 select volume 🡪 create volume 🡪 volume name 🡪 storage capacity 🡪 select ec2 instance availability zone 🡪 if required snapshot give the name 🡪create volume.**

**Step2: select volume 🡪 actions 🡪 attach 🡪 select ec2 instance 🡪 attach.**

**Note: we can use one volume at a time only one single ec2 server.**

1. **How to attach volume to windows instance:**

**Step1: Go to ec2 🡪 select volume 🡪 create volume 🡪 volume name 🡪 storage capacity 🡪 select ec2 instance availability zone 🡪 if required snapshot give the name 🡪create volume.**

**Step2: select volume 🡪 actions 🡪 attach 🡪 select ec2 instance 🡪 attach.**

**Note: we can use one volume at a time only one single ec2 server.**

**Step3: do the need full admin changes in windows server.**

**Go to windows server 🡪 open control panel 🡪 select catrgory = small icons 🡪select administrative tools 🡪 computer management 🡪Diskmanagement 🡪select our volume give right click 🡪 select online 🡪 give again right click on our volume 🡪now select the initialize the disk 🡪 select volume unallocated 🡪give right click 🡪 new simple volume 🡪 next 🡪 next 🡪 next 🡪next 🡪 finish.**

1. **IAM (Identity and Access Management):**

Learn about USERS, ROLES, GROUPD and Policies advantages.

Learn about MFA and download Microsoft mfa authenticator app.

Assign MFA login to your AWS account.

**How to create Groups:**

Go to services 🡪 search for ‘IAM’ 🡪 Choose the Groups 🡪 create group 🡪 group name 🡪 next step 🡪 attach policies 🡪 select the policies 🡪 nextstep 🡪 create group.

Select group 🡪 actions 🡪 edit group name or add or remove users 🡪 select users 🡪 add or remove User.

**How to create USERS:**

Go to IAM 🡪 select Users 🡪 create user 🡪 Username 🡪 select the access type 1. Programmatic access 2. AWS management console access 🡪 Console password 🡪

Next permissions 🡪 select gropus to add or create new group and provide the access or 🡪 tag 🡪 create user 🡪 **Note**: share the below details to users : 1. copy the URL link and share with them 🡪 username 🡪 password 🡪 accesskey and secret key.

**How to create Roles:**

By creating this roles we will attach to AWS services,

Ex: once attached the role to ec2 then ec2 server will perform those operations based on role access.

Services 🡪 IAM 🡪 select roles 🡪 create role 🡪 choose service type 🡪 next step 🡪 permissions 🡪 select policies 🡪 tag 🡪 Review 🡪 Role-name 🡪 create role.

**How to attach role to Ec2:**

Select Ec2 instance 🡪 actions 🡪 instance settings 🡪 attach or replace IAM Role 🡪select I am role 🡪 attach or proceed.

**Policies:**

Policies are the permissions to provide in limits. We have different policies which is provided by the AWS.

If we need we can create our own policies also.

**S3 Buckets :**

Services ✍ s3 buckets ✍ create bucket ✍ bucketname ✍ selct region ✍ if required choose existing bucket settings (optional) ✍ block all public access ✍versioning enable ✍ create bucket.

Select s3 bucket:

Upload objects ✍ upload ✍ add file or folder ✍ choose file or folder ✍ upload.

What is s3 bucket lifecycle advantages?

What is s3 glacier and advantages ?

Try to lear s3 plugin configuration with Jenkins ?

Difference Between s3 bucket and AWS-CODE Artifact ?

Reference Link: <https://www.serverkaka.com/2018/08/upload-build-aws-s3-from-jenkins.html>

**Autoscaling:**

Step1:

Create Launch Configurations :

Go to Ec2 🡪 Select Autoscaling groups 🡪 Create Autoscaling Group 🡪 Name 🡪 choose the required launch configurations 🡪 **note:** if you don’t have exist templates create new template 🡪create Launch new template 🡪 Launch Template Name 🡪 Launch Template version description 🡪 Select AMI image 🡪 Select Instance Type 🡪 Create Keypair 🡪 Select Network Security group after launching the autoscaling groups or modify the existing security group and add inbound and outbound rules 🡪 storage 🡪 add volumes 🡪 tags 🡪 **Note:** if required use the advanced variables 🡪 create Launch Configuration 🡪 View Launch Template.

Create Autoscaling group:

Go back to AWS Autoscaling groups 🡪 create Autoscaling group 🡪 Name 🡪 select our launch configuration template 🡪 Next 🡪 if required select instance purchase options 🡪 network 🡪 required subnets selection 🡪 Next 🡪 Next 🡪 add the required instance details desired capacity = 3 , minimum capacity = 3 , Maximum capacity =3 🡪 next 🡪 Add Notifications 🡪 SNS topic 🡪 create new topic 🡪 add name & Email id 🡪 next 🡪 tag 🡪 next 🡪 verify all details and edit if required (or) create autoscaling groups.

Go to autoscale group check the details 🡪 go to EC2 Dashboard 🡪 check the servers are launching or not.

**####################################################################**

**Load Balancer:**

**Step1: create the required servers (3 +-..)**

**Step2: connect to the servers and check the security groups.**

**Step3:**

1. yum install java httpd -y
2. use this commands to start the httpd service:
3. **sudo service httpd start**
4. **sudo service httpd status**

2 vi /var/www/html/index.html

Add the required data in the file & save.

3 service httpd status

4 service httpd start

1. history

# add the webpage content data in index.html

###EX: go to browser, select ctrl+u , copy the content and edit and replace (google = your reqired name🡪 add this content to insex.html and start the service 🡪 copy instance ip address 🡪 search in google browaer 🡪 here we can see our data. )

Step4: go to instances copy public DNS name and go to browser check the webpage data.

**Step5:** go to loadbalancer 🡪 select classic load balancer 🡪 create load balancer 🡪 loadbalancer name 🡪Assign security groups 🡪 select security groups 🡪 Configure security settings 🡪 select check the health 🡪 add instances 🡪 add the tags 🡪 Review and Create 🡪 create 🡪 select load balancer 🡪 after 3 mns select load balncer 🡪 check the instances 🡪 instances must be in available or active state 🡪 copy the loadbalncer DNS name 🡪go to browser paste the link 🡪 and do refresh to get the all server web data.

**VPC : Virtual Provate Cloud**

What is IGW, DNS, ROUTETABLE, NACL , SUBNETS, CIDR ?

How to create VPC and launch Ec2 server in VPC:

Select services 🡪 VPC 🡪 Create VPC 🡪 VPC Name 🡪 Define CIDR BlockRange 🡪 Ex CIDR = 12.0.0.0/16 🡪 create VPC.

Create Subnets :

Based on requirement we can select and define the subnets.

Create subnet 🡪 choose VPC 🡪 subnet name 🡪 CIDR range of subnet 🡪 create subnet.

**RouteTable :**

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Select route table 🡪 routetable name 🡪 select vpc 🡪 create route table.

**IGW:**

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Select IGW 🡪 create IGW 🡪 Name 🡪 creae IGW.

Select the IGW 🡪 actions 🡪 attach to Vpc 🡪 Select VPC 🡪 attach InternetGateway.

**Launch Ec2 in VPC:**

Select Ec2 🡪 launch instance 🡪 select instance Which Req uired 🡪 select instance Type 🡪 Configure instance details 🡪 tag 🡪 storage 🡪 security groups 🡪 Keypair 🡪 launch 🡪 View instance .

Select and find the Private IP address range , If it is under our VPC CIDR range then this VPC is working fine.

Note : if you are not able to see the Public DNS host details, do the below changes.

Go to VPC 🡪 select VPC 🡪 actions 🡪 Edit DNS host Name 🡪 Enable 🡪 save or apply the changes.

**AWS CODE ARTIFACT:**

By using this we can store the our packages.

It will support for multiple tools Ex: MVN, Python, NPM etc…

**How to create Repo & integrate with tools:**

**Create Repo:**

Services 🡪 codeartifact 🡪 create repo 🡪 Repo name 🡪 Description 🡪 select tolls type to integrate codeartifact 🡪 Next 🡪 select domain 🡪 select required AWS account 🡪 select domain 🡪 Next 🡪 Review and Create Repo (create repository) 🡪 and check the repo is created or not.

**How to configure with tools :**

Select repository 🡪 click on the repository 🡪 check the Details (Ex: domain name, repo ARN , upstream repos, tags and repo policies) 🡪 click on Ciew Connection instructions 🡪 select package manager client tools 🡪 see the instructions to connect with pom.xml.

**###########################################**

**DOCKER INSTALLATION:**

1. Create EC2 Instance & connect to instance.
2. sudo yum install docker -y
3. service docker status

3 service docker start

4 service docker status

5 docker info

Sudo su

6 chmod 777 /var/run/docker.sock

7 docker images

8 docker ps

11 docker pull centos

12 docker images

13 docker pull ubuntu

14 docker images

15 docker pull sonarqube

16 docker images

17 docker ps

18 docker ps -a

19 docker run --name reddy\_centos -it centos

20 docker ps

21 docker exec 47b4b72362b0 /bin/bash

22 docker exec -it 47b4b72362b0 /bin/bash

23 docker ps

24 docker -it 47b4b72362b0

25 docker attach docker -it 47b4b72362b0

26 docker attach 47b4b72362b0

27 docker ps

28 docker ps -a

29 docker start 47b4b72362b0

30 docker ps

31 docker stop 47b4b72362b0

32 docker ps

33 docker start 47b4b72362b0

34 docker pause 47b4b72362b0

35 docker ps

36 docker unpause 47b4b72362b0

37 docker ps

38 docker stop 47b4b72362b0

39 docker rm 47b4b72362b0

40 docker ps

41 docker ps -a

42 docker run -itd --name reddy\_centos centos

43 docker ps

44 docker exec -it 499899c9a654 /bin/bash

45 docker ps

46 docker tag centos reddy\_centos:1.0

47 docker images

49 docker save centos > testcentos.tgz

51 docker export 499899c9a654 > myoperations.tgz

56 docker images --help

57 docker images

58 docker rmi 4dd97cefde62

68 docker logs 499899c9a654

69 docker attach 499899c9a654

70 docker logs 499899c9a654

71 docker tag reddy\_centos redprasa/centos:1.0

72 docker tag centos redprasa/centos:1.0

76 docker inspect 499899c9a654

77 history

**Docker Latest Commands:**

1. To share the volumes or data between host server to docker container

**CMD:** docker run -itd -v /Users/redprasa/Reddy-Data/appceter/fastlane:/home/ centos

**Note:** /Users/redprasa/Reddy-Data/appceter/fastlane:/ = Hostserverpath

/home/ = this is container path

Centos = images name

1. To check the docker logs & Container logs

**Cmd:** docker logs containerid

**Ex:** docker logs 13e822d6a662

1. To save the container changes into image format please follow the below command

**CMD:** docker commit 13e822d6a662 mycentos:1.0

**Note:** 13e822d6a662 – container id

mycentos:1.0 : our own required image name & version = 1.0

1. How to Rename docker Container

**CMD:** docker rename existing-container-name newname

**Ex:** docker rename reddy-centos reddy-own-centos

**Note:** reddy-centos = oldname

Reddy-own-centos = newname

1. How to search star-rated images in docker

**CMD:** docker search -s 100 Jenkins

Note: -s = stars

100 = stars count

Jenkins = image-name

1. How to login with username in container

**Cmd:** docker exec -it -u username container-id /bin/bash

1. How to assign a hostname to container

Cmd: docker run -it --name 13e822d6a662 -h Reddy-Host centos

Note: 13e822d6a662 = container-id

Reddy-Host = Required Hostname

Centos = image-name

1. To check the Docker History Details

**Cmd:** docker history image-name

1. Docker-TAG:

CMD: docker tag SOURCE\_IMAGE[:TAG] TARGET\_IMAGE[:TAG]

EX: docker tag centos reddy-centos-tag

docker tag centos:7 reddy-centos-tag:7

1. Save & Load:

by using this we can save our images and upload it into other server or we can download in other server.

CMD: docker save imagename > requiredimagename.tgz

(or)

Docker save imagename > required imagename.tar

Ex: docker save centos > reddy-centos.tgz

Load: by using this we can load the images whatever we are saved images.

CMD: docker load < imagename.tgz

Or

docker load < imagename.tar

EX: docker load < reddy-centos.tgz

1. Import & Export:

Export: by using this we can move container into imageid.

CMD: docker export containerid > required-imagename.tgz

Or

docker export containerid > required-imagename.tar

EX: docker export c00876910814 > reddy.export.tgz

Import: by using this whatever images we are exported those images we can import.

CMD: docker import exported-imagename.tgz

Or

docker import exported-imagename.tar

EX: docker import reddy-export.tgz

26. DIFF:

by using this we can find out the whatever changes we are done in container.

means inside the container changes we can see.

CMD: docker diff container-id

EX: docker diff c00876910814

NOTE: Here C=change

A=adding

D= deleting

27. DOCKER DATA SHARING:

**Data Sharing B/W HOST-OS to Container-OS:** here we can share the data between Host-OS to Container OS.

CMD: docker run –itd –v hostpath:containerpath image-name

Example:

**Data sharing B/W Container to Container:** here we can share the data-sharing (VOLUMES) between container to container.

CMD: docker run –itd –volumes-from container-id imagename

28. DOCKER LINK:

By using this we can provide the link between two containers

**How to run Jenkins in Docker:**

\*Docker pull Jenkins (it will pulling from DockerHub )

\* Docker Images (we can see our Jenkins image)

\* Docker images –q (to get the container id only)

\* Docker run –p 8080:8080 –p 50000:50000 jenkins (-p=port , 50000 is docker port, 8080 is our required port number)

\* Docker run –p 8080:8080 –p 50000:50000 –v /var/Jenkins\_home Jenkins

Note: this command is using for to change the port number as per our requirements, here we are changing Jenkins port number from 50000 to 8080 port number. Mainly “–v” is using to set the home path of Jenkins and “/var/Jenkins\_home” is path and Jenkins is imagename.

\*Docker run –name required-name –p 8080:8080 –p 50000:50000 –v /User/reddy/Desktop/Jenkins\_Home:/var/Jenkins\_home Jenkins

Note : here we are creating the Jenkins in our Desktop along with we are getting the path from previous command path in this document along with we are giving our required name for the container.

\*\*\*\* \*\*\*\* By using physical space we can share our Jenkins jobs and details and data, and we can use the same credentials and we can change the port numbers also as per our requirement\*\*\*\*\*\*

**DOCKER FILE:**

\*FROM \*RUN \*CMD

\* Docker file is a textfile with instructions to build Docker image

\*Automation of Docker Image Creation

\*\* Create a Docker file name should be Docker-file

\*Goto – Desktop – Create Folder – Create a file Name Dockerfile

- Edit and Write the Instructions.

Basic-DockerFile-Commands:

**RUN**: by using Run we can install or perform image related operation.

**CMD**: when we need to perform operations on running container we can use the CMD.

**Copy**: by using this we can copy our files or directories from local to image. Here it is going to work like source and destination. If you’re copying in local files to your Docker image, always use COPY because it’s more explicit.

**Add**: use case for ADD is when you want to extract a local tar file into a specific directory in your Docker image.

**Entrypoint:**  **ENTRYPOINT** instruction allows you to configure a container that will **run** as an executable.

# getting Base image

FROM image-name

MAINTAINER Reddy Prasad <[rprasad21@sapient.com](mailto:rprasad21@sapient.com)>

RUN Yum update (or) RUN apt-get update

CMD [“this is my first image”]

***REALTIME-SENARIO:***

# selecting the base image of centos

FROM centos

# This docker file is maintained by Reddy Prasad

MAINTAINER reddyprasad

# Creating a directory

RUN mkdir -p /opt/jenkins/

# Create the user jenkins

RUN useradd jenkins

# Installing the jdk

RUN yum install java-1.8.0-openjdk-devel -y

# copying the tomcat to host machine

COPY tomcat\_7.0.72 /opt/jenkins/

# change the user of the tomcat

RUN chown -R jenkins.jenkins /opt

# Copy the jenkins war file

COPY jenkins.war /opt/jenkins/webapps

# starting the tomcat instance with user jenkins

USER jenkins

CMD /opt/jenkins/bin/startup.sh && tail -f /dev/null

EXPOSE 8080

**KUBERNETES:**

1. installation purpose use the below link
2. <https://www.radishlogic.com/kubernetes/running-minikube-in-aws-ec2-ubuntu/>

(OR)

USE THE SHARED EKS DOCUMENT TO CREATE KUBERNETES CLUSTER IN AWS.

Kubernetes Basic Commands:

Note: To crate the deployment purpose please use the below commands and before this create the required YAML files (EX: deployment.yaml, Service.yaml and pod.yaml etc…..)

# to see the all details like deployments, services etc,….

640 kubectl get all

# to see the list of pods

641 kubectl get pods

# to see the list of services

642 kubectl get svc

#to see the list of deployments

644 kubectl get deployments

# to see the list of namespaces

645 kubectl get namespaces

# create the deployment yaml files

649 vi deployments-definition.yml

######\*\*\*\*\*\*\*\*\*###############

Yaml-Code : apiVersion: apps/v1

kind: Deployment

metadata:

name: testapp-deployment

labels:

app: mywebsite

tier: testapp-deployment

spec:

replicas: 5

template:

metadata:

name: myapp-pod

labels:

app: myapp

spec:

containers:

- name: nginx

image: nginx

selector:

matchLabels:

app: myapp

#########\*\*\*\*\*\*\*\*\*############

# create the service.yml file to access deployment applications

651 vi service-definition.yml

##########\*\*\*\*\*\*\*\*\*\*##########

Yaml-Code: apiVersion: v1

kind: Service

metadata:

name: devops3-deployment

labels:

app: myapp

spec:

type: NodePort

ports:

- port: 80

targetPort: 80

nodePort: 30004

selector:

app: myapp

##########\*\*\*\*\*\*\*\*\*########

# to create the deployments use the below command

654 kubectl apply -f deployments-definition.yml

Note: deployments-definition.yml =File name of yaml

# to check the deployment is created or not

655 kubectl get deployments

# to get the pod & Service details

656 kubectl get pod

657 kubectl get svc

# to access the application use the below link to get the URL link

658 minikube service testapp-deployment --url

######## goto Browser and Paste the URL link ########