DevOps

1. How to Jenkins in linux
2. Create First Free Style Jenkins Job:
3. Remote build trigger:
4. Parameterized job:
5. Webhook and Poll SCM:
6. Jenkins Maven Integration:
7. Creating users :
8. Role Base Access Control:

9. Jenkins configure system:

10. Create First Jenkins pipeline Job:

11. Building Maven java Project With Jenkins Pipeline:

12. Running Jenkins on Docker:

13. Presistant Jenkins Data Volume:

14. Jenkins Files

What is Jenkins?

Jenkins is a open source automation server.

It helps a to automation software development activities related to building and testing and deploying and facilitating continuous delivery and continuous integration

1000+ plugins available that helps for continuous integration

Continuous integration is most important tool part of DevOps and Jenkins is most important tool continuous integration

What is DevOps?

Development + operations

DevOps is the combination word is development and operations

DevOps is a culture, it promote to collaboration between Development and operation team to make software production and development faster automated, repeated a way, improve time to market.

What is Continuous Integration?

Continuous Integration (CI) is practice of automating the integration the code changes from multiple contributors into a single software project and allowing the developer to frequently merge the code changes from central repository which builds and test then run.

DevOps Life cycle:

Continuous development:

Plan

Code

Continuous Testing:

Build

Test

Continuous integration

Continuous Deployment

Release

Deploy

Continuous Monitoring:

Operate

Monitor

4.Define the process of Jenkins?

Commit the changes

Detect the sources code changes

The build either passes or fails

Generate the feedback

5.What are the benefits of using Jenkins?

Cache the build failure

Automatic build report notification

Achieves the Continuous integration

Automation the maven project

Easy to tracking the bugs

6.What is the relation between Jenkins and Hudson?

Hudson is the earlier name and version of current Jenkins after some issues the project changes the name from Hudson to Jenkins.

7.What is a Jenkins pipeline?

A Jenkins pipeline is a suite of plugins which supports implementing and integrating

Continuous delivery pipelines into Jenkins.

A continuous delivery pipelines is an automated expression of your process for getting software from version control right through to your users and customers

Jenkins pipeline provides an extensible set of tools for modelling simple-to-complex delivery pipelines”as code”.

The definition of a jenkins pipelines is typically written into a text file is called a jenkinsfile which is turn is checked into a project sources control respoitory.

Types of Jenkins pipelines:

There are two types of Jenkins pipeline code

1.Declarative pipeline

2.Scripted pipeline

Pipeline Concept:

. pipeline

. Agent/Node

.Stage

.Step

How to Jenkins in linux:

Go to putty using some commands:

sudo wget -O /etc/yum.repos.d/jenkins.repo \

<https://pkg.jenkins.io/redhat-stable/jenkins.repo>

amazon -linux -extras

amazon-linux-extras install   epel

amazon-linux-extras install java-openjdk11

yum install Jenkins

service jenkins status

service jenkins start

chkconfig jenkins on

cat

Jenkins Architecture Jenkins master slave setup:

Jenkins Distributed Architecture:

Jenkins master: 1.jenkins slave1

2.Jenkins slave 2

Jenkins master:

Scheduling jobs

Communicating and give instractions to slaves

Mointor the slaves

Can build the jobs as well

Jenkins slave:

Execute jobs

Takes the command from master

Finally Go to Jenkins dashboard:

Go to manage Jenkins

The manage nodes and clouds click on that

Build node or master node

On click on new node

We give the node name ,give the permanents Agent and click on ok

Next the description , remote root directory( c:/temp/)

Labels to gives the windows

Usages : only build with jobs label expressions matching this node

Launch method: Launch agent by connecting it to the controller

And save continuous, any errors click on go to security

Go to this settings:

Agents:

Click on Random and save it

Go to manage nodes and clouds

Again any errors got to ,agent not connetecd,click user name go to next page and uses some commands

:Jenkins master slave setup windows | Jenkins slave as windows service architecture.

(thetips4you channel go to)

Now lets move on the

Create First Free Style Jenkins Job:

Then go to Jenkins home pages, to create new job on the click on new item , and give the name to click on freestyle project and go to click on ok to create a job.

Next page

General :

Description on job: this is sample job

Source code management : ----

Build triggers:-----

Build environment:-----

Build : add builds steps: to select the execute shell : command : echo “This is my first project”

And to click on save

And now go back to back to dashboard and we can see the project are available here.

We can click on inside the job the go to so more option: status, changes, workspace, build now

,configure, delete project , rename . and we have build history.

Click on build now, we can the build history we show build 1…… and click on the build 1 or show the option we can click console ,we can see the output.

We go to putty :

Cd /var/lib/Jenkins/

Ls

Cd workspace

Ls

Cd samplefirstjob/

Ls

Cd ..

Clear

Cd ~

Scheduling job in Jenkins:

Cron Expression:

<minute> <hour> <day-of-month> <month> <day-of-week>

0-59 0-23 1-31 1-12 0-6

\*\*\*\*\* <<command to execute>

How to check the this command then got to project ,click on configure, got to build trigger got to build periodically

This is command :

\*/2\*\*\*\* click on save

Then go to check the build history check the every two min build the code.

After u see the result . then remove the command.

Remote build trigger:

Then the go to project ,go to configure then click on the build trigger, click on trigger builds remotely we can give the authentication token (sampleproject)

Then following the url to trigger build copy the url “/jobs to token name on wards copy “ the open new google tap and enter the url and change the tokenname you already the token name enter this and enter Jenkins port id . then you login and check it.

Parameterized job:

If have go back to project and then go to configure ,go to general option of the This project is parameterized click the add parameter then more option Boolean parameter like ……

Click on string parameter

Name :env

Default : prod

Description:

The click on save

Then go to project we have add option is build with parameters ,

Click on the ,

Env option change the name prod is rename : Dev

Then the build it.

Then same process then click on project the click on parameterited:

Click on choice parameters:

Name: Env

Choices: prod

Dev

Stag

Click on save

Then go back

Click on Build with parameters, go to env then more option prod,dev,stag click on build.

Webhook and Poll SCM:

It be go back to project

Click on project ,

Then go to configure

Go to source code management

Then click on git ,

Repositortories

Repositortories url:

Then you have github account :

Copy the http url copy

Paste the repositortories .

We have get any errors on url

Go to putty or command promote,

Give this commands

Cd ~

yum install git -y

then go back to Jenkins

refresh the url

then we select the branchs ,then default branch on master

then we go to github account we check the branch

then back to project to change the branch

then go to build triggers

click on poll SCM

schedule: \* /2 \* \* \* \*

click on save

and just wait 2 mins check the build history

every 2 mins build the job.

Then got to command prompt or putty

Cd /var/lib/Jenkins/workspace/

Ls

Cd samplefirstjob

We have two file

Then back to Jenkins

Create the new project

The give the name githubwebhook

Click free style project

Click on save it

Then go to project

Go to source code management

Click on git

Paste github http link

Go to branch change the branch specifier

Go to build triggers

Click on github hook trigger for GITScm polling

Then go to build

Click on add build steps

Click on execute shell

Write the command ./project.sh

ex: (“github project name”)

then click on save the project

then go to config

then go to build trigger

click on github hook trigger for gitscm polling

after reading the instructions

click on save it

after the go to github project

go to setting

go to webhooks

then go to webhooks

then delete the url and create the new one

then add webhook

payload url :

add the url : Jenkins url and github-webhook/

then the click on

just the push event

------

but not working webhook then go to putty or command prompt

some commands:

cd ~

wget <https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz>

ls

sudo tar xvzf ~/Downloads/ngrok-v3-stable-linux-amd64.tgz -C /usr/local/bin

cd /usr/local/bin

ls

./ngrok http 8080

Then go to ngrok by inconshreveable

Copy the forwarding url : <https://ea9d-103-81-35-107.ngrok.io>

Paste the chrome

Open the Jenkins

Copy the url on chorme

Go to github account

Payload url : paste the url

Then update the webhook

Then connected the url

Go to click on samplefirstproject

Click on project.sh

Make so change in project

Then edit it

Add this this test complete

Click on commit

Then go to Jenkins

Build the project

Then the build history failed

Click on console output

Failed the project ,

Then the permission denied

Then the go back to configure the job

Just the change the build, execute shell ,then change the code

Give the permission

Chmod 774 ./project.sh

Click on save it.

If have go back to my github make different commit

Change the code

Click on commit change

Then go back to Jenkins

To check the build history then check it

Build success

We the output successfully

Fully video on webhook on Jenkins

Jenkins gitlab part 2

Channel is thetips4 you

Jenkins Maven Integration:

Then we go to maven lifecycle / Goal

Example:

mvn clean

mvn clean install

then default-lifecycle:

validate : test-compile

compile : integration-test

generate-test-resources : install

etc …..

then go to github

to take the project on spring boot

or java project

or small simple project

some dependences on github

to install the maven on Jenkins

go to putty go the some commands

yum install maven -y

instillation completed

to check the version

mvn –version

echo $JAVA\_HOME

echo $MAVEN\_HOME

alternatives –config java

vi /etc/profile

press the insert

export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-11.0.16.0.8-1.amzn2.0.1.x86\_64/

export PATH=$JAVA HOME/bin:$PATH

export MAVEN\_HOME=/usr/share/maven

save it

source /etc/profile

echo $JAVA\_HOME

echo $MAVEN\_HOME

then go to Jenkins

we create the project

then name : maven

click on free style project

click on save it

copy the url on github http

paste it Jenkins

branch change : change to main

then go to build:

click on invoke top -level maven targets:

goals

we write on goal compile

and save the project

let the build it job

build history id failed the project

console is failed.

Then change branch

Save it

And compile then

Build sucesss

Then go to putty or command prompt

Cd /var/lib/Jenkins/workspace/maven/target/classes

Ls

Cd /var/lib/Jenkins/workspace/

Ls

Cd maven/

Ls

Then go to back

Cd target/

Ls

Cd classes/

Ls

Cd example/

Ls

Cd smallest/

Ls

Let us go Jenkins

Go to configure

Change the build status

Change the package

And save it

Again build it

Then go to console output

Build success

Then go back to putty or command prompt

Check command

Ls

Completed to maven.

Creating users :

It how to create the user in Jenkins

Lets go to Jenkins

Then go to manage Jenkins

Go to security

Click on manage users

Then go to users

Let us create new user

Click on create users

User name : test

Password: test

Full name: testuser1

e-mail id: [test@gmail.com](mailto:test@gmail.com)

click on create

lets us create 2 more users.

Let us log out the Jenkins

Let us login test

Log in users

Lets login original users

Let go to manage Jenkins

Click on configure global security

Then no change the file

On click on save.

Role Base Access Control:

Let us go back to manage Jenkins

Click on configure global security

Go to authorization

Click on matrix-based security

See the option and permission

Then leave the page

Or

go to back to manage Jenkins

go to manage plugins

click available

then go to search

Role-based Authorization Strategy

Click on this

Then go to new page on instruction how to use this strategy.

Then go to back

Select the role based authorization strategy

Click on install and restart

Installation done then go back to manage Jenkins

Click on configure global security

We can able to see the role based strategy

Then select this option

Save it

Then I go back to manage Jenkins

Then go to security

We can see the new option

On manage and Assign Roles

Click on manage and assign roles

Let us create the own role us assign the roles

Then click on manage Jenkins

Lets we create the role and add

Give the name and click on add.

Let give the permission on job,users…..etc

Then add the other role ,

Then give the Dev and add

Give the permission on role.

Then click on save it.

Then go back to click on assign role

Click on them

We add user/group to add.

Give the user name: testuser1

Then add.

Give the permission on : operations

Then add the other name is user name is :testuser2

Click on add.

Give the permission on :dev

Then save it.

Let us go back to dashboard

Logout the Jenkins

Login the testuser1

Login to check permission

How permission you have in Jenkins

Again login the testuser2

Login to check the permission in Jenkins.

Then login original admistor

Then go to manage Jenkins then click manage and assign roles

Click on manage roles the to check the perimission on jobs

Completed.

Jenkins configure system:

Let go back to Jenkins home page .

Then click on manage the jenkins

Let us go back to configure system

click on that

To check the home directory

System messages

Hello to Jenkins

Click on save it.

Then go back to check the configure ,

System message ,previous or hide

Click on previous

Then go back

Click on manage Jenkins

Go to security

Click on configure global security

Then go to markup formatter

Click on option on the safe html ,

Click o this

Save it

Go back to configure system

Change the system message

<h1> welcome to jenkin</h1>

Save this

Come back

Then see it.

Then again go to configure system

Then click on executors

Give the 5 executors

Click on save

Then click the global tool configuration

To check it

Come back.

Introduction to Jenkins pipelines:

What is a Jenkins pipeline?

A Jenkins pipeline is a suite of plugins which supports implementing and integrating

Continuous delivery pipelines into Jenkins.

A continuous delivery pipelines is an automated expression of your process for getting software from version control right through to your users and customers

Jenkins pipeline provides an extensible set of tools for modelling simple-to-complex delivery pipelines”as code”.

The definition of a jenkins pipelines is typically written into a text file is called a jenkinsfile which is turn is checked into a project sources control respoitory.

Types of Jenkins pipelines:

There are two types of Jenkins pipeline code

1.Declarative pipeline

2.Scripted pipeline

Pipeline Concept:

. pipeline

. Agent/Node

.Stage

.Step

Create First Jenkins pipeline Job:

Then go to dashboard

Create the new job on pipeline project

Click on pipeline

Click on save it

Then go to Advanced project options

Pipeline

Definition

Change the option

Pipeline script click on this

Then go down the scroll down

Pipeline sysntax

Click on that

Then go to next open the new tap

Then got to steps

Sample steps

We can select the git.Git

Then the open the steps

Then go to github account

Copy the url https

Back to pipeline sysntax

Then paste the url

Then change the branch : main

And

Sroll down the one option there

Generate pipeline Script

Click on it.

Then copy the gitbranch

Go back to dashboard

Then go to script

Write some script

pipeline{

agent any

stages{

stage('Git clone'){

steps{

git branch: 'main', url: 'https://github.com/shazforiot/Samplefirstproject.git'

}

}

stage('Test'){

steps{

echo "Test completed"

}

}

stage('Build'){

steps{

echo "Build completed"

}

}

}

}

Save it build it

Compile successful

Then the check the console output

Testing is copleted

Build is compled

Then come back we can the stage view

Gitclone test build then check it.

Then go to pipeline syntax page

Lets us check the pipeline synatax

Giving the lot of information on pipeline synatax

Building Maven java Project With Jenkins Pipeline:

Create the new job on the pipeline

Click on ok

Then go to Advanced project options

Click on that

Pipeline

Definition

Pipeline script

Script.

Then go to github project

Copy the url https

Then the come to pipeline syntax

Paste the url

Then change the branch

Then click on generate pipeline Script

And generate the url

Then paste the url

Come to back to project dashboard

Write the some script

pipeline{

agent any

stages{

stage('Git clone'){

steps{

git 'https://github.com/shazforiot/HelloWorld-Springboot-App.git'

}

}

stage('Maven Test'){

steps{

sh 'mvn test'

}

}

stage('Maven Build'){

steps{

sh 'mvn package'

}

}

stage('Maven Deploy'){

steps{

echo "Deploying the war file to the server"

}

}

}

}

Then save it

Build it

Build success

Running Jenkins on Docker:

lets go to putty

install the docker commands

amazon-linux-extras install docker -y

docker v

systemctl start docker

systemctl status docker

clear

docker run --name Jenkins -p 9090:8080 jenkins/jenkins:latest

sudo yum install firewalld

sudo systemctl start firewalld

sudo systemctl enable firewalld

sudo systemctl status firewalld

docker ps

let login the Jenkins port number 9090

let stated the Jenkins on docker

let us cancel installation

let us remove the docker containers

docker ps -a

docker rm container id

Presistant Jenkins Data Volume:

docker run --name jenkins -d -p 9090:8080 -v jenkins\_data:/var/jenkins\_home jenkins/jenkins:latest

docker ps -a

docker volume ls

cd /var/lib/docker/volumes

ls

cd Jenkins\_data/

ls

cd \_data/

ls

Jenkins File:

Let go to Jenkins

Let go to mavenpipe project

Go to configure

Copy the pipeline script

Lets go to git hub

Create a floder

Then paste the mavenscript

Let go to Jenkins

Create a Jenkins project pipeline project

Go to advanced project options

Definition

To select the

Pipeline script from Scm

Scm is git

Paste the url git hhtp