



- Jenkins

Benefits

- ① Automated builds
- ② → ← Tests
- ③ → ← CI/CD pipelines
- ④ → ← deployments
- ⑤ Ability to install Jenkins locally
- ⑥ Supports plugins

Why we need Jenkins

ex. consider you got a code from Github and you have to check if the application is running smoothly the way it suppose to be to do that you will setup the environment locally to do that you install dependencies then you clone code you run the tests via running application using commands you browse through the webpages, checks api, checks the status into logs now imagine checking this code whenever there is change in code it can be two times a day or

even 100 times a day. that's painful
to prevent that **Jenkins**.

- Installing of Win
 - does need Java
- To have management and restriction over o users we need a plugin called
 - **Roles - based Authorization strategy**
- * Once its installed it has to be enabled from Authorization settings in Global settings.
- To add users & assign roles little bit confusing at first glance
- Alternative to this plugin for projects
 - **project - based Matrix Authorization**

- Main files to backup



Best option to backup use plugins

- **ThinBackup** is an good option.

↳ We have to put the path for Backup first in settings with appropriate permission to write in that directory

Then we can Backup and restore.

We can also add monitoring tools plugin to monitor the different metrics on prometheus.

To Scope Jenkins on prometheus we need to also add the endpoint settings inside the jenkin server

Here:-

The screenshot shows a terminal window with two panes. The left pane displays a configuration snippet for a Jenkins job named 'Jenkins' with a metrics path of '/prometheus/'. The right pane shows a root shell session where the user runs 'mvn test', which fails because 'mvn' is not found. It then runs 'service jenkins restart', which successfully restarts the Jenkins service. The user also runs 'echo hello' and 'vim /etc/prometheus/prometheus.yml', then 'service prometheus restart', which successfully restarts the Prometheus monitoring system.

```
task Solution 49:51 Terminal 1 +  
1 2 3 4 5 6 7 8 9 10 11 12  
  
Let us now add a scrape config for Jenkins. Update the prometheus configuration file located at: ./etc/prometheus/prometheus.yml and add the following config under the scrape_configs section:  
- job_name: 'Jenkins'  
  metrics_path: /prometheus/  
  static_configs:  
    - targets: ['localhost:8085']  
  
Once the file is updated, restart prometheus using the below command:  
service prometheus restart  
  
Wait for few seconds before submitting your question.  
  
Note: Here localhost:8085 refers to our Jenkins Controller and the /prometheus/- metrics path is exposed by the prometheus-metrics plugin that we installed previously.
```

```
root@jenkins-server ~ ➔ mvn test  
-bash: mvn: command not found  
root@jenkins-server ~ ➔ service jenkins restart  
* Restarting Jenkins Automation Server jenkins  
Correct java version found  
Correct java version found  
Setting up max open files limit to 8192  
  
root@jenkins-server ~ ➔ echo hello  
hello  
root@jenkins-server ~ ➔ vim /etc/prometheus/prometheus.yml  
root@jenkins-server ~ ➔ service prometheus restart  
* Restarting Prometheus monitoring system - prometheus  
  
root@jenkins-server ~ ➔
```

We can see the current configuration in prometheus UI too → status → config

- Jenkinsfile

pipeline - CI, CD, CI/CD
task that you trying to accomplish

agent any → Build agent
stages - build, dev, prod
steps - work to be done in the pipeline

two types

- ① One stage
- ② Multi stage

- To actual create an image from the build we can use CI/CD pipelines

Heart Running a script &
Building docker image from that.

- To build a pipeline we need an pipeline plugin installed first

- Build Agents
 - ex - Windows
 - Linux
 - Centos
 - MacOS
 - Docker

Why we need Build Agent for Jenkins ?

- To test and run the image in isolations and with security standpoint

If we run the build on the same jenkins server that might cause issues to the jenkins server and speed might be decreased.

* To add a new build agent

We create a new user in that server
then we put the same credentials inside

and password as linux

Once done that we add a new node
to have a build agent to test things

- permanent Agent
- Manually verified key strategy

- To run agent on particular systems we have to restrict where this agent runs.

To build new agent

- ① get new container/server
- ② Add new user Jenkins
- ③ give permissions to user `usermod -aG Jenkins`
- ④

In base Jenkins server do this

① Dashboard → Credentials → System →

Global credentials → Add new →

Add same username and password

② Add node from manage Jenkins → nodes

Add the same userinfo we created

- Add name → usage → only builds with label expression matching this node.

- Add path

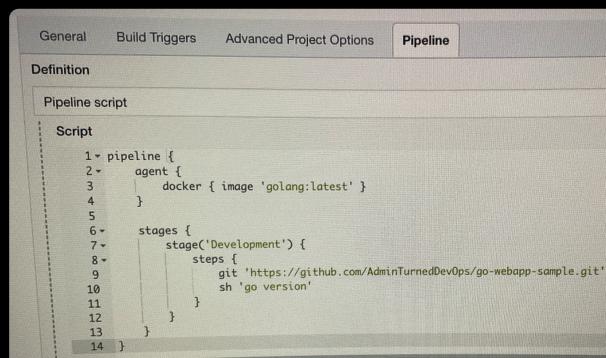
- Login via SSH

- Host key manually verify

* * * We can create an ephemeral containers images for the builds which will have all the environment already set up and then we can run the builds on the same.

that's actually smart.

ex. pipeline code



```
General Build Triggers Advanced Project Options Pipeline
Definition
Pipeline script
Script
1~ pipeline {
2~   agent {
3~     docker { image 'golang:latest' }
4~   }
5~ 
6~   stages {
7~     stage('Development') {
8~       steps {
9~         git 'https://github.com/AdminTurnedDevOps/go-webapp-sample.git'
10~        sh 'go version'
11~      }
12~    }
13~  }
14~ }
```

- Blue Ocean ← Best Theme!

- Security
 - Add roles based security
 - Add limitations to the user
 - Add Security Plugins -
 - Role based Authorization
 - Authorize project

Exclude Built-in node from
running any executors to dedicate
it for Jenkins only

- Go to that node in settings and put the no. of executors as 0.
- for Agent Security we must use plugins.
- Matrix Authorization ← Plugin

- Jenkins final Lab

① We installed required plugins such as

- SSH agent
- Pipeline
- Docker
- Authorization tokens

② Then we added credentials to odd nodes

Manage Jenkins → manage credentials → global → Domains → Add credentials
→ username pass of that user from your Linux Server

③ Add node

Manage Jenkins → Manage nodes & cloud → New node → name →
• permanent agent → Remote root directory (which you created and given a proper permissions
true) → Launch agent via ssh →
agent name → Non verifying veri strategy → Save

④ New item → Development name → pipeline → Add script → save

Thoughts on course Notes

- Not much to write since most of the content is via demo.

I personally think this notes are just one time record.

- Be the witness!