

2.16.3

Jenkins: →

Jenkins: → Jenkins is a leading Open-Source automation tool written in Java with plugins built for Continuous Integration(CI), Continuous Delivery(CD), Continuous Deployment purpose in Software Development.

(CI/CD/CD): → It is a software development practice in which we will automatically build, test & deploy software projects. Everytime whenever a developer pushes the code changes to the repository

Continuous Integration: → pulling the code from repository and building the application with the help of build tools automatically.

Continuous Delivery: → Along with CI if we automatically deploy the application on test environments i.e. Continuous Delivery.

Continuous Deployment: → If we automatically deploy the application on production environment, after the testing on test env, the entire process automatic is called Continuous Deployment.

Advantages of Jenkins: →

- * Open-Source tool with great community support.
- * Easy to install, easy Configur.
- * 1000+ plugins are available.
- * free of cost.
- * it is built with Java, it is portable to all major platforms.

→ Any alternative tools do you know for CI/CD? →

→ Bamboo / uDeploy / teamcity / tfs . . .

→ Types of jobs u configured in jenkins: →

→ Freestyle, maven. } Jenkins Dashboard Newbie → Other Good →

9) Types of jobs are available in Jenkins:

→ Freestyle project, Mavenproject, Pipeline, Multi Configuration
Project, ExternalJob, Folder, Github Organisation, Multipeline.

10) Diff b/w freestyle & pipeline jobs?

→ Freestyle job is configure everything in web UI

→ Pipeline job is configure to write groovy DSL script

11) Pipeline: → A pipeline is a userdefined model of a CI pipeline. → A pipeline code defines your entire build process, which typically includes stages for building an Application, testing it and then deploy it.

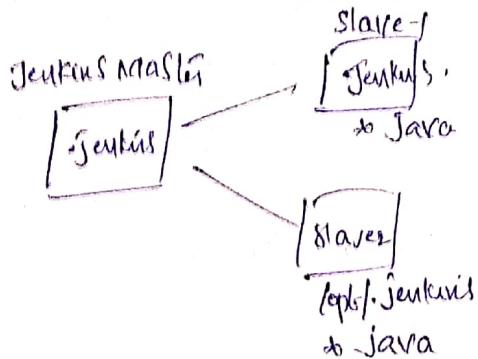
12) Declarative pipeline: →

13) Diff b/w Declarative pipeline / Scripted pipeline

14) write the pipeline Syntax.

15) what is master-slave architecture? use of master/slave?

→ If u want run 2-10s², or extreme u can go with master-slave



→ Jenkins master will distribute the load to multiple slave.

16) How many ways we can connect the slaves.

→ launch agent via SSH

→ launch agent via web start

→ launch agent via execution of command on the master

→ ~~Windows agent~~ lets Jenkins control this windows slave as a window service

17) How to add node.

Jenkins Dashboard → Manage Jenkins → manage nodes → New node

18) How many ways we can provide security for your Jenkins Server?

19) What type of artifactory repository tool we're used → Nexus/Jfrog/S3

20) Use of artifactory tool → u can maintain multiple versions of our project.

21) How you declare a variable in pipeline.

22)

22) what is upstream/down stream projects? what is the use of it?
which Scenario you Configure

- if want to have the dependencies of one project to another project
- if u want to configure multiple jobs in a pipeline flow.

23) How to take backup my Jenkins? → thin backup plugin.

24) How to delete old builds automatically

Jenkins → Jobs → General → Discard old builds

25) pre requisites for using Jenkins.

- A Source Code repository which is accessible, for instance, a git repo
- A working build script : a maven script

26) Commands to start Jenkins manually.

- To Start Jenkins: Jenkins.exe start
- Jenkins.exe stop
- Jenkins.exe restart

27) How can u setup Jenkins Job

Jenkins Dashboard → New item → freeStyle project → General →
Run → Build triggers → Build Env → Build → post-build actions.

28) How to create a backup & copy files in Jenkins.

• install thin backup plugin

• → create directory mkdir Jenkins_backup

chown -R Jenkins:Jenkins Jenkins_backup

• goto manage Jenkins → manage plug → give the directory name

29) how will u secure Jenkins?

→ manage Jenkins → Configuration global security → Enable security →

Authorization → (1) Matrix based security

(2) project Based - Matrix Authorization Strategy.

30) How can u deploy a custom build of core plugin

(1) Stop Jenkins

(2) Copy custom hpi -to → \$jenkins/plugin.

(3) Delete previous expanded build plugin directory

(4) Create an empty file <plugin>.hpi-pinned,

(5) Start Jenkins

(31) What u do when u see a broken build for ur project in Jenkins

→ Open the console opf of broken build and try to see if any files were missed, unable findout any issue . I will clean & update my local workspace to replicate the problem on my local. Solve it

(32) Explain how you can move (or) copy Jenkins from one server to another?

Copy .jenkins folder from present Jenkins server to new server.

(33) what are the various ways in which build can be scheduled.

1. SCM commit

2. After Completion of other builds

3. CRON (scheduled at specific time)

4. Manual Build request S Times

34) which SCM tools Jenkins support.

CVS, Subversion, git, Mercurial, perforce

35) Two Components Jenkins mainly integrated with.

VCS like git/SVN

Build tools like Ant/maven.

36) What is the programming language used to build Jenkins?
Jenkins open-source Automation Server written in "JAVA"

37) Features of Jenkins,

1. CI/CD
2. Easy installation, Easy Configure, plugin, Extensible
Distributed.

38) How many types of start Jenkins

39) Which commands can be used to start Jenkins man
Jenkins.exe start

40) How to make sure that your project builds doesn't break in Jenkins

Before scheduled the job, u have to perform
Pre-check build list

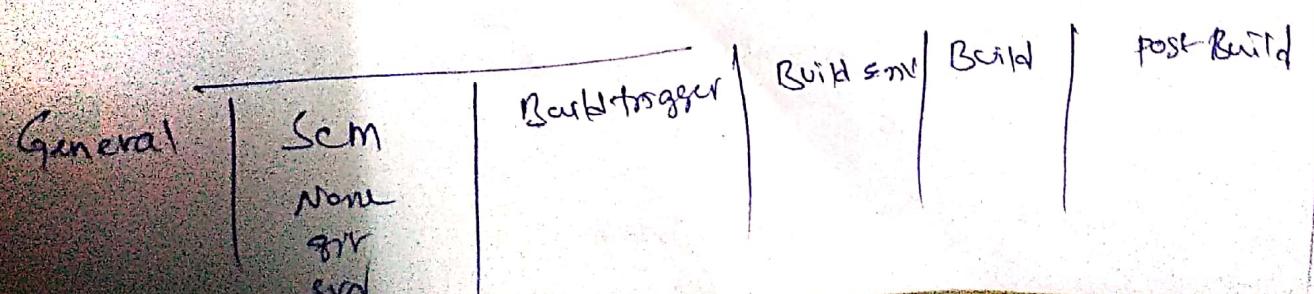
41) How can u clone a git repository via Jenkins.

u have to enter email & username for ur Jenkins System.
Switch to your job directory and execute "git config"

42) How to configure 3rd party tool in Jenkins.

1. Install 3rd party software
2. U need to have

43) Blue Ocean:→ Blue Ocean is a new user interface and provides and
provides an interactive view for Jenkins pipeline



Build Triggers

- Build whenever a snapshot dependency is built (Only recent job)
- Trigger builds remotely (e.g. from script)
- Build after other projects are built
- Build periodically
- Github hook triggered for Git SCM polling.

Build Env

- Delete workspace before build starts
- Use Secret (env) files
- Provide Configuration files
- Send files (or) execute commands over SSH before the build starts
- Send files (or) execute commands over SSH after the build runs
- Abort the build ~~if~~ if PMS & cut
- Add time stamp on the console output

⇒

General

- Discard old builds
- This project is parameterized
- Restrict where this project can be run.
- Execute concurrent builds if necessary.

Build:

- (1) Execute Shell
- (2) Invoke Ant
- (3) Invoke top level maven-targets
- (4) Send files (or) execute command over SSH

Post-build

- Email notifications
- Build other project
- Delete workspace when the project is done
- Publish Junit test result report
- Stop Docker containers

→ After Build other projects (Set upstream & downstream)

- + Build pipeline View
 - Select initial job.

Jenkins Dashboard

- Network
 - people
 - Build history
 - Project relationship → upstream downstream (Component).
 - Check file fingerprint → U got a file, but u don't know which version it is
So find out by checking its Fingerprint against the database in Jenkins
 - Manage Jenkins
 - My Views
 - Locatable Resources
 - Credentials
 - New View
- Credentials → You can see all jobs

New view (d) Build pipeline view

With View
My View

plugins

- 1) Backup plugin
- 2) Build pipeline plugin
- 3) Build Timeout
- 4) Greenballs
- 5) Maven Integration plugin
- 6) Email extension plugin
- 7) Publish over SSH
- 8) Role-based Authorization

Manage Jenkins

- (1) Configure System → Configure global settings & path
- (2) Configure global Security → Secure Jenkins server.
- (3) Configure Credentials.
- (4) Global tool Configuration → Configure tools & locations
- (5) Reload Configuration from Disk → (Discard all the loaded data in memory & Reload Everything from file System)
- (6) Manage plugins
- (7) System information → Plugins that can extend the functionality of Jenkins
- (8) System log → Displays environments.
- (9) Load Statistics → Capture outputs from java.util.logging
- (10) Jenkins CLI → Check your resource utilization.
- (11) Script Console
- (12) Manage Nodes
- (13) Manage and Assign Roles
- (14) In process
- (15) Manage users.
- (16) Backup manager
- (17) Prepare for Shutdown (Stop executing next build)

(ACL's) - ?

Configure Global Security

Enable security.

Security realm

→ Jenkins Own User Database

Authorization

- (1) Anyone can do anything
- (2) legacy mode
- (3) logged-in user can do anything
- (4) Matrix Based Security
- (5) project based matrix authorization strategy
- (6) Role-Based Strategy

Manage Plugins

Update | Available | Installed | Advanced

Jenkins → Manage Jenkins → Manage Nodes → New node

Launch method

- 1) Launch agents via SSH
- 2) Launch agent via Web Start
- 3) launch agent via execution of command on master
- 4) let Jenkins control this windows slave as windows service.

Manage and Assign Roles.

465

- Manage Roles
- Assign Roles
- Role Strategy Macros

Backup manager

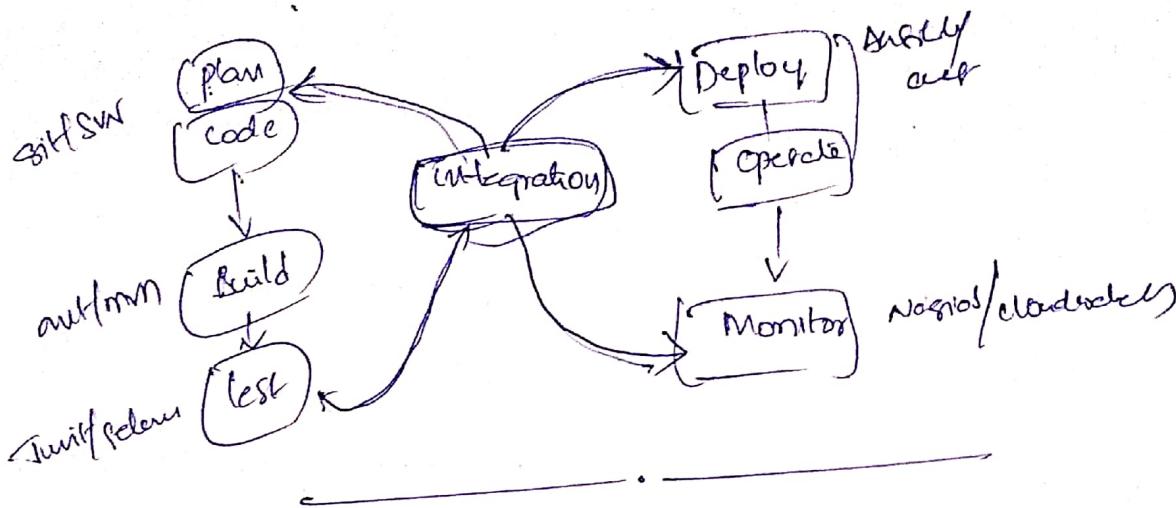
- ① Setup
- ② Backup-Hudson Configuration
- ③ Restore-Hudson Configuration

Jobs

- ① Freestyle project
- ② Maven project
- ③ Pipeline
- ④ GitHub organization
- ⑤ Folder
- ⑥ Multibranch pipeline
- ⑦ ~~Parameterized project~~
- ⑧ Multi Configuration project

Jenkins Issues:

- 1) Timeout issues (long build)
- 2) plugin issue
- 3) file path mismatch
- 4) Jenkins site down, not available
- 5) heap memory issues
- 6) user permission issue
- 7) memory spike
- 8) disk full
- 9) Site is up but no job is displayed (AWS seems to be slow but didn't mark properly)



Pipeline job

General | Build Triggers | Advanced project options | Pipeline
 → Pipeline Script
 → Pipeline Script for Scm

Jenkins pipeline → is a declarative pipeline
Declarative pipeline

Pipeline {

agent any

Stages {

Stage ('intro project') {

steps {

sh "cd \$root/introproject"

}

Stage ('compile') {

steps {

sh "mvn compile"

}

Stage ('deploy')

steps {

sh "cp \$root/devels/world"

- - targets/mypr

/opt/apache-tomcat/web

}

}

}

}

}

}

}

}

}

}

}

}

}

}

- ① Node is a machine which is part of the Jenkins environment and is capable of executing a pipeline.
- ② Jenkinsfile → Jenkins pipeline details are stored in Jenkinsfile.
- ③ Control flow in Jenkins pipeline → Scripted pipeline are executed sequential order from top to the downwards then it is called Control flow.
- ④ Agent: the agent section specifies whether the entire pipeline will execute in the Jenkins environment depends on where the agent section is placed.
- ⑤ post: it defines one or more additional steps after running the pipeline.
- ⑥ Stages → having sequence of one or more stage directive.
Each stage directive having it's the part of Continuous integration.
- ⑦ Step → One or more steps to be executed stage directive.
- ⑧ Environment directives → specifies a sequence of key-value pair which will be defined as env variables for all the steps.
- ⑨ Parameter directives → parameter directive provide list of parameters when user should provide when triggering the pipe.
- ⑩ How you can deploy a custom plugin of a core plugin.
Stop Jenkins.
Copy .hpi → Jenkins_home/plugins
Delete previous plugin file.
(plugins.hpi.pinned)
Start Jenkins

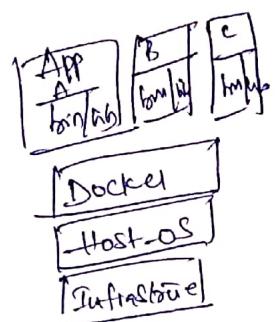
Docker

[18.03.1 C2]

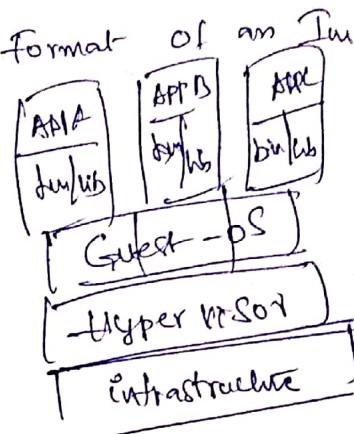
Docker → is a containerization tool if is a tool for developers & system admins to develop, deploy & run applications with containers.

Image:→ Image is an executable package that includes everything needed to run an application

Container:→ is a executable format of an Image



Docker



VM

VM

- more time to create
- Dedicated 'OS'
- Heavy weight
- Ram cannot share
- low performance
- complex to create

Docker

- less time
- common OS
- lighter weight
- Ram shared
- high performance
- easy to create

Getting a Image Slipped

Docker pull

Docker Commit containerId name (by container)

Docker Build -t . (by Dockerfile)

Dockerhub :→ Dockerhub is a cloud-based registry service

Docker Compose → Set of containers.

1. Dockerfile
2. docker-compose.yml
3. docker-compose up
- 4.

→ [docker log -f].

Docker Swarm:

Docker Swarm mit --listen-addr ip:port: 2377

Node: → A instance of the Docker Engine participating in the Swarm.

① 2377 (Swarm management port)

Virtual machine uses own OS → Docker uses host OS

Dockerhub: → Dockerhub is a cloud based registry service.

* Docker Swarm: → Docker Swarm is a cluster of Docker engines (or) Node where we deploy services.

Manage Node will control the Swarm Nodes

Dockerfile: → Docker can build images automatically by reading the instructions of Dockerfile.

+ What platform can Docker run

Ubuntu

CentOS, 6+

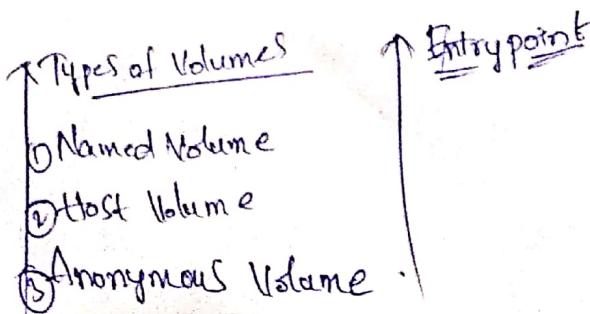
Fedora

RHEL

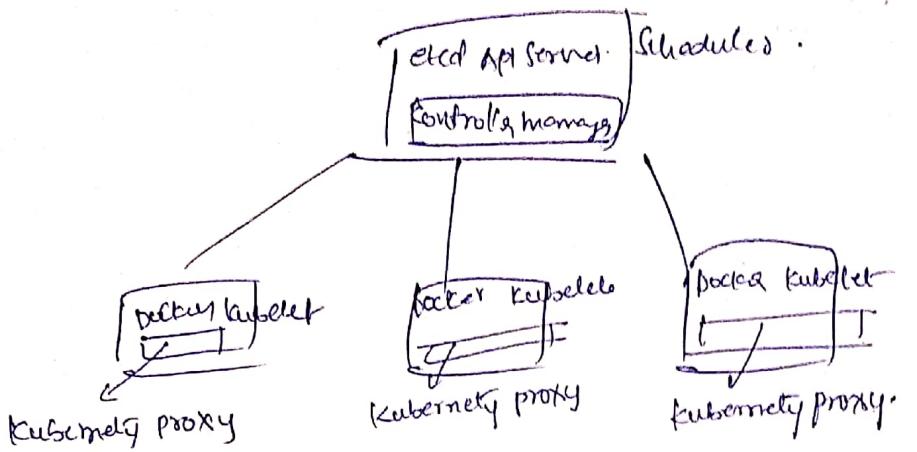
OpenSUSE

Default networks in Docker

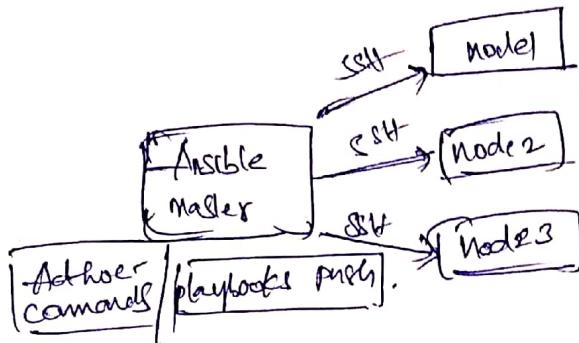
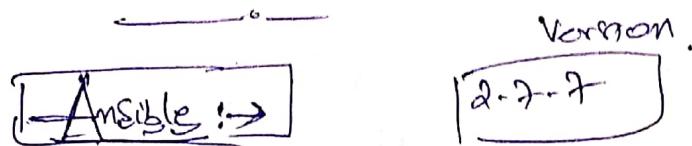
- ① Bridge Network
- ② Host Network
- ③ None



Kubernetes is a container management technology developed in Google Cloud.



Playbook vs Play
organise tasks / set of actions
set of playbooks
working in single playbook
may result in unscalable solution.



Ansible: → Ansible is command tool, it allows to user to deploy and update application using SSH

UIC: → managing IT infrastructure and deploy software apps to remote nodes

Inventory → contain information about the servers.

ansible all -m file -a 'name = /opt/leand
node = * Owner = group'

ansible all -m file -a 'name = /opt/file1 state=touch'

ansible all -m cron -a 'name = 'myjob' minute='0' hour='1'

Job = 18-1>/tmp/file.log 10

VIM (by me)

- hosts : all

- tasks:

- name : to update repo

| Models: Models are dealing with package.

| Ping to node connection

| Setup node information.

p - install

v - verbose

m - host

rpm (redhat package manager)

rpm -iVh httpd_package (package install)

rpm -Uvh httpd_package (update)

rpm -e httpd_package (remove package)

rpm -q httpd_package (to query package)

yum list installed | grep Java

Systemctl tomcat

yum list available | grep tomcat

Systemctl stat tomcat

yum list → list all packages

Systemctl stop tomcat

yum update

apt-get install OpenJDK-7-jdk -y

Service apache Start

apt-get remove apache-2 -y

Stop

(del all files)

Restart

apt-get purge apache-2 -y

(del all unseed file)

apt-get autoremove -y

Webserver → to deploy static webpages

apache, nginx

port = 80

/var/www/html

app → to deploy dynamic pages & static

/var/lib/tomcat7/webapps / Redhat

Tomcat, weblogic, websphere, JBoss

/var/lib/tomcat7/webapps → Oracle

8080

Distribution method!

yum → yellowdog updated modified.

Rpm is unable to track dependencies

while yum can

Ansible galaxy : → where users can share roles, and to a command tool for installing, creating and managing roles.

- installing roles → ansible-galaxy install role_name
- Creating roles → ansible-galaxy init role_name
- Search roles
- Remove an installed role.
- Import role

Ansible Vault : → it allows to keep the sensitive data (w/ passwords) in encrypted format.

```
ansible-vault create pb1.yml
edit
rekey
encrypt
decrypt
ansible-playbook -o
```

Ansible Tower : →

wh

Ansible Tower : → is a web-based solution, that makes Ansible even more easy to use for IT teams of all kinds

Ansible modules:

- Cloud
- cluster
- Command
- Database modules
- File modules
- Identity
- Inventory
- Message

Module! Dealing with packages

- Monitoring
- Nut-wheels
- Notification
- package module.
- Remote management modules
- Source Control modules
- Storage system

⇒ ansible all -m shell -a ' apt list -installed | grep tonicet'
↳ Verify the installed packed ^{remote by} in Ansible Controlled machine (master)

Copy module
Remote_src : yes → copy a file on the remote machine for editing.

⇒ Method to check inventory vars defined for the host.

ansible -m debug -a "var = hostvars[¹hostname]" localhost

Variable Name: It can be built by Adding Strings

Environment var. → To access EV you need to access existing Variables

facts: → it contains node information.

to run a particular task on playbooks

tags:
package

-name: --

-- -- .

tags:
service

Ansible-playbook --yaml --tags "package" --

Ansible-playbook --yaml --skip-tags "no"

unarchive:

-name: extract Soub2- > ,

Unarchive:

Src : JavE2

dest : path

| If zip file in
| remote node
| ↓
| Remote_src : yes.)

archive : How to execute a playbook on particular host
 name : --limit "ip" ?
 archive : hostname
 path : or
 dest : groups
 /
 Ansible written python & PowerShell
 requirements → Linux / Windows , Python 2.6 Version and higher
 Setup → to get file information
 ask_pass is a control in Ansible playbook
 ask_pass = true
 utilities modules are used for developers creating own modules

→ Ansible written python & PowerShell
 requirements → Linux / Windows , Python 2.6 Version and higher
 Setup → to get file information
 ask_pass is a control in Ansible playbook
 ask_pass = true
 utilities modules are used for developers creating own modules

Copy files
 print \$read A1s
 read f1 f2
 cp \$f1 \$f2
 echo done copied

arithmetic operators
 print "real values"

read a b
 echo ADD = `expr \$a + \$b`
 /*
 /

print "in enter a value"
 read n
 if [\$expr \${n%1.2} -eq 0]

then
is valid even

else
echo odd
fi

```

for ip in $(cat ips)
do
ping $ip -c 1 > dev/null
if [ $? -eq 0 ]
then
valid
else
invalid
fi
done -
  
```

read username -

grep \$username /etc/passwd > dev/null

if [\$? -eq 0]

{then

echo valid user

else

invalid user

fi

given file is regular (or) Directory

read f1

if [-e \$f1]

{then

if [-f \$f1]

{then

echo file is regular

elif [-d \$f1]

{then

echo file is directory

else

else file is not exists

fi

file exist

prompt "Enter value"

read f1

if [-e \$f1]

then file exists

else

doesnt exist

fi

1 2 3 4

read op

case \$op in

1) hostname -

2) Date

3) cal

real Greg Sy

4) invalid option

present in

cal \$my \$y

esac

Checking port status of Tomcat

and it is down going to start automatic

-i cally

if result = netstat -nat grep 8080 | awk

{print \$7} | wc -l

then

if (result > 0)

{then

echo - mail -

else

echo file DK

fi

Net
out
grep port
mail sent
file DK

81

81

Ansible Replace module :-

```
---  
hosts: all  
become: true  
tasks:  
  - name : Replace example
```

```
    replace:  
      path: /home/ansible/mirrors.txt  
      regexp: "Hello"  
      replace: "World"
```

{ Hello : changes "Hello" to "World" }

```
---  
hosts: all
```

```
tasks  
  - name: Ansible Zip directory example
```

```
archive:
```

```
  path:  
    </home/ansible  
  dest: /home/ansible/mirrors.zip  
  format: zip
```

"....."