Git

Ensure that you have git client installed

If you want to install git on any linux machine , its yum install git / apt-get install git

Type git to see if it's installed .

Create github account

https://github.com/

Please give your email , username and choose a password

Verify the mail and you should be able to access

To make accessibility easier , please sure the google docs from inside the lab , its Convenient for you to refer and paste few commands for practice

With the git client

Git bash -linux
Git cmd - windows
Git GUI will be installed - graphical user interface

Start using git bash by going to search

Go to c drive

Create a folder called gitpractice

commands

Cd c:

Mkdir gitpractice

Cd gitpractice

Ls (its empty folder)

To initiate a git repo on local system

Use

Git init

You can go back and check the folder , we have lot of contents related to tracking the files (config files)

Ls -a (it will list all files , hidden files as well , .git)

Create a file inside git practice (ensure you are at the root directory of git practice)

Vi hello

Press i to insert ,

Type sample text

Press escape

:wq to save /quit

Now type

Git status

Shows untracked files

Hello.java

Now we can add them

Git add Hello.java

Now type **git status**

It will add to the staging directory (this staging is like where you pile up all the files before you go for a commit)

Incase if you want to unstage

You can use the following command

Git rm --cached Hello.java .

Exercise 1 : Create a text file , initialize the git repo , check the status , add to the staging And unstage it later .

Exercise 2: Try deleting the unstaged file , experiment what happens if you delete a Unstaged file , also experiment what happens if you delete a staged file

Exercise 3: Try committing the staged files

Git commit -m “made changes to Hello.java”

If it asks for your identity

Git config --global user.email “srikanth@gmail.com”
Git config --global user.name “srikanth”
Please give your email and username (not mine :))

Now go to github.com

Open with your credentials : username and password

Once you login , you need to create a remote repository

On the extreme right hand side , you will see a + symbol

Click on it and add a repository

Give a repository name

Select public (by default)

Create repository

Now it will show the steps , after creating the repository

The common steps that you need to follow on your local prompt are the following

git remote add origin https://github.com/SrikanthPB/sampleabc.git (repo name)

git branch -M main

git push -u origin main

While pushing , git will ask you to enter the credentials , Please enter the credentials

It may open a new page where you will enter the credentials , please enter

And authorize the git to use it .

Now you can see everything is pushed to your remote ,

Go Back to the github and check if everything is pushed

To check the branches at local gitclient

Use

Git branch -l (local)

Git branch -r (remote)

Git branch -a(all branches)

Branching strategy

Always we keep our main branch untouched .

We can always create a new branch preferably matching to your JIRA ticket (or any bug tracking tool number)

Git branch feature-101
This will create branch

In order to move to the feature-101 branch , give the following command

Git checkout feature-101

Start fixing the issue on this branch -> and commit it (later we can merge if wanted)

Modify the existing file add some text
Git add Hello.java

git commit -m “Added”

Push to the remote origin feature-101 branch by giving following command

Git push -u origin feature-101

Sometimes , if we want to ignore few unnecessary files , we can use the .gitignore

Vi .gitignore

Add something with wildcard so that all such kinds of files will be ignored

Like

*.bak (all bak will files will be ignored)
*.txt (all txt files will be ignored)

Please add .gitignore file to staging and commit it as well

Git add .gitignore
Git commit -m “added”
Push to the remote (feature-101)

Whenever we type git status

Usually git can pick up all the local related commit information so , after a new commit
It says your local repo is ahead of the remote / origin .

When want an information whether your remote repo is having a commit ahead of your local branch , we should use git fetch (so that it fetches all the information about the newest commits happened at the remote repo)

When we want to make a merge go to the branch where we want the merge to be taken place

Go to the main branch

Git checkout main

Now type **git merge feature-101** (so you are getting all the files from feature-101 to main)

Difference between git fetch and git pull
Git fetch will get only the commit information
Git pull will do merging and also fetch

Git pull = git fetch + git merge

Stash

When we unintentionally do coding on a wrong branch , We dont need to rewrite again

Instead we can use stash

For this

Example :

Vi Hello.java

I typed , hi from main branch (my intention is main , but am in feature 101 branch)

What I can do is in this type

I can type

Git stash or git stash save
This will steal all data and put it into a temporary location .

Now we can go to the other branch and type

Git stash list (and see all the list of stash available) , In order to make it effective

We can type

Git stash apply stash@{0} (this is the one which we wan to stash)

Lesson of day1 :
The most famous commands in git from adding to ending is
Git init
Git add
Git commit
Git push

Git cherry-pick

Git cherry pick is used when you already committed the code (stash is used before commit)

But one you commit the code in the wrong branch we have to use the cherry-pick

Now, we will go the right branch and you can use the following cherry-pick command

Git cherrypick commit id

Git log (will help you to cherry-pick any of the commit id , but mostly we will pick up the recent commit id)

In my case ,

Git cherry-pick 9eef

After the above command , It will get the modified file , Now I can go ahead and use the same commit

Exercise 4 : Try committing a new file on a feature branch and then use the cherry-pick command to get the committed code from another branch to this branch.

Difference between stash and cherry-pick : stash is used before commit (only after staging) , Cherry Pick is used after the commit

Git show

You can use git show to see particular changes made during the commit .

Git show commit id

Excercise 5 : Please use **git show commitid (give your commit id)** , to see changes you made during a particular commit

Git reset

Git reset is used when we want to rewind back to old commit (so now your head will be pointing towards to the older commit)

[gh43] [6454] []3232]
||
HEAD

Currently my git head is pointing at gh43 , if I want it to rewind / goback to earlier commit

I can use

Git reset --hard 6454

This will take me back to the earlier commit (even in logs you can check)

Now after reset

[gh43] [6454] []3232]
||
HEAD

Git add all files

Git add -A

Git add filename1 filename2 (this will add these two files)

Git commit all files

Git commit -m “commit all files” -a

Git commit -m “commit file1 and file2” filename1 filename2

Git diff branchname1..branchname2

We can check the differences between the two branches

Exercise 6 : try to find out the differences between the branches
Observe the changes

Git GUI

This is a graphical interface tool , Which we can use with Windows and Mac only.

We can do many things like staging , adding and committing

You can make changes and see them by clicking rescan inside GU and you can seeunstaged , stage to commit from the commit option on the top ,

Once you see the changes in the staged
Commit by giving message and push it (while push ti will ask for a branch)

Exercise 7 : Please add somes changes on local , stage them , commit them

Deleting a branch

Git branch -d branchname --force

Maven

Maven is a build tool which is widely used for dependency management , packaging , versioning And execute most of the SDLC goals like

Clean
Compile
Install
Package
Test
Deploy

To get started with the installation , you will need to download maven and extract it

For example :

I downloaded in my downloads folder

Now copy the following path

C:\Users\Administrator\Downloads\apache-maven-3.6.3-bin (download here in my case)

Inside the bin we have the mvn command which is used for running our maven projects.

Setting up the path .Inside the environment variables , so that we can run maven command from anywhere in our system .

Right click on this PC go to advanced system settings and click on
environment variables , so inside the environment variables
Go to path and click edit to edit the path variable
 and inside the path variable click new and add the following

C:\Users\Administrator\Downloads\apache-maven-3.6.3-bin

Click ok for all

Maven for linux installation

Yum install maven (centos)
Apt-get install maven (ubuntu)

Sample project , please download from the following

https://github.com/SrikanthPB/nexus

XML

Stands for expensive mark up language (the tags inside xml are user defined)

Unlike html (which has predefined tags , which means you cannot use userdefined tags)

Every xml tag will have starting and ending

<starting> -- this is the starting of the tag
</starting> → / represents ending of the tag
Between these starting and ending tags we will insert our data
(mostly xml is used for communicating or information passing scenarios)

After downloading , open pom.xml using Edit plus

Replace the existing lines (1-9 lines) with the following lines

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>1.0.0</modelVersion>
  <groupId>com.accenture.mavensample</groupId>
  <artifactId>accenture</artifactId>
  <packaging>jar</packaging>
  <version>5.0.0</version>
  <name>my-maven</name>
  <url>http://maven.apache.org</url>
```

We have changed the model version to 1.0.0 (any new project starts with 1.0.0
Groupid make it as com.accenture.mavensample
<groupid>com.accenture.mavensample</groupid>
<artifact>accenture<artifact>

Prerequisite for maven is java , ensure java is installed

You can check by typing java in the command prompt

Now time for hands on session ,

C:\Users\Administrator\.m2\repository

You can look at the repository you may have probably some dependencies which are previously Downloaded .

Go to the project root directory where pom.xml is located .

And typ was e cmd in the top address bar (remove any other path , only type **cmd**)

mvn compile

Compiles all the files

mvn clean

Clean will delete the target directory , go back and check , target folder will be deleted
This folder has contents related to previous maven builds .

mvn install

This will install all the dependencies and if any plugins are required will be downloaded

As per our example , it has taken us 1:18 sec

Lets delete everything in the repo

C:\Users\Administrator\.m2\repository

and issue the **mvn install** again

Observe the difference ,

Basically , since we deleted all in the repo it started downloading everything again

Exercise : 8 Please use mvn install by deleting everything in the local repo
C:\Users\Administrator\.m2\repository

And please let us know how much time it has taken post deleting the repo

mvn package

mvn package is used to mainly package your project as per the given package option inside pom.xml

Change the <package>**jar**<package> to **war**

Project may fail because web.xml is not present , please include the plugins as follows line 38 in editplus

```
<plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-war-plugin</artifactId>
        <version>2.6</version>
        <configuration>
          <failOnMissingWebXml>false</failOnMissingWebXml>
        </configuration>
      </plugin>
```

Code updated

https://github.com/SrikanthPB/nexus/blob/main/pom.xml (this always the latest code)

Exercise:9

Now you can combine goals and give something like

mvn clean package

If you want latest artifacts (no aging artifacts) , combine with clean

mvn test

Test is used for running junit test cases written inside the class

Nexus

https://drive.google.com/file/d/1Hm9IFsWd6QgqiwnPy3QILBmRoa0vavX3/view

Please access the above link and download the nexus document

Download

https://sonatype-download.global.ssl.fastly.net/repository/downloads-prod-group/professional-bundle/nexus-professional-2.14.19-01-bundle.zip

Extract the folder

Go to

C:\zarantech-devops\nexus-professional-2.14.19-01-bundle\nexus-professional-2.14.19-01\bin

Type cmd in the address bar

go to the location and run nexus.bat install

now run **nexus.bat install**

run nexus.bat start

it will take time to start nexus

Now go to browser

localhost:8081/nexus

You can navigate to nexus where it asks you for license

Now for the license , we need to create one

I have created using my gmail

https://my.sonatype.com/profile/licenses

here we need to download the license after giving the credentials

click download license

you will download a file

sonatype-repository-manager-trial

Now go back to the nexus page where we started on local

http://localhost:8081/nexus

click already have license , upload the license by navigating to the right directory

now click I agree , license will be installed successfully

The page should look something like above after login successfully

On left hand side click repositaries

We will create a repositary

Click on add button as following screenshot

now click hosted repositary

Select save at buttom

I will totally create two of them , one for release and one for snapshot

now get the url of it

http://localhost:8081/nexus/content/repositories/nexus-zarantech-release/

http://localhost:8081/nexus/content/repositories/nexus-zarantech-snapshot/

you can see the url next to your created repository

Proxy repositary (this is like a replica of public repo)

For example go to

https://repo.maven.apache.org

and click on maven2 , now copy the url , you will get url like

https://repo.maven.apache.org/maven2/

now click proxy repository ,

Now lets create a group , now click add and create a group

and select

release

snapshot

proxy as below

User roles

We should enforce user according to the roles .

Now go back to the nexus home page , Under security click roles

Create role , name and id as zarantech-devops-admin

Select all from add and add all the repos

Ideally we will create role specific to the project requirement

Next we will go to settings.xml

C:\Users\administrator\Downloads\apache-maven-3.6.2-bin\apache-maven-3.6.2\conf

```
<server>
  <id>zarantech-devops-deploy</id>
  <username>zarantech-devops-deploy</username>
  <password>srikzz@1</password>
</server>
```

now ensure that this id matches inside your pom.xml , example like below ,then only your upload will work properly

```
<distributionManagement>
```

```
            <repository>
                <id>nexus-zarantech-deploy</id>
                <name>Internal Releases</name>
<url>http://localhost:8081/nexus/content/repositories/nexus-zarantech-release/</url>
            </repository>

            <snapshotRepository>
                <id>nexus-zarantech-deploy</id>
                <name>Internal Releases</name>
<url>http://localhost:8081/nexus/content/repositories/nexus-zarantech-snapshot/</url>
            </snapshotRepository>

</distributionManagement>
```

The entire working project is in the github

https://github.com/SrikanthPB/nexus

Go to the project root directory where pom.xml is located .

And type cmd in the top address bar (remove any other path , only type **cmd**)

mvn deploy

Your local repo will be uploaded , and the same be downloaded as well

I am sharing my jenkins.ppt inside softwares folder , going forward you guys can check for any material , I will place inside my C:/Softwares folder

To access

Go to start menu

Run command

\\192.168.19.108

Jenkins

Lets install jenkins

https://console.cloud.google.com/?pli=1

You can use two of the gcp accounts

There was an error while loading /home/dashboard?project=careful-trainer-150607&pli=1&authuser=2.

You are missing at least one of the following required permissions:

Project

`resourcemanager.projects.get`

Check that the project ID is valid and you have permissions to access it. Learn more

Send feedback

Please select a project from the top

Select Myfirstproject

And refresh the page

On the extreme left you have three parallel lines

Click that in the dropdown , go to compute engine and click vm instances

Create instance

Give a name like name-jenkins (use your name)

Ubuntu 16.04 LTS - select his OS

For firewall

Allow http

Allow https

Jenkins installation

sudo su

sudo apt-get update

sudo apt-get install openjdk-8-jdk

sudo wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -

**sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > **

/etc/apt/sources.list.d/jenkins.list'

sudo apt-get update

sudo apt-get install jenkins

systemctl status jenkins (this will show jenkins as active)

How to create firewall

jes

Description

Logs

Turning on firewall logs can generate a large number of logs which can increase costs in Stackdriver. Learn more

On

Off

Networkdefault

1000

Priority *

Priority can be 0 - 65535

DirectionIngress

Action on matchAllow

Specified service account

Targets

Service account scope

In this project

In another project

Compute Engine default service account

Target service account

IP ranges

Source filter

0.0.0.0/0

Source IP ranges *

None

Second source filter

Protocols and ports

Allow all

Specified protocols and ports

tcp :

8080

udp :

Other protocols

Zone

Zone is permanent

Machine configuration

Machine family

Machine types for common workloads, optimized for cost and flexibility

Series

CPU platform selection based on availability

Machine type

vCPU

1 shared core

Memory

4 GB

GPUs

-

CPU platform and GPU

Confidential VM service

Enable the Confidential Computing service on this VM instance.

Container

Deploy a container image to this VM instance. Learn more

Boot disk

New 10 GB standard persistent disk

Image

Ubuntu 18.04 LTS

Identity and API access

Service account

Access scopes

Allow default access

Allow full access to all Cloud APIs

Set access for each API

Firewall

Add tags and firewall rules to allow specific network traffic from the Internet

Allow HTTP traffic

Allow HTTPS traffic

try with http://ipaddress:8080 you can open jenkins page

Ensure that your are removing https and put only http

Inside the jenkins page

You can that it is asking you password

Go to the ssh gcp console and type

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

You will get the password

Now paste the password

Type next - select install suggested plugins

Once installed

Skip continue as admin

Save and finish

Day4

Incase if jenkins asks you username and password

Please give admin as username

And password get it from

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

In oder to do any automation inside the jenkins , please remember 3 most important things

For any tool

1) **Install plugin → manage jenkins -> manage plugins to to available plugins**

   And search for the plugin you are looking for and click install without restart

   Some times we need to download manually , we can go to the advanced tab

   And upload the hpi plugin

   https://plugins.jenkins.io/git/ (we will ffind all plugins there

   Or

   https://updates.jenkins-ci.org/download/plugins/

2) **Global tool configuration - install the tool for example we have installed ansible plugin previously , here we will select which ansible version we want to install**

3) **Configure system : where we can configure token based authentication and integrate API**

We will see some token authentication for webhook , sonar

4) **Create freestyle project : here we will give details about the particular job**

5) **Build option : like what we need to build**
6) **Post build option : if we want to notify something after the build we will sue this option ,**

For example : if you want to send email notification we will use post build notification

Lets begin the hands on exercises :

1) **Github jenkins integration**

Go to the zarantech c:/softwares/zarantech/drive please open

Here please extract audio script and demos , once you extract

Click on github jenkins integration ppt , if prompted for password

Please click read only .

You can also go through my videos for the same . trec files

https://www.videolan.org/vlc/download-windows.html

All of you please install the VLC player and you can start watching my videos for hands on excercises

Github link

https://github.com/SrikanthPB/mavenwebhook use this forgithub integration with jenkins

While installing the github plugin

Anytime if you need to restart jenkins
Use systemctl restart jenkins

Jenkins maven integration

Please take reference of jenkins maven integration ppt

and also watch video if possible

While installing java

You need to give java username password (oracle account)

Please use mine
Username : srikanth_pbs@yahoo.com
Password : Bun878reef945

Automatic builds

So far we did manual builds by clicking on build now button

We need to automate the builds from developer machine to Jenkins

What he wants to send : Github code

We need to do , Github - Jenkins -Webhook - Automated integration

Expectation : the moment developer pushes the code to github repository the build should happen

All of you take my code repo

Create folder called maven inside c drive (zarantech machine)

Go to the folder open cmd prompt and type

git init

git clone https://github.com/SrikanthPB/mavenwebhook

Use cd mavenwebhook (you need to ensure that your pom.xml is at the root folder ,so please dont forget to change directory use cd mavenwehbhook

git add -A

git commit -m “committing all the files “ -a

Now create your own repository in github.com and
Use the following commands to push to the github repo

Replace my name with your github name

\\//

```
git remote set-url origin https://github.com/SrikanthPB/mavenwebhooks.git
git branch -M main
git push -u origin main
```

Once you pushed to the repo

Since you have your own repository , you can go to settings

And click webhook

Add a webhook

Basically , we would like to create a webhook which can be hooked to jenkins

Now

We can add the payload url , something like

http://35.226.201.43:8080/github-webhook/ (here ip address and port number belong to jenkins and githubwebhook belong to your github (inside webhook you can have all the rest api methods which will help us to get information from github and send to jenkins

Now we will go to the jenkins

Manage jenkins -> Configure system : -> GitHub

Click advanced

Click on specify another url for github configuration

Please be mindful that , your page may scroll down automatically , scroll up to the same section

We have successfully integrated github with jenkins

Now you have to enable your project to use the same feature

Come to freestyle project that you created earlier , go to build triggers

Select GitHub hook trigger for GITScm polling and apply

Build notifications :

Sometimes we need to notify our team members about the build status or build failures

Install the Email Notification plugin

Rest is available in the IRC / Email notification ppt

SonarQube

Is a code review tool , which can analyse the code defects and vulnerabilities

In order to start the sonarqube

Go to the location ,

C:\Software's\zarantech drive\sonar

Here we have 3 folders which are extracted

Sonarqube-6.4 - SonarQube server

Sonar-scanner-3.0.3.778-windows - Scanner for scanning the projects

Sonar-scanning-examples-master - Sonar qube projects where we have the source code

First we need to go inside

Sonarqube-6.4

C:\sonar\sonarqube-6.4\sonarqube-6.4\bin\windows-x86-64

And type cmd in the address bar ,

And type : **StartSonar.bat**

This will bring sonarqube server up

jvm 1 | 2020.12.18 09:34:08 INFO app[][o.s.a.SchedulerImpl] Process[es] is up

(This means your sonarqube server started properly)

What we need next ?

We should be able to scan our projects using our sonar-scanner

Lets copy the location of the sonar-scanner and come back to your sonar projects

Where you have the different examples , We will run our scanner from this project

C:\sonar\sonar-scanner-3.0.3.778-windows\bin\sonar-scanner.bat

After copying the above location , go back to the place where you have the projects

I.e

C:\sonar\sonar-scanning-examples-master\sonar-scanning-examples-master\sonarqube-scanner
|||
Type cmd here |||

Please note the above location should contain my properties file

C:\sonar-scanner-3.0.3.778-windows\bin\sonar-scanner.bat

It will open 9000 port ,

Type localhost:9000 in the browser

Please note (google drive for all materials is here)

https://drive.google.com/file/d/1XCvWJEYkDJPLJDrWqdPrsbughZaBWVeH/view?usp=sharing

Login into sonarqube

Username admin
Password admin

Docker

Its a lightweight container , where we can run our applications and manage them effectively

To install docker

sudo su

apt-get install docker.io

On ubuntu

To check list of all running containers

docker ps

docker images

Jenkins pipeline

Start to end of delivery :

We will use jenkins pipeline

excercise
install docker on ubuntu lab

apt-get install docker.io

create docker hub ac
https://hub.docker.com/
docker

login give ur dockerhub
access
sudo chmod 777 /var/run/docker.sock

go to jenkins

jenkins global credentials
add docker hub credentials as following

username srikss

password Bun878reef945

id : dockerhub

manage plugins - available - docker pipeline plugin install without restart

manage plugins - available - nodejs plugin install without restart

global tool configuration - nodejs installation

name : nodejs

select any version

```
pipeline {
 environment {
  registry = 'srikss/ubuntu'
  registryCredential = 'dockerhub'
  dockerImage = ''
 }
 agent any
 tools {nodejs 'nodejs' }
 stages {
  stage('Cloning Git') {
    steps {
      git 'https://github.com/SrikanthPB/pipelinescript.git'
    }
  }
  stage('Build') {
     steps {
       sh 'npm install'
       sh 'npm run bowerInstall'
     }
  }
  stage('Test') {
    steps {
```

```
      sh 'npm test'
    }
   }
   stage('Building image') {
    steps{
     script {
      dockerImage = docker.build registry + ':$BUILD_NUMBER'
     }
    }
   }
   stage('Deploy Image') {
    steps{
      script {
        docker.withRegistry( '', registryCredential ) {
        dockerImage.push()
       }
      }
    }
   }
  }
}
```

Mandatory test all should finish this test

https://app.mymapit.in/code4/tiny/Kg1wza

Ansible

All the notes are inside latest course materials : C:/softwares
There you can find Ansible folder
Please refer my notes for more verbiage

Install ansible inside the server

Yum install ansible

Vi /etc/ansible/hosts

Give the hosts ip address

Ssh-keygen (will generate certs for secured communication with nodes)

On server

```
 1  yum update -y
   2  clear
   3  yum install ansible
   4  clear
   5  whereis ansible
   6  clear
   7  vi /etc/ansible/hosts
   8  ssh-keygen
   9  ssh-copy-id -i root@34.70.40.28
  10  ssh-copy-id -i root@34.70.40.28
  11  ssh-copy-id -i root@34.70.40.28
  12  ssh-copy-id -i root@130.211.237.185
  13  clear
  14  history
```

On node1

```
1  clear
   2  vi /etc/ssh//sshd_config
   3  systemctl restart sshd
   4  passwd
   5  history
```

Node2

```
   1  clear
   2  vi /etc/ssh/sshd_config
   3  systemctl restart sshd
   4  passwd
   5  history
```

Ansible roles

. Ansible roles are consists of many playbooks, which is similar to modules in puppet and cook books in chef. We term the same in ansible as roles.
2. Roles are a way to group multiple tasks together into one container to do the automation in very effective manner with clean directory structures.
3. Roles are set of tasks and additional files for a certain role which allow you to break up the configurations.
4. It can be easily reuse the codes by anyone if the role is suitable to someone.
5. It can be easily modify and will reduce the syntax errors.

How do we create Ansible Roles?

To create a Ansible roles, use ansible-galaxy command which has the templates to create it. This will create it under the default directory /etc/ansible/roles and do the modifications else we need to create each directories and files manually.

```
[root@learnitguide ~]# ansible-galaxy init /etc/ansible/roles/apache --offline
- apache was created successfully
[root@learnitguide ~]#
```

where, ansible-glaxy is the command to create the roles using the templates.
init is to initiliaze the role.
apache is the name of role,
offline - create offline mode rather than getting from online repository.

List out the directory created under /etc/ansible/roles.

```
[root@learnitguide ~]# tree /etc/ansible/roles/apache/
/etc/ansible/roles/apache/
|-- README.md
|-- defaults
|   `-- main.yml
|-- files
|-- handlers
|   `-- main.yml
|-- meta
|   `-- main.yml
|-- tasks
|   `-- main.yml
|-- templates
|-- tests
|   |-- inventory
|   `-- test.yml
`-- vars
    `-- main.yml
```

8 directories, 8 files
[root@learnitguide ~]#
We have got the clean directory structure with the ansible-galaxy command. Each directory must contain a main.yml file, which contains the relevant content.

Directory Structure:
tasks - contains the main list of tasks to be executed by the role.
handlers - contains handlers, which may be used by this role or even anywhere outside this role.
defaults - default variables for the role.
vars - other variables for the role. Vars has the higher priority than defaults.
files - contains files required to transfer or deployed to the target machines via this role.
templates - contains templates which can be deployed via this role.
meta - defines some data / information about this role (author, dependency, versions, examples, etc,.)

Lets take an example to create a role for Apache Web server.

Below is a sample playbook codes to deploy Apache web server. Lets convert this playbook codes into Ansible roles.

```
---
- hosts: all
  tasks:
  - name: Install httpd Package
    yum: name=httpd state=latest
  - name: Copy httpd configuration file
    copy: src=/data/httpd.original dest=/etc/httpd/conf/httpd.conf
  - name: Copy index.html file
    copy: src=/data/index.html dest=/var/www/html
    notify:
    - restart apache
  - name: Start and Enable httpd service
    service: name=httpd state=restarted enabled=yes
  handlers:
  - name: restart apache
    service: name=httpd state=restarted
```

First, move on to the Ansible roles directory and start editing the yml files.
cd /etc/ansible/roles/apache
1. Tasks

Edit main.yml available in the tasks folder to define the tasks to be executed.

```
[root@learnitguide apache]# vi tasks/main.yml
---
- name: Install httpd Package
  yum: name=httpd state=latest
- name: Copy httpd configuration file
  copy: src=/data/httpd.original dest=/etc/httpd/conf/httpd.conf
- name: Copy index.html file
  copy: src=/data/index.html dest=/var/www/html
  notify:
  - restart apache
- name: Start and Enable httpd service
  service: name=httpd state=restarted enabled=yes
```

Altogether, you can add all your tasks in this file or just break the codes even more as below using "import_tasks" statements.

```
[root@learnitguide apache]# cat tasks/main.yml
---
# tasks file for /etc/ansible/roles/apache
- import_tasks: install.yml
- import_tasks: configure.yml
- import_tasks: service.yml
```

Lets create install.yml, confgure.yml, service.yml included in the main.yml with actions in the same directory.

install.yml

```
[root@learnitguide apache]# cat tasks/install.yml
---
- name: Install httpd Package
  yum: name=httpd state=latest
```

configure.yml

```
[root@learnitguide apache]# cat tasks/configure.yml
---
- name: Copy httpd configuration file
  copy: src=files/httpd.conf dest=/etc/httpd/conf/httpd.conf
```

- name: Copy index.html file
copy: src=files/index.html dest=/var/www/html
notify:
- restart apache
service.yml
[root@learnitguide apache]# cat tasks/service.yml

- name: Start and Enable httpd service
service: name=httpd state=restarted enabled=yes
2. Files
Copy the required files (httpd.conf and index.html) to the files directory.
[root@learnitguide apache]# ll files/*
-rw-r--r-- 1 root root 11753 Feb 4 10:01 files/httpd.conf
-rw-r--r-- 1 root root 66 Feb 4 10:02 files/index.html
[root@learnitguide apache]# cat files/index.html
This is a homepage created by learnitguide.net for ansible roles.
[root@learnitguide apache]#
3. Handlers
Edit handlers main.yml to restart the server when there is a change. Because we have already defined it in the tasks with notify option. Use the same name "restart apache" within the main.yml file as below.
[root@learnitguide apache]# cat handlers/main.yml

 # handlers file for /etc/ansible/roles/apache
- name: restart apache
service: name=httpd state=restarted
4. Meta
Edit meta main.yml to add the information about the roles like author, descriptions, license, platforms supported.
[root@learnitguide apache]# cat meta/main.yml
galaxy_info:
author: LearnItGuide.net
description: Apache Webserver Role
company: LearnITGuide.net
 # If the issue tracker for your role is not on github, uncomment the
 # next line and provide a value
 # issue_tracker_url: http://example.com/issue/tracker
 # Some suggested licenses:
 # - BSD (default)
 # - MIT
 # - GPLv2
 # - GPLv3
 # - Apache
 # - CC-BY

```
  license: license (GPLv2, CC-BY, etc)
  min_ansible_version: 1.2
  # If this a Container Enabled role, provide the minimum Ansible Container version.
------skipped
List out the created files now,
[root@learnitguide apache]# tree
.
|-- README.md
|-- defaults
|   `-- main.yml
|-- files
|   |-- httpd.conf
|   `-- index.html
|-- handlers
|   `-- main.yml
|-- meta
|   `-- main.yml
|-- tasks
|   |-- configure.yml
|   |-- install.yml
|   |-- main.yml
|   `-- service.yml
|-- templates
|-- tests
|   |-- inventory
|   `-- test.yml
`-- vars
    `-- main.yml
8 directories, 13 files
[root@learnitguide apache]#
We have got all the required files for Apache roles. Lets apply this role into the ansible
playbook "runsetup.yml" as below to deploy it on the client nodes.
[root@learnitguide apache]# cat /etc/ansible/runsetup.yml
---
 - hosts: node2
   roles:
   - apache
[root@learnitguide apache]#
We have defined this changes should be run only on node2, you can also use "all" if
need. Specify the role name as "apache", also if you have created multiple roles, you can
use the below format to add it.
- apache
- nfs
- ntp
```

Lets verify for syntax errors:

```
[root@learnitguide apache]# ansible-playbook /etc/ansible/runsetup.yml --syntax-check
playbook: /etc/ansible/runsetup.yml
[root@learnitguide apache]#
```

No errors found. Let move on to deploy the roles.

```
[root@learnitguide apache]# ansible-playbook /etc/ansible/runsetup.yml
PLAY [node2]
*****************************************************************************************************
TASK [Gathering Facts]
*****************************************************************************************
ok: [node2]
TASK [apache : Install httpd Package]
*************************************************************************
changed: [node2]
TASK [apache : Copy httpd configuration file]
*******************************************************************
Changed: [node2]
TASK [apache : Copy index.html file]
*************************************************************************
changed: [node2]
TASK [apache : Start and Enable httpd service]
*******************************************************************
changed: [node2]
RUNNING HANDLER [apache : restart apache]
*********************************************************************
changed: [node2]
PLAY RECAP
*****************************************************************************************************
node2                      : ok=6    changed=5    unreachable=0    failed=0
```

That's it, We have successfully deployed the Apache webserver using Ansible Roles to the client node "node2".

Login into the client node "node2" and verify the following things.

```
[root@node2 ~]# rpm -q httpd
httpd-2.4.6-67.el7.centos.6.x86_64
[root@node2 ~]# systemctl status httpd
httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled)
   Active: active (running) since Sun 2018-02-04 10:23:44 IST; 1min 58s ago
     Docs: man:httpd(8)
           man:apachectl(8)
```

Google drive for entire batch

https://drive.google.com/u/0/uc?export=download&confirm=QUiR&id=19Rq8lzLlYEEk8FfDHg_JnSWZszc7m1kr

Feedback line for the entire batch

https://tcheck.co/nNStRb