1. Conduct the assessment using Git and git hub handle local, working and remote directories

Procedure:

Open you git bash terminal in your working local directory folder

Create Repository

$git init

$git clone

Synchronizing repositories

$git add remote origin <link>

Now for pulling the contents from remote repository perform

$git pull origin main/master

Changes to local directory

Now create the text file in your local repository and check the status in your git bash

$git status

Now add the contents to stating area by typing

$git add <createdfile.txt.>

After this commit the file by operation of commit

$git commit

Creating the branches

$git branch new_branch

$git new_branch checkout

Now in your local directory create few more file and edit the file of the earlier created

$git status

$git add

Again checkout to master

$git checkout master

Edit the file in your local repo

It should not list the files of new branch

Check by typing

$git checkout new_branch

$ls

Push your local directory files to remote repository

$git push origin master/main

All the files need to be popped in your remote github repository

5. Continuous build integration and deployment of client-server method through maven project

Prerequisite : Jenkins.war, java 8, mobaxtrem client app, aws, Tomcat server setup by ec2, gi thub

Setting up a Maven Project

1. Open Jenkins and create new item with maven option select ok
2. From your github repo copy the https url and paste it in Jenkins job created by selecting git
3. Under build you will be directed with pom.xml and provide clean install package
4. Go to build now and console wil be success
5. Now open your AWS and copy the setup servers public IP to mobaxterm client
6. Remote hoste: Public IP (AWS) ans username :ec2-user
   Advanced settings: select the key generated
7. After the ec2 instance is enabled at your terminal, switch to root ($ sudo su -)
   $ cd /opt
   $ cd apache folder
   $ls
   $cd bin
   $./startup.sh
8. Access the browser with IP publicip:8080 -→ this leads to tomcat server page . Note: only if you have configured the server
9. You can change the port number of tomcat server to 8090 by selecting
   $cd conf
   $vi server.xml -> here change the 8080 number to 8090
10. Got to Jenkins and your project configuration , in post steps no war file will be found to enable it got to step 11
11. Now go back to your jenkins and install deploy to container plugin
12. Got to manage Jenkins and mange credentials, now add credentials , credentials can be found at client, Type the command as
    $ cat tomcatusers.xml
13. Now in Jenkins credentials add the contents both password and username as deployer and description as tomcat_credentials even id is same as description
14. Got to step 10 and repeat the procedure to find war content
15. Now war is found and click on it and input the following
16. In add container select tomcat 8

And save the content

17. Now go to client app and type in this
    $ cd webpp (under apache directory)
    $ ls
    $ls –ltr
    Now you will be able to vie the new file

2. Enabling Tomcat application manager from client IP through continuous deployment

Pre-requisites

1. EC2 instance with Java v1.8.x

Install Apache Tomcat

1. Download tomcat packages from https://tomcat.apache.org/download-80.cgi onto /opt on EC2 instance

   Note: Make sure you change <version> with the tomcat version which you download.
   # Create tomcat directory
   $cd /opt
   $wget http://mirrors.fibergrid.in/apache/tomcat/tomcat-8/v8.5.35/bin/apache-tomcat-8.5.35.tar.gz
   $tar -xvzf /opt/apache-tomcat-<version>.tar.gz

2. give executing permissions to startup.sh and shutdown.sh which are under bin.
3. $chmod +x /opt/apache-tomcat-<version>/bin/startup.sh
   $chmod +x /opt/apache-tomcat-<version>/bin/shutdown.sh

   Note: you may get below error while starting tomcat incase if you don't install Java
   Neither the JAVA_HOME nor the JRE_HOME environment variable is defined At least one of these environment variable is needed to run this program

4. create link files for tomcat startup.sh and shutdown.sh
5. $ln -s /opt/apache-tomcat-<version>/bin/startup.sh /usr/local/bin/tomcatup
6. $ln -s /opt/apache-tomcat-<version>/bin/shutdown.sh /usr/local/bin/tomcatdown
   tomcatup

Check point :

access tomcat application from browser on port 8080

- http://<Public_IP>:8080

Using unique ports for each application is a best practice in an environment. But tomcat and Jenkins runs on ports number 8080. Hence lets change tomcat port number to 8090. Change port number in conf/server.xml file under tomcat home

$cd /opt/apache-tomcat-<version>/conf
update port number in the "connecter port" field in server.xml
restart tomcat after configuration update
$tomcatdown
$tomcatup

Check point :

Access tomcat application from browser on port 8090

- http://<Public_IP>:8090

1. now application is accessible on port 8090. but tomcat application doesnt allow to login from browser. changing a default parameter in context.xml does address this issue
2. #search for context.xml
   $find / -name context.xml

3. above command gives 3 context.xml files. comment () Value ClassName field on files which are under webapp directory. After that restart tomcat services to effect these changes. At the time of writing this lecture below 2 files are updated.
4. $/opt/tomcat/webapps/host-manager/META-INF/context.xml
5. $/opt/tomcat/webapps/manager/META-INF/context.xml
6.
7. # Restart tomcat services
8. $tomcatdown
   $tomcatup

9. Update users information in the tomcat-users.xml file goto tomcat home directory and Add below users to conf/tomcat-users.xml file
10. <role rolename="manager-gui"/>
11. <role rolename="manager-script"/>
12. <role rolename="manager-jmx"/>
13. <role rolename="manager-status"/>
14. <user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>
15. <user username="deployer" password="deployer" roles="manager-script"/>
    <user username="tomcat" password="s3cret" roles="manager-gui"/>

16. Restart service and try to login to tomcat application from the browser. This time it should be Successful

3) Enabling Docker image on client terminal through Jenkins deployment

1. Launch an EC2 instance for Docker host
2. Install docker on EC2 instance and start services

```
$yum install docker
$service docker start
```

3. create a new user for Docker management and add him to Docker (default) group

```
$useradd dockeradmin
$passwd dockeradmin
$usermod -aG docker dockeradmin
```

4. Write a Docker file under /opt/docker

```
$mkdir /opt/docker
### vi Dockerfile
# Pull base image
From tomcat:8-jre8
# Maintainer
MAINTAINER "ISE"
# copy war file on to container
COPY ./webapp.war /usr/local/tomcat/webapps
```

5. Login to Jenkins console and add Docker server to execute commands from Jenkins
   Manage Jenkins --> Configure system --> Publish over SSH --> add Docker server and credentials
6. Create Jenkins job

A) Source Code Management
Repository : url from git hub
Branches to build : */master

B) Build Root POM: pom.xml
Goals and options : clean install package

C) send files or execute commands over SSH Name: docker_host
Source files : webapp/target/*.war Remove prefix : webapp/target Remote directory : //opt//docker
Exec command[s] :

```
docker stop runshaw_demo;
docker rm -f runshaw_demo;
docker image rm -f valaxy_demo;
cd /opt/docker;
docker build -t valaxy_demo .
```

D) send files or execute commands over SSH
Name: docker_host
Exec command : docker run -d --name valaxy_demo -p 8090:8080 runshaw_demo

7. Login to Docker host and check images and containers. (no images and containers)
8. Execute Jenkins job
9. check images and containers again on Docker host. This time an image and container get creates through Jenkins job
10. Access web application from browser which is running on container

<docker_host_Public_IP>:8090

4. Enabling Ansible server IP for duplication with client for webapp deployment

First step up is
Launching of two ec2 instances one consisting of ansible host and control server respectively
Note: for security group you need to enable only ssh 22
And launch the two instances
Open client mobaxterm SSH
Copy public IP of host and control server ansible from AWS that have recently launced

In Control Server follow this procedure

$sudo su –
$ yum update

In ansible client

$sudo su –
$ yum update

In Control Server follow this procedure

$ yum install ansible
$ rpm –Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
$ yum install ansible
$ ansible --version
$ useradd ansadmin
$ passwd ansadmin
Provide the password of your own in the password section
$ yum install ansible
$ rpm –Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
$ yum install ansible
$ ansible –version

In ansible client

$ useradd ansadmin
$ passwd ansadmin
Provide the password of your own in the password section

In Control Server

$ visudo
Type in the following instruction at last of the editor

In ansible client

$ visudo
Type in the following instruction at last of the editor

In Control Server

$ cd /etc/ssh
$ ls
$ vi sshd_config
Type in the following by enabling the password authentication has **yes**
$ service sshd restart

In ansible client

$ cd /etc/ssh
$ ls
$ vi sshd_config
Type in the following by enabling the password authentication has **yes**
$ service sshd restart

Now right click on the ansible control server tab in your client click on duplicate tab. This will open a duplicate tab and features the replication of ansible server

In Duplicate control server tab (here you need to generate ssh)

$ sudo su ansadmin
$ssh-keygen
Press enter don't type password
$ls –la

There should be file by name .ssh or else error
Go to ansible client
$ ifconfig
This gives you the private ip of your ansible client and copy this IP
In Duplicate control server tab
$ cd .ssh
$ ls
$ ssh-copy-id **<private ip of ansible client> //this is how you deploy several clients in to server control area dynamically//**
Below snap shot result with copy without authentication
Note: if it asks the password than its an error
To come out of client ip
$ ssh <client private ip>
$ exit
Connection will be closed
To execute and deploy features of ansible configuration in devops strategy , in duplicate server follow the procedure
$ sudo vi /etc/ansible/hosts
Delete the content and type the following this way like snapshot
For reference we a have taken an example IP , you go to type your respective host/client IP
$ ansible all –m ping
$ls
$ cd ..
$ls
$pwd
$cat > hello.html
<h1> hello welcome .. welcome to devops </h1>
$ls
Displays hello.html file

In your ansible client

$cd /home/ansadmin

$ls

$pwd

$ls

Go back to your control server

$ ansible all –m copy "src=/home/amsadmin/hello.html dest=/home/ansadmin"

In your client

$ls -l

This is the final output