Module1: Jenkins Administration – Installation of Jenkins,Java and Git ,Maven in Redhat Linux machine

**Hands-on Exercise Objective**

After completing the hands-on exercises, you will be able to completely understand and obtain thorough working knowledge on **Installation** module in Linux and Windows machine and its features like Git, Java and Maven.

**Exercise 1**

Problem Statement:

* **Creating installation script file**
* **Script will be executed and install the Jenkins, Git, Maven and Tomcat in Linux machine.**
* **Configuring Tomcat to install the Jenkins in port 8080.**

Deliverables Expected:

Git, Maven, Java and Tomcat. Jenkins Global tool configuration.

Step by instructions to install the Jenkins, Git, Maven and Tomcat:

**Scenario 1:** All the tools instructions given in the Tools\_installation\_guide document. Just copy all the contents and paste it in your VM machine. All the tools will be installed in your VM.



**Scenario 2**: Configuration of Tomcat and Jenkins plugin installation steps. Please check the below document and follow the instructions accordingly.



**Scenario 3**:

* Installation of Jenkins in windows machine
* Place the war file in apchetomcat webapps and execute the windows batch file. Access the Jenkins URL

http://localhost:8080/jenkins or http://ec2-13-127-234-110.ap-south-1.compute.amazonaws.com:8080/jenkins



Module2: Jenkins Administration – Role based authentication

**Hands-on Exercise Objective**

After completing the hands-on exercises, you will be able to completely understand and obtain thorough working knowledge on **Role** **Authorization** module in Jenkins and its features like users authorization and Role creation.

**Exercise 2**

Problem Statement:

* **Creating users in Jenkins and assign the project to users**
* **Role based authentication plugin to be installed in Jenkins.**
* **Creating roles and assign to users.**

**Deliverables Expected**:

Login through different users and check if authorization is working fine.

**Step by instructions to achieve the role based authorization**:



**Hardware / Software requirement: Jenkins**

Module 3: Configuration of jobs – Git configuration

After completing the hands-on exercises, you will be able to completely understand and obtain thorough working knowledge on Git configuration and its features like executing jobs and project configuration.

**Exercise 3**

**Problem Statement**:

* + - **To connect with the Git repository for a particular given project**
    - **To execute the job and verify the outputs**

**Deliverables Expected**: Code would be pulled from the GitHub repository and placed in Jenkins workspace directory.

Stepwise Instruction on how to complete this exercise



Module 4: Configuration of jobs – Code build configuration

After completing the hands-on exercises, you will be able to completely understand and obtain thorough working knowledge on build tool configuration in Jenkins and its features like Maven and its commands so on.

**Exercise 4**

**Problem Statement**:

* + - **Configure the Build tool**
    - **Choose the build option to compile the code**
    - **Execute the jobs and check the build status.**

**Deliverables Expected**: Complied code would be packaged into the target server.

Stepwise Instruction on how to complete this exercise should be given in the below document.

****

Module 5: Configuration of jobs - Deployment in Web server

After completing the hands-on exercises, you will be able to completely understand and obtain thorough working knowledge on deployment and its features like Tomcat and it is functionalities access through web portal.

**Exercise 5**

**Problem Statement**:

* + - **Create a new job and configure the code deployment**
    - **Check if the code is deployed into the web server**
    - **Access the web server through web portal and check if the code exists**

**Deliverables Expected**: Code deployment in Tomcat web server.

Stepwise Instruction on how to complete this exercise should be given for this



Module 6: Configuration of jobs – Build with parameters

After completing the hands-on exercises, you will be able to completely understand and obtain thorough working knowledge on **Runtime features with Build** and its features like Run time parameters to select the build options and run the job appropriately.

**Exercise 6**

**Problem Statement**: Achieve the build options to be set at run time and execute the job.

* + - **Create a new job and configure build options**
    - **Run time parameters to be set and select the build options at run time**
    - **Choose the build option and see the output log**

**Deliverables Expected**: Build will be executed based on our choice when we select it in run time and output will be delivered.

Stepwise Instruction on how to complete this exercise given below:



****

Module 7: Configuration of jobs – Automated Build configuration

After completing the hands-on exercises, you will be able to completely understand and obtain thorough working knowledge on **Automated Build** and its features like scheduled build in GitHub and Jenkins.

**Exercise 7**

**Problem Statement**: Achieve the auto build options in Github and trigger through job.

* + - **Create a new job and configure build options**
    - **Setup a auto trigger build in GitHub**
    - **Job will be configured in Jenkins to trigger it automatically from GitHub repository whenever any commit happens.**

**Deliverables Expected**: Auto build- whenever any commits happened in the GitHub repository.

**Stepwise Instruction on how to complete this exercise given below**:

Scenario 1: Complete the setting as per the below document.



Sceanrio2: Detail steps available in the below document how to setup a Github settings and Jenkins job :



Module 8: Configuration of jobs –Triggering Build using remotely

After completing the hands-on exercises, you will be able to completely understand and obtain thorough working knowledge on **Automated Build** and its features like scheduled build in GitHub and Jenkins.

**Exercise 8**

**Problem Statement**: Achieve the auto build trigger option through authorized token option.

* + - **Create a new job and configure authentication token options**
    - **Setup a auto trigger build**
    - **To schedule a nightly build**

**Deliverables Expected:**

* Once script file executed, jenkin job needs to be automatically triggered.
* Job needs to be triggered at the specific time



Module 9: Configuration of jobs – Delivery pipeline workflow

After completing the hands-on exercises, you will be able to completely understand and obtain thorough working knowledge on **pipeline workflow in Jenkins** and its features like delivery pipeline and dependency jobs would be triggered.

**Exercise 9**

**Problem Statement**: To create a delivery pipeline and triggered the all the required jobs.

* + To create a pipeline workflow in Jenkins
  + Jobs will be created to sync with the workflow structure
  + To install a Delivery pipeline plugin in Jenkins
  + To configure the Delivery pipeline plugin and assign the jobs to be triggered
  + Pipeline view to be created after the configuration made in the Delivery pipeline
  + Trigger the Delivery pipeline and see all the dependency jobs are triggered

**Deliverables Expected**: Delivery pipeline view will be shown in the screen for the all the jobs are tagged with the pipeline.

Stepwise Instruction on how to complete this exercise should be given for this



Module 10: Jenkins Administration – Jenkins master and slave configuration in windows and Linux machine

After completing the hands-on exercises, you will be able to completely understand and obtain thorough working knowledge on **Jenkins slave configuration** module in Linux and Windows machine .

**Exercise 10**

Problem Statement:

* **Creating a Linux and Windows machine**
* **Using Java web start to connect the master machine**
* **Configuring job in Jenkins to run the job in Slave machine.**

Deliverables Expected:

Jobs running in slave machine.

Step by instructions to achieve the role based authorization:



Module 11: Configuration of jobs – Code analysis

After completing the hands-on exercises, you will be able to completely understand and obtain thorough working knowledge on **code analysis** module through Jenkins job.

**Exercise 11**

Problem Statement:

* **Creating a freestyle job to setup code analysis module**
* **To install the code analysis plugin (Install checkstyle plugin)**
* **Execute the code analysis job to see the output of code warnings and graph**

Deliverables Expected:

Code analysis of project output delivered.

Step by instructions to achieve the code analysis report:



Module 12: Configuration of jobs – Continuous delivery deployment using build pipeline

After completing the hands-on exercises, you will be able to completely understand and obtain thorough working knowledge on **continuous delivery pipeline** module through build pipeline.

**Exercise 12**

Problem Statement:

* **Creating a set of jobs to configure in Jenkins**
* **To install required plugins in Jenkins to work with the build pipeline (Copy artifacts, deploy to container, checkstyle and build pipeline plugins)**
* **Execute the code job to see the delivery pipeline output in the screen**

Deliverables Expected:

Build output will be packaged and copy into the tomcat servers.

Step by instructions to achieve the code analysis report:



Module 13: Configuration of jobs –Deployment using deploy to war container plugin

After completing the hands-on exercises, you will be able to completely understand and obtain thorough working knowledge on automated deployment in Jenkins.

Exercise 13:

**Problem Statement: Automated deployment in Jenkins**

* **Create a freestyle project, configure Git and maven.**
* **Install Deploy plugin to deploy the war in apache tomcat**
* **Post build step : Add war/ear to container**

**Deliverables Expected**: Get the code from Git and delete the existing target directory and build the new war and place in the target directory using maven and deploy the war in apache tomcat. Access the developed application.

**Stepwise Instruction on how to complete this exercise should be given for this**



Module 14: Configuration of jobs – Notify the build failure in email

After completing the hands-on exercises, you will be able to completely understand and obtain thorough working knowledge on notification of build failure through email.

**Exercise 14:**

**Problem Statement: Email notifications**

* **Notify on the build failure in email.**

**Deliverables Expected**: Need to send an email once build failed occurs.

**Stepwise Instruction on how to complete this exercise should be given for this**

****

Module 15: Configuration of job – Jenkins as a code

After completing the hands-on exercises, you will be able to completely understand and obtain thorough working knowledge on groove script and setup a multiple job though code.

**Exercise 15:**

**Problem Statement: Script usage in Jenkins**

* **Use Jenkins as a code**
* **Groove script will be used to configure the pipeline**

**Deliverables Expected**: Multi pipeline jobs would be triggered.

.

**Stepwise Instruction on how to complete this exercise and follow the below:**

****