

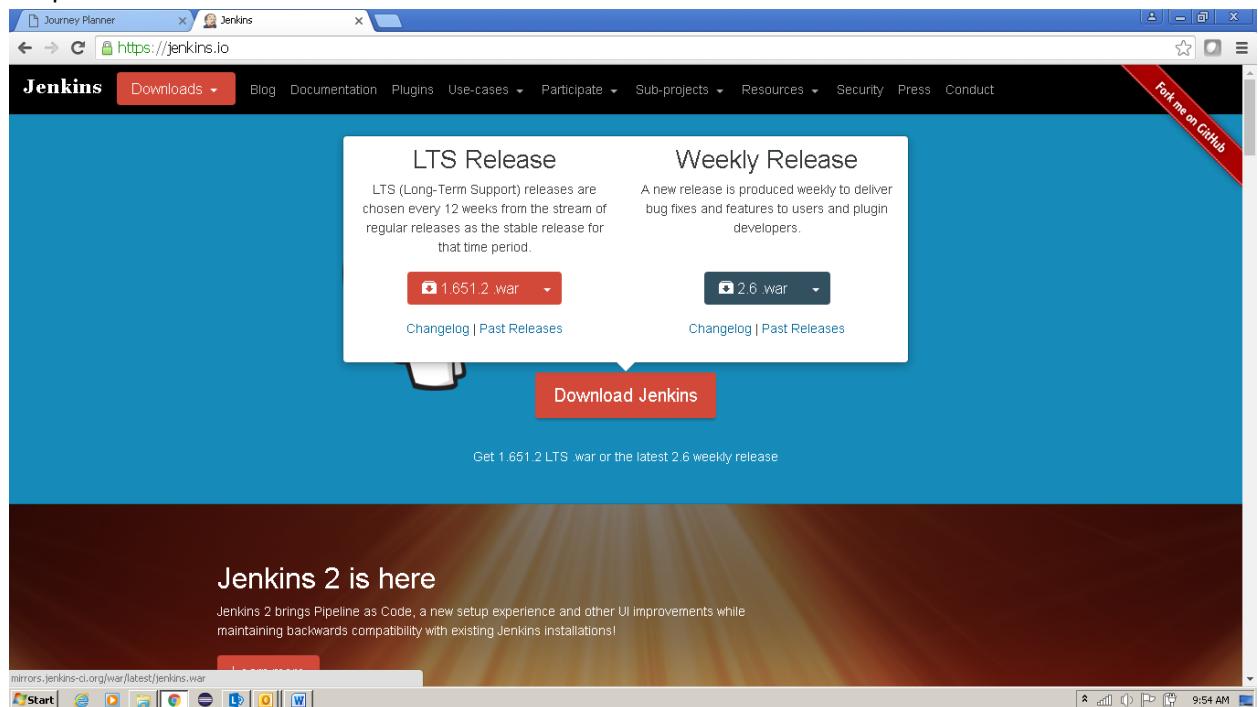
1.	Jenkins war download	2
2.	Apache download	3
3.	Running Jenkins war	4
3.1.	Starting tomcat server with jenkins-	4
3.2.	Using admin initial password-.....	5
3.3.	close the getting started page. On next screen-	5
3.4.	Following screen will be displayed. Click on Start Using Jenkins-	6
3.5.	Following home page will be displayed. This means that Jenkins is running now.....	6
4.	Installing required plugins-	7
4.1.	Git plugin.....	7
4.2.	Gradle plugin.....	10
4.3.	Multijob plugin.....	11
4.4.	Configuring JDK and Git-	12
4.5.	Installing Jmeter Html Report Plugin-.....	13
5.	Installing Docker	14
6.	Installing Amazon CLI.....	14
7.	Setting up credentials for Amazon ECR	15
8.	Job creation.....	17
8.1.	Job to create and start auth container.....	17
8.2.	Job to create and start mail container	21
8.3.	Job to run integration test-.....	22
8.4.	Job to stop and remove auth container	24
8.5.	Job to stop and remove mail container	26
8.6.	Job to buildImage and upload to ecr repository	27
8.7.	Job to Run Jmeter Job.....	31
8.8.	Master Job to call child jobs –	33
9.	Running Job.....	36
9.1.	MasterJob-	36
9.2.	Build and upload image to ECR-	38

1. Jenkins war download

Get Latest War of Jenkins by following link (I downloaded 2.6.war)-

<https://jenkins.io/>

Snapshot below-



2. Apache download

Download Apache container zip to run the war from following link-

<https://tomcat.apache.org/download-80.cgi>

The screenshot shows a Windows desktop with a taskbar at the bottom. The taskbar icons include Start, Internet Explorer, File Explorer, Task View, Taskbar settings, and system status. A browser window is open to the URL <https://tomcat.apache.org/download-80.cgi>. The page content is as follows:

which contains the OpenJDK keys or Tomcat's release managers. We also provide MD5 and SHA-1 checksums for every release file. After you download the file, you should calculate a checksum for your download, and make sure it is the same as ours.

Mirrors

You are currently using <http://mirror.fibergrid.in/apache/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are [backup mirrors](#) (at the end of the mirrors list) that should be available.

Other mirrors: <http://mirror.fibergrid.in/apache/> ▾ Change

8.0.35

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [tar.gz \(pgp, md5, sha1\)](#)
 - [tar.gz \(pgp, md5, sha1\)](#)
 - [32-bit Windows zip \(pgp, md5, sha1\)](#)
 - [64-bit Windows zip \(pgp, md5, sha1\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, md5, sha1\)](#)
- Full documentation:
 - [tar.gz \(pgp, md5, sha1\)](#)
- Deployer:
 - [zip \(pgp, md5, sha1\)](#)
 - [tar.gz \(pgp, md5, sha1\)](#)
- Extras:
 - [JMX Remote Jar \(pgp, md5, sha1\)](#)
 - [Web services Jar \(pgp, md5, sha1\)](#)
 - [JULI adapters Jar \(pgp, md5, sha1\)](#)
 - [JULI log4j Jar \(pgp, md5, sha1\)](#)
- Embedded:
 - [tar.gz \(pgp, md5, sha1\)](#)
 - [zip \(pgp, md5, sha1\)](#)

3. Running Jenkins war

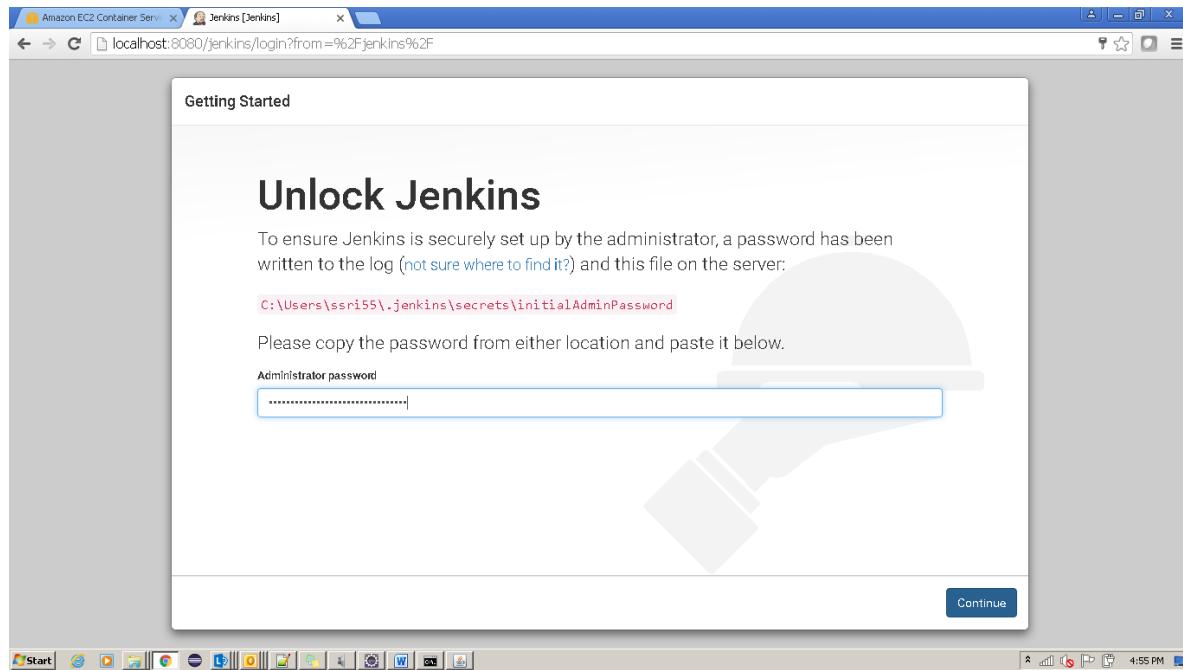
3.1.Starting tomcat server with jenkins-

Extract the tomcat container. Put the Jenkins.war in webapps folder of tomcat (apache-tomcat-8.0.35\webapps).Now go to bin folder of tomcat and start the tomcat container by running – **startup.bat**

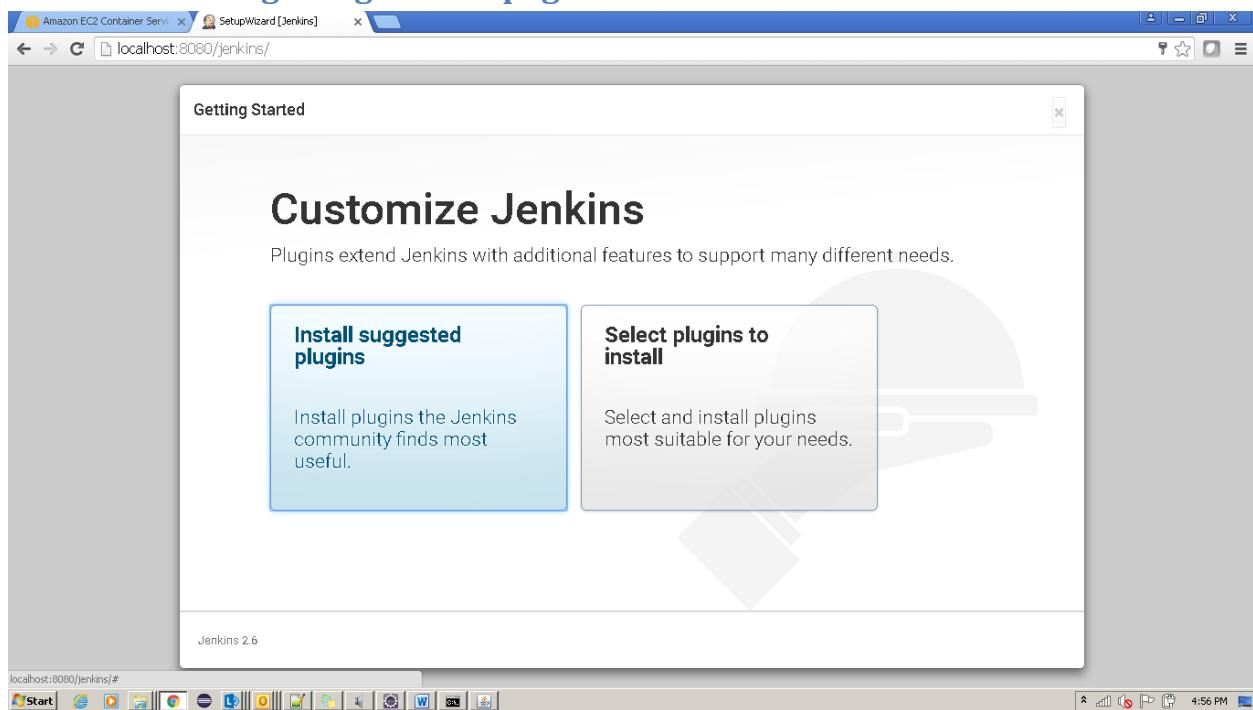
When you start Jenkins it will also need JAVA_HOME set up on the system. Following screen will be displayed when server starts. Copy the password printed in console and highlighted in below image. It will be required later.

3.2. Using admin initial password-

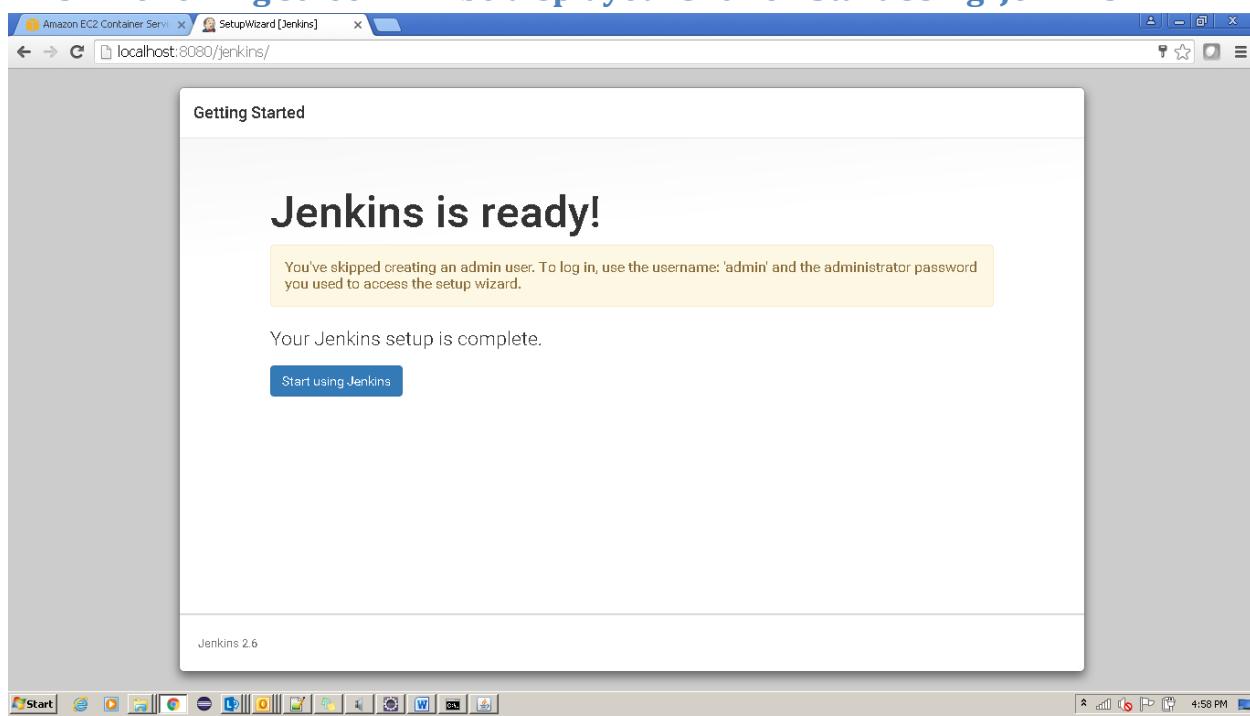
Now hit the url – <http://localhost:8080/jenkins/> It will display below screen. Enter the password copied earlier in this page and click Continue-



3.3. close the getting started page. On next screen-



3.4.Following screen will be displayed. Click on Start Using Jenkins-



3.5.Following home page will be displayed. This means that Jenkins is running now.

A screenshot of the Jenkins dashboard home page. The title bar says "Dashboard [Jenkins]". The main header features the Jenkins logo and the text "Welcome to Jenkins!". Below this, a message says "Please [create new jobs](#) to get started." On the left sidebar, there are links for "New Item", "People", "Build History", "Manage Jenkins", and "My Views". Under "Manage Jenkins", there is a "Build Queue" section with the message "No builds in the queue." and a "Build Executor Status" section showing "1 Idle" and "2 Idle". The top right corner shows the user "admin" and a "log out" link. The bottom right corner shows the page was generated on "26-May-2016 16:59:04 IST" and "Jenkins ver. 2.6". The browser's address bar shows "localhost:8080/jenkins/". The taskbar at the bottom includes icons for Start, Internet Explorer, File Explorer, and others.

4. Installing required plugins-

4.1. Git plugin

Hit Url - <http://localhost:8080/jenkins> and enter username - admin and password- the initial password setup earlier (*If not set otherwise this will be in file- C:\Users\<yourusername>\.jenkins\secrets\initialAdminPassword*)



Open manage Jenkins-

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links for New Item, People, Build History, Manage Jenkins (which is selected), and My Views. Below the sidebar are two collapsed sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). The main content area is titled 'Manage Jenkins' and lists several management options with icons:

- Configure System: Configure global settings and paths.
- Configure Global Security: Secure Jenkins; define who is allowed to access/use the system.
- Global Tool Configuration: Configure tools, their locations and automatic installers.
- Reload Configuration from Disk: Reload all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.
- Manage Plugins: Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- System Information: Displays various environmental information to assist trouble-shooting.
- System Log: System log captures output from `java.util.logging` related to Jenkins.
- Load Statistics: Check your resource utilization and see if you need more computers for your builds.
- Jenkins CLI: Access/manage Jenkins from your shell, or from your script.
- Script Console: Executes arbitrary script for administration/trouble-shooting/diagnostics.

At the bottom, there's a 'Manage Nodes' link. The status bar at the bottom right shows 'Waiting for localhost...', '5:05 PM', and other system icons.

Click on Manage Plugins. And click on Available Tab

The screenshot shows the Jenkins Update Center [Jenkins] page, specifically the 'Available' tab. The top navigation bar includes 'Back to Dashboard', 'Manage Jenkins', and 'Update Center'. The 'Available' tab is selected, indicated by a blue border. Other tabs include 'Updates', 'Installed', and 'Advanced'. A 'Filter' input field is present. The main content area displays a table of available plugins:

Install	Name	Version
CCM Plug-in	This plug-in generates reports on cyclomatic complexity for .NET code.	3.1
Change Assembly Version	This plugin change the <code>\AssemblyVersion</code> and <code>\AssemblyFileVersion</code> from AssemblyInfo.cs sources.	1.5.1
FxCop Runner plugin	<code>FxCopCmd.exe</code> execute plugin.	1.1
MSBuild Plugin	This plugin allows you to use MSBuild to build .NET projects.	1.25
MSTest plugin	This plugin converts MSTest TRX test reports into JUnit XML reports so it can be integrated with Jenkins's JUnit features. This plugin converts the coveragexml files found in the project workspace to the EMMA format. You can use <code>MSTestRunner plugin</code> or <code>VsTestRunner plugin</code> to run the test and use this plugin to process the results.	0.19
MSTestRunner plugin	This plugin allow you to execute test using MSTest command line tool.	1.2.0
NAnt Plugin	This plugin allows for the execution of a <code>NAnt</code> build as a build step.	1.4.3
NCover plugin	Archive and publish .NET code coverage HTML reports from <code>NCover</code> .	0.3
PowerShell plugin		1.3

The status bar at the bottom right shows 'Waiting for localhost...', '5:06 PM', and other system icons.

Search for GIT. Select the plugin highlighted below-

The screenshot shows the Jenkins Plugin Manager interface. The top navigation bar includes links for Back to Dashboard, Manage Jenkins, and Update Center. The main content area has tabs for Available, Installed, and Advanced, with Available selected. A search bar at the top right is set to 'GIT'. Below the tabs is a table with columns for Name, Version, and a brief description. The table lists several GitHub-related plugins:

Install	Name	Version
[checkbox]	GitHub Authentication plugin	0.24
[checkbox]	Gitcolony Build Notification plugin	1.1
[checkbox]	Ontrack Jenkins plug-in	2.19.2
[checkbox]	AsakusaSatellite Plugin	0.1.1
[checkbox]	VS Team Services Continuous Deployment	1.2
[checkbox]	Git Parameter Plug-in	0.5.1
[checkbox]	Stash Branch Parameter Plug-in	0.2.0
[checkbox]	bootstrapped-multi-test-results-report	1.4.12

At the bottom are three buttons: 'Install without restart', 'Download now and install after restart', and 'Check now'.

This screenshot is similar to the first one, showing the Jenkins Plugin Manager. The 'Available' tab is selected, and the 'GIT' search term is still present. The table lists various GitHub-related plugins, with the 'Git plugin' entry highlighted by a red rectangle. The 'Git plugin' row contains the following information:

[checkbox]	Git plugin	2.4.4
------------	------------	-------

The 'Git plugin' description states: "This plugin allows use of [Git](#) as a build SCM, including repository browsers for several providers. A recent Git runtime is required (1.7.9 minimum, 1.8.x recommended). Interaction with the Git runtime is performed by the use of the [Git Client Plugin](#), which is only tested on official [git client](#). Use exotic installations at your own risk."

At the bottom are three buttons: 'Install without restart', 'Download now and install after restart', and 'Check now'.

Following screen will be displayed-

Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Plugin	Status
Credentials Plugin	Success
JUnit Plugin	Success
Script Security Plugin	Success
Matrix Project Plugin	Success
Windows Slaves Plugin	Success
SSH Credentials Plugin	Installing
Icon Shim Plugin	Pending
Matrix Authorization Strategy Plugin	Pending
OWASP Markup Formatter Plugin	Pending
Git client plugin	Pending
SCM API Plugin	Pending
Mailer Plugin	Pending
Git plugin	Pending

[Go back to the top page](#)
(you can start using the installed plugins right away)

Once All components are success go to Manage Plugins again to search other plugin.

4.2.Gradle plugin

Go to plugin manager again and search for gradle plugin and click on install it without restart option-

Plugin Manager

Filter:

Install	Name	Version
<input checked="" type="checkbox"/> Gradle plugin	This plugin makes it possible to invoke a Gradle build script as the main build step.	1.24
<input type="checkbox"/> Artifactory Plugin	This plugin allows deploying Maven 2, Maven 3, Ivy and Gradle artifacts and build info to the Artifactory artifacts manager.	2.4.7

[Install without restart](#) [Download now and install after restart](#) Update information obtained: 19 hr ago [Check now](#)

Once the success is shown as below-

Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Gradle plugin Success

[Go back to the top page](#)
(you can start using the installed plugins right away)

[Restart Jenkins when installation is complete and no jobs are running](#)

4.3. Multijob plugin

Now go to manage plugins again. And in available tab search for ***jenkins-multijob-plugin***, Select the plugin as shown below and click install without restart–

Filter:

Install ↓	Name	Version
<input checked="" type="checkbox"/> Multijob plugin	This plugin, created by Tikal ALM team, gives the option to define complex and hierarchical jobs structure in Jenkins.	1.21

Install without restart **Download now and install after restart** Update information obtained: 21 hr ago **Check now**

Now Once the installation is complete, the below screen will be displayed-

The screenshot shows the Jenkins Update Center page at localhost:8080/jenkins/updateCenter/. The title is "Installing Plugins/Upgrades". The left sidebar has links for Back to Dashboard, Manage Jenkins, and Manage Plugins. The main content area shows a table of installed plugins with status indicators:

Plugin	Status
Gradle plugin	Success
Javadoc Plugin	Success
Maven Integration plugin	Success
Token Macro Plugin	Success
PAM Authentication plugin	Success
LDAP Plugin	Success
Ant Plugin	Success
External Monitor Job Type Plugin	Success
Run Condition Plugin	Success
conditional-buildstep	Success
Environment Injector Plugin	Success
Parameterized Trigger plugin	Success
Multijob plugin	Success

At the bottom, there is a link to "Go back to the top page" with the note "(you can start using the installed plugins right away)".

4.4. Configuring JDK and Git-

Go to manage Jenkins and Click on Configure Global Tool configuration-

The screenshot shows the Jenkins Manage Jenkins page at localhost:8080/jenkins/manage. The title is "Manage Jenkins". The left sidebar has links for New Item, People, Build History, Manage Jenkins (selected), My Views, and Credentials. The main content area shows the "Global Tool Configuration" section with various tools listed:

- Configure System
- Configure Global Security
- Configure Credentials
- Global Tool Configuration (selected)
- Reload Configuration from Disk
- Manage Plugins
- System Information
- System Log
- Load Statistics
- Jenkins CLI

Click on add JDK and set values for JDK and Git as below and save-

The screenshot shows the Jenkins Global Tool Configuration page. It has two main sections: 'JDK' and 'Git'.

JDK section:

- JDK installations:**
 - Name:** java_1_8
 - JAVA_HOME:** C:\Program Files\Java\jdk1.8.0_73
 - Install automatically
- Add JDK** button
- Delete JDK** button

Git section:

- Git installations:**
 - Name:** Default
 - Path to Git executable:** C:\Program Files\Git\bin\git.exe
 - Install automatically
- Add Git** dropdown menu
- Delete Git** button

At the bottom are 'Save' and 'Apply' buttons.

4.5. Installing Jmeter Html Report Plugin-

Install Plugin as shown in the Figure below.

The screenshot shows the Jenkins Plugin Manager page, listing installed plugins. The 'HTML Publisher plugin' is highlighted.

Plugin Name	Version	Action
Git server plugin	1.6	Uninstall
GitHub API Plugin	1.75	Uninstall
GitHub Branch Source Plugin	1.7	Uninstall
GitHub Organization Folder Plugin	1.3	Uninstall
GitHub plugin	1.19.1	Uninstall
Gradle plugin	1.24	Uninstall
HTML Publisher plugin	1.11	Uninstall
Icon Shim Plugin	2.0.3	Uninstall
Javadoc Plugin	1.4	Uninstall
JavaScript GUI Lib: ACE Editor bundle plugin	1.1	Uninstall
JavaScript GUI Lib: Handlebars bundle plugin	1.1.1	Uninstall
JavaScript GUI Lib: jQuery bundles (jQuery and jQuery UI) plugin	1.2.1	Uninstall
JavaScript GUI Lib: Moment.js bundle plugin	1.1.1	Uninstall
JUnit Plugin	1.13	Uninstall
LDAP Plugin	1.12	Uninstall

5. Installing Docker

Install docker as explained in the link below-

<https://docs.docker.com/windows/>

Get Started with Docker for Windows

This is written for users of Windows. If you are not using Windows, see the [Linux](#) or [Mac OS X](#) version.

This getting started is for non-technical users who are interested in learning about Docker. By following this getting started, you'll learn fundamental Docker features by performing some simple tasks. You'll learn how to:

- install Docker software using the Docker Toolbox
- use Docker Engine to run a software image in a container
- browse for an image on Docker Hub
- create your own image and run it in a container
- create a Docker Hub account and an image repository
- create an image of your own
- push your image to Docker Hub for others to use

The getting started was user tested to reduce the chance of users having problems. For the best chance of success, follow the steps as written the first time before exploring on your own. It takes approximately 45 minutes to complete.

6. Installing Amazon CLI

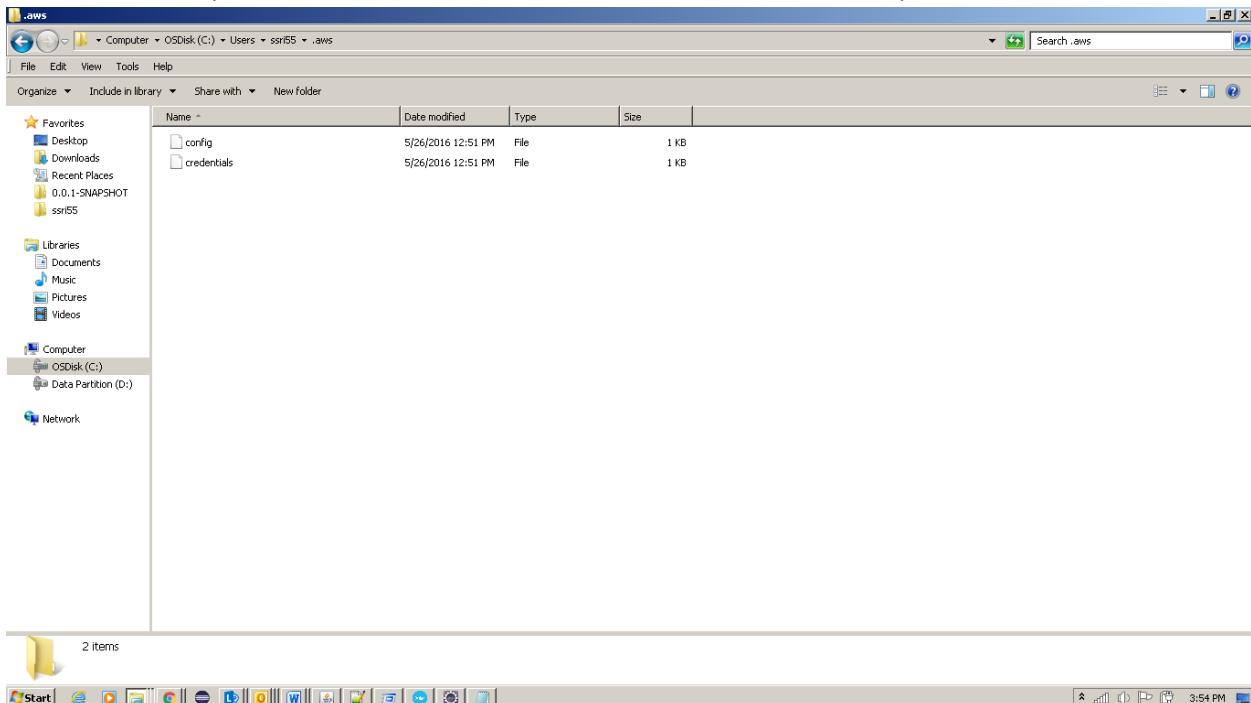
Go to below url –

<https://aws.amazon.com/cli/>

download and run the installer as per your Windows-

7. Setting up credentials for Amazon ECR

In the C:\Users\<your username>\.aws there will be two files as below snap-

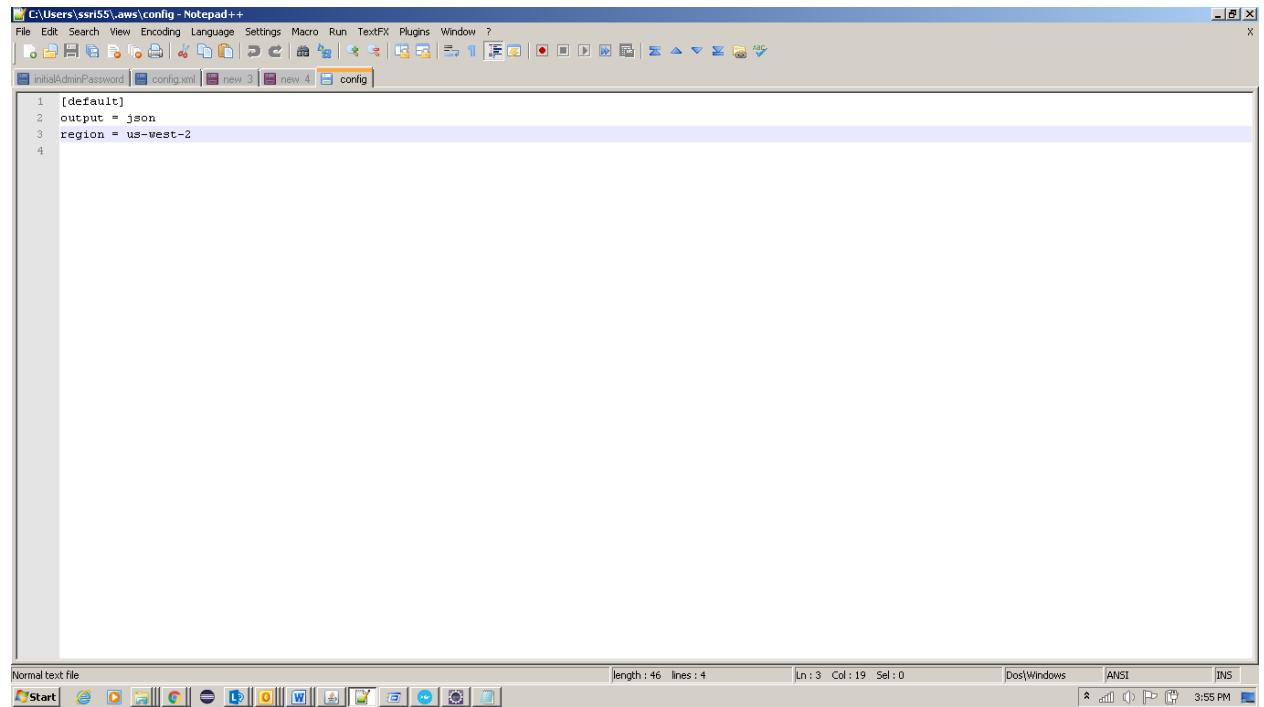


Change the content of config to –

[default]

```
output = json
```

```
region = <your region>
```



A screenshot of the Notepad++ application window. The title bar reads "C:\Users\ssri55\.aws\config - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Macro, Run, TextFX, Plugins, and Window. The toolbar has various icons for file operations. The status bar at the bottom shows "Normal text file", "length : 46 lines : 4", "Ln : 3 Col : 19 Sel : 0", "Dos\Windows", "ANSI", and "INS". The main editor area contains the following text:

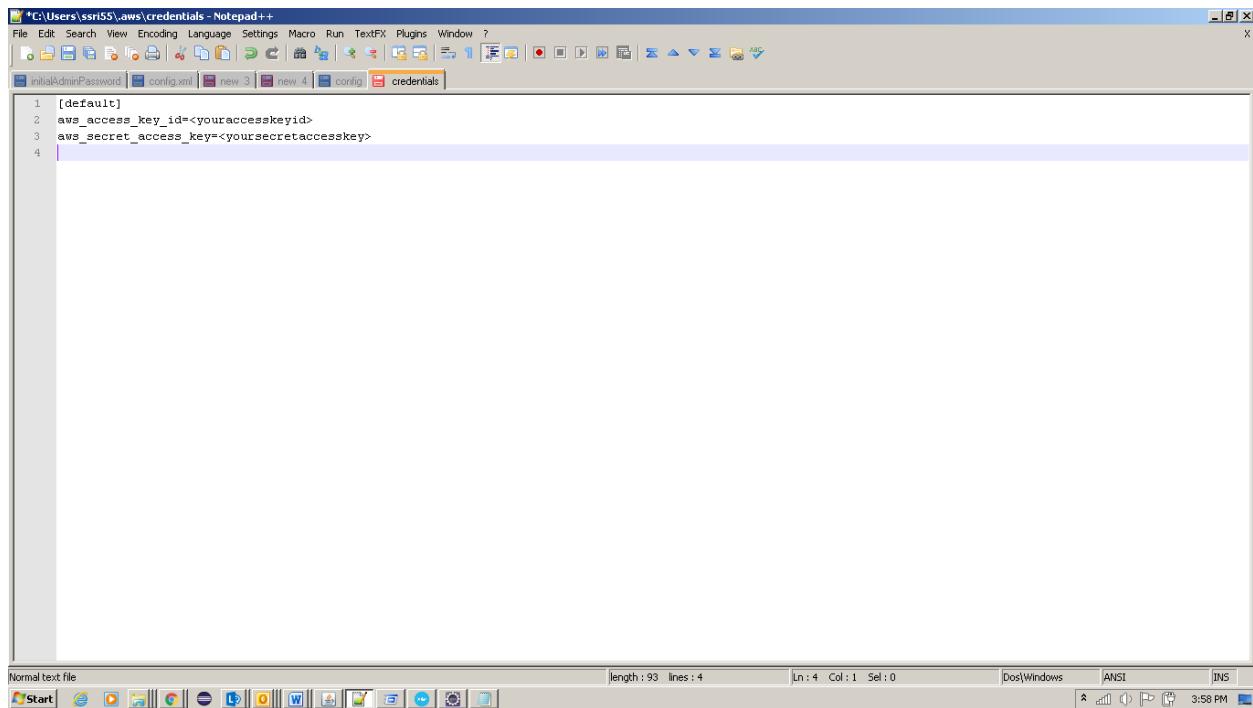
```
1 [default]
2 output = json
3 region = us-west-2
4
```

Change the credentials file content to-

```
[default]
```

```
aws_access_key_id=<your aws access key id>
```

```
aws_secret_access_key=<your aws secret access key value>
```



The screenshot shows a Notepad++ window with the file path `C:\Users\ssri55\.aws\credentials`. The content of the file is as follows:

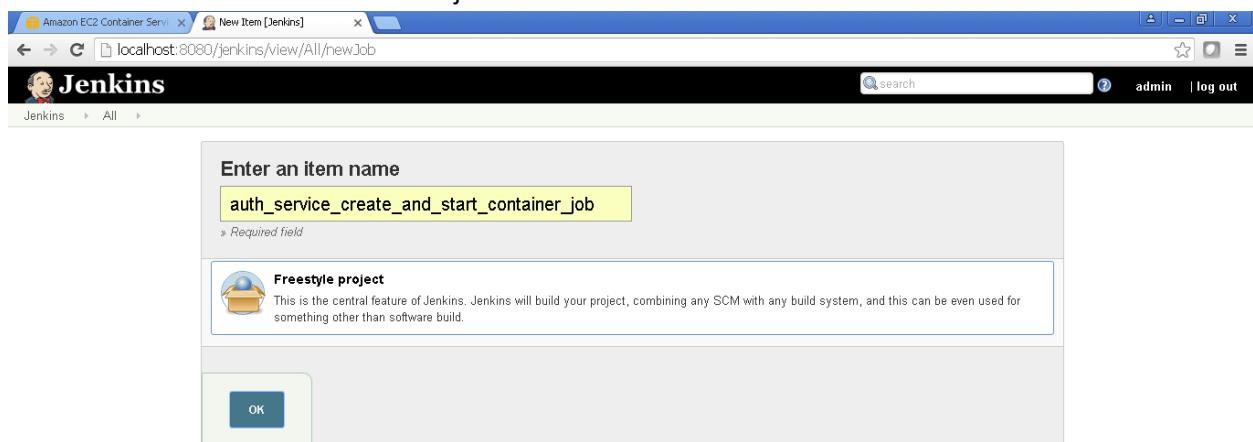
```
[default]
aws_access_key_id=<youraccesskeyid>
aws_secret_access_key=<yoursecretaccesskey>
```

8. Job creation

8.1. Job to create and start auth container

Go to home page of jenkins (<http://localhost:8080/jenkins/>)

Click on **NewItem** and enter name of job in the next screen and click OK-



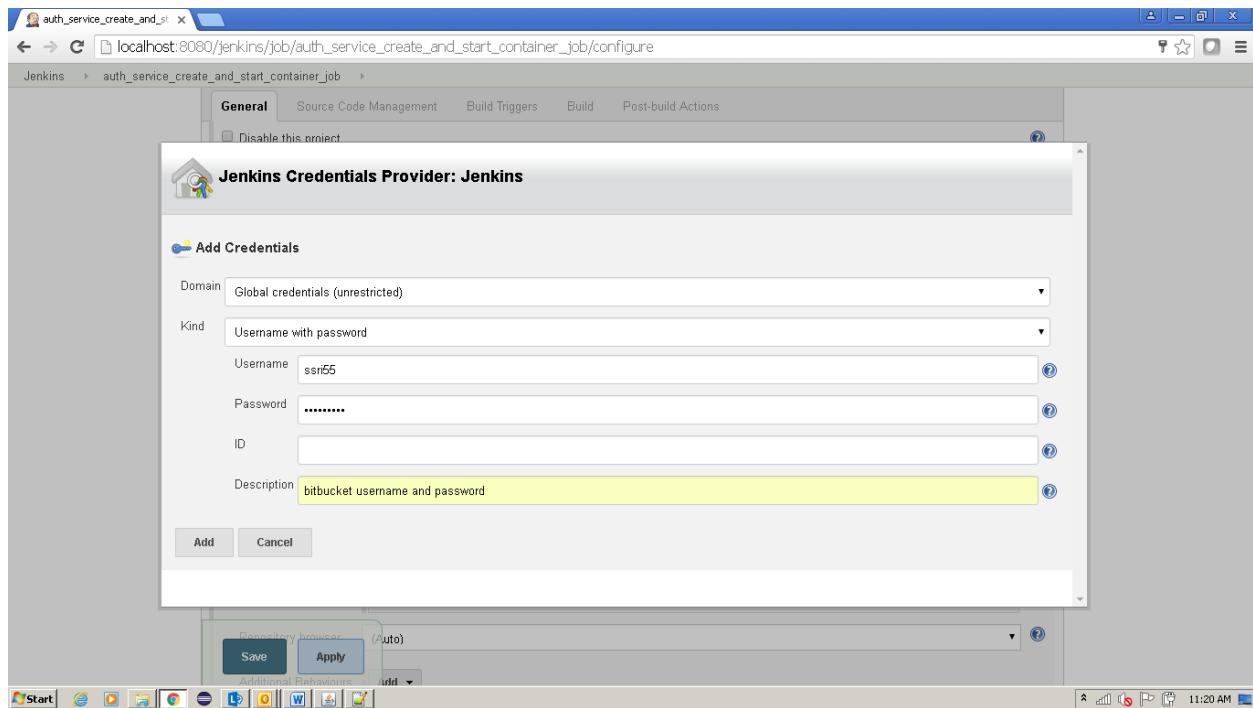
The screenshot shows a browser window for the Jenkins 'New Item' creation dialog. The URL is `localhost:8080/jenkins/view/All/newJob`. The form has a single input field labeled 'Enter an item name' containing the value 'auth_service_create_and_start_container_job'. Below the input field is a note: 'Required field'. A 'Freestyle project' section is visible, featuring a 'Freestyle project' icon and a brief description: 'This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.' At the bottom of the dialog is a blue 'OK' button.

The screenshot shows the Jenkins interface for a job named 'auth_service_create_and_start_container_job'. The top navigation bar includes links for 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build Now', 'Delete Project', and 'Configure'. On the right, there are buttons for 'add description' and 'Disable Project'. Below the navigation, the job name is displayed in bold. To the right of the job name are 'Recent Changes' and 'Workspace' links. A 'Build History' section is present with a search bar and RSS feed links. The bottom of the screen shows the Windows taskbar and system tray.

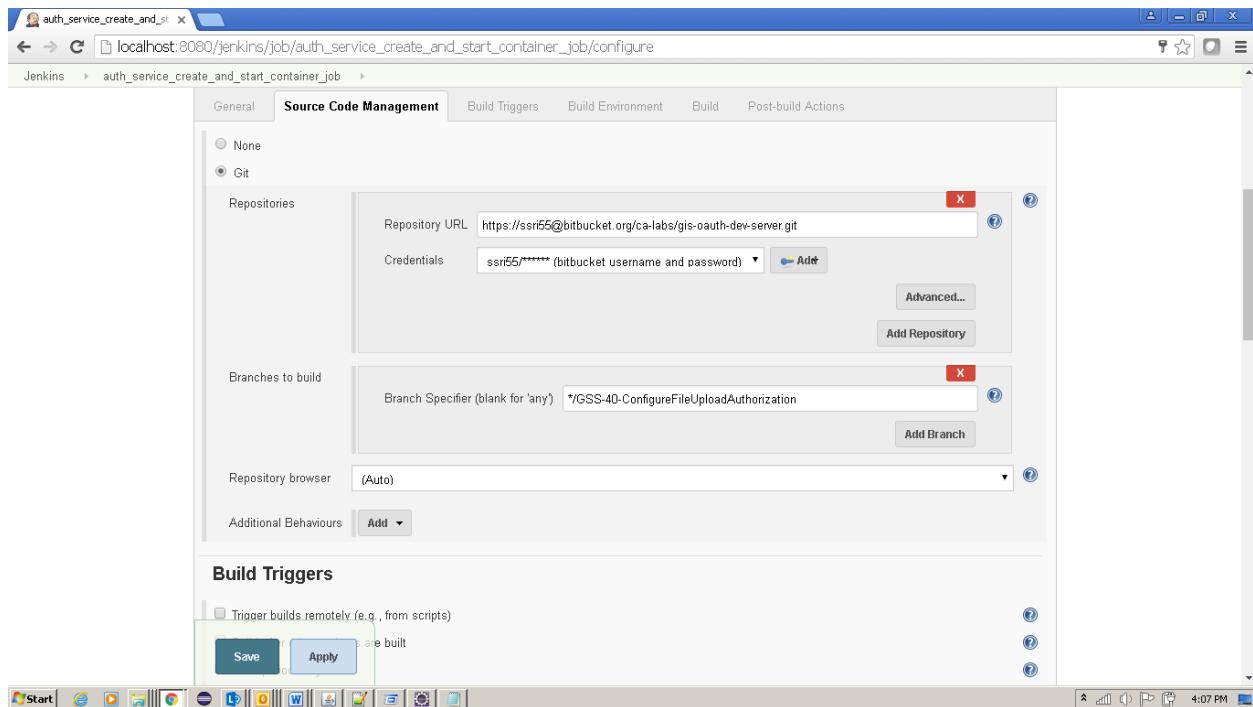
In the screen displayed, select Git in Source Code Management and then click on add in credentials-

The screenshot shows the Jenkins configuration page for the same job. The 'General' tab is selected. Under 'Source Code Management', the 'Git' option is chosen. A modal dialog box is open over the 'Repository URL' field, which is empty and has a red error message: 'Please enter Git repository.' The 'Credentials' dropdown is set to 'none'. The 'Jenkins' button in the 'Add' dropdown is highlighted. The 'Save' and 'Apply' buttons at the bottom are visible.

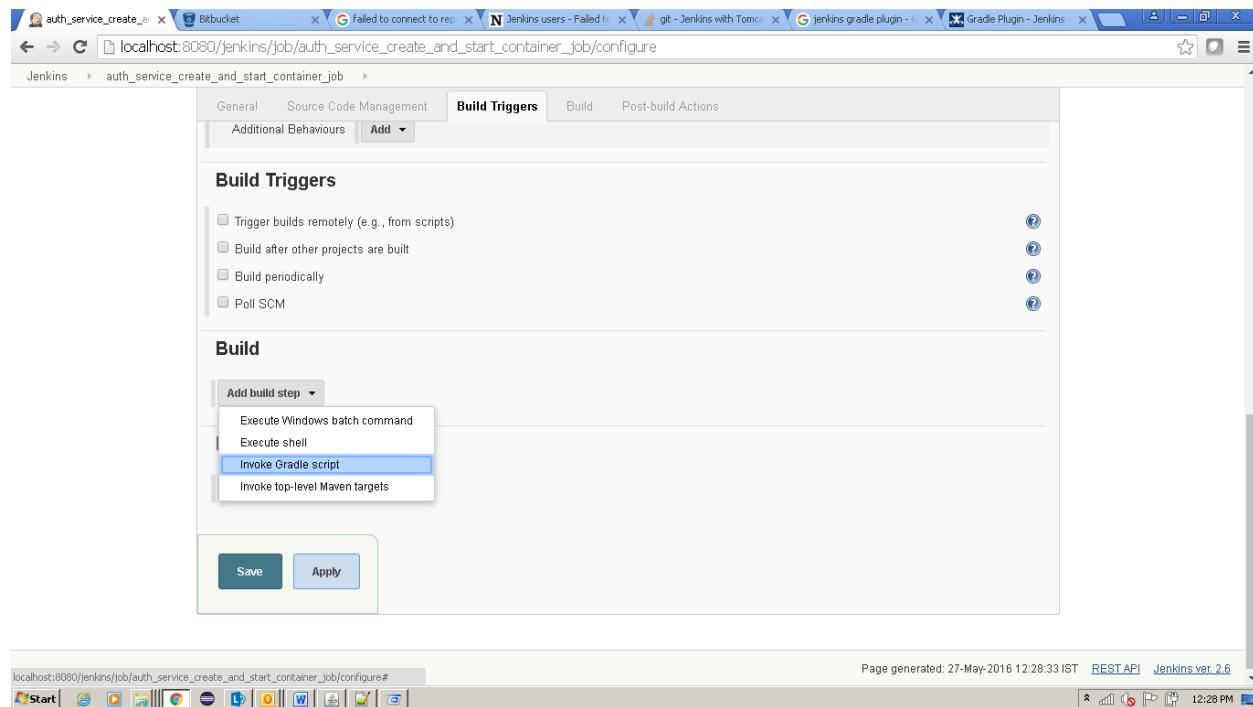
In the screen enter your bit bucket credential and click on add-



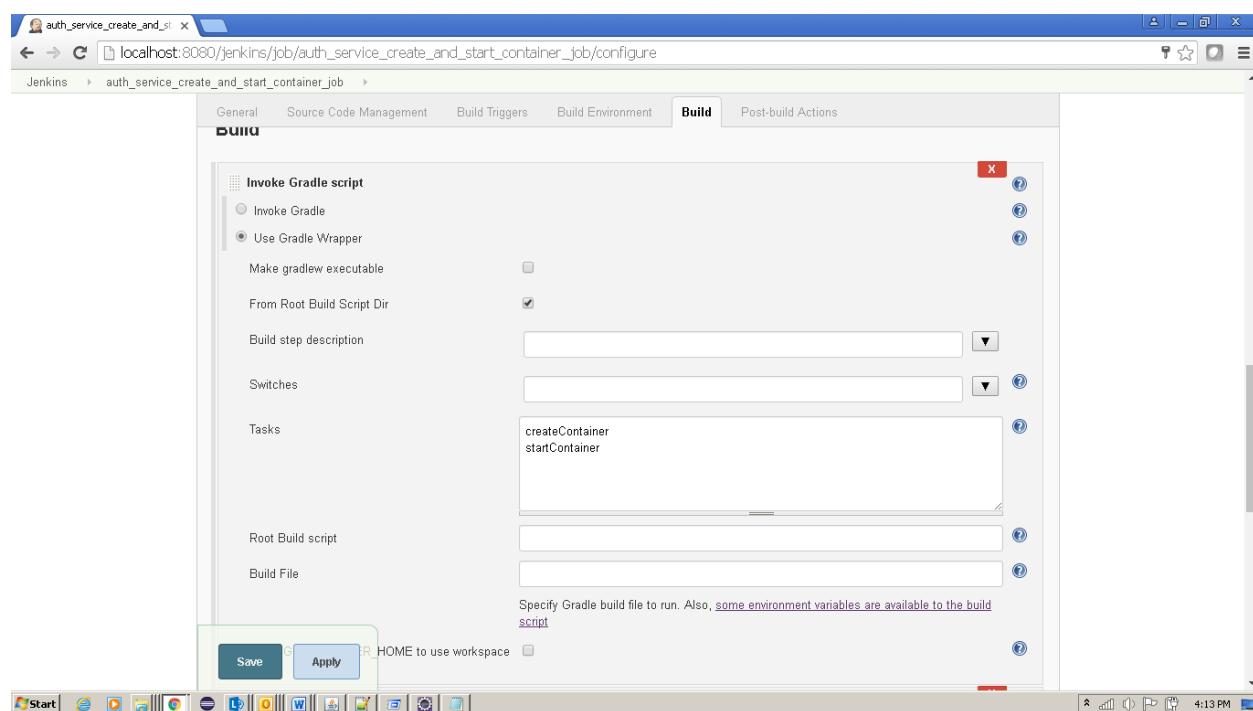
Now in the screen below select bitbucket credentials from dropdown and also enter git repository of the code you want to download. **Make sure in the branch checked in gradle.properties has correct entry for “usercertPath” according to your system certificate path e.g. usercertPath =C:\\\\Users\\\\<your username>\\\\.docker\\\\machine\\\\certs)**



Select build step **Invoke gradle script** as below-



And give following values in the section and save-



8.2. Job to create and start mail container

Similar to the above job this can also be created. Some snapshots are below-

The screenshot shows the Jenkins job configuration page for 'mail_service_job_stop_and_remove_container-job'. The 'Source Code Management' tab is selected. Under 'Repositories', a Git repository is configured with URL `https://ssr55@bitbucket.org/ca-labs/gis-mail-service.git` and credentials `ssr55*****`. Under 'Branches to build', a branch specifier `*GSS-68-mail-client-change-response-error` is defined. The 'Build Triggers' section includes 'Save' and 'Apply' buttons. The 'Build Environment' section has checkboxes for 'Inject environment variables to the build process' and 'Inject passwords to the build as environment variables'. The 'Build' section shows a dropdown menu for 'Add build step' with options like 'Conditional step (single)', 'Invoke Gradle script' (which is selected), and 'Trigger all builds on other projects'.

The screenshot shows the Jenkins job configuration page for 'mail_service_job_stop_and_remove_container-job'. The 'Build Triggers' tab is selected. It lists three trigger types: 'Build after other projects are built', 'Build periodically', and 'Poll SCM'. The 'Build Environment' and 'Build' tabs are also visible at the bottom of the configuration panel.

The screenshot shows the Jenkins job configuration page for 'mail_service_job_stop_and_remove_container-job'. The 'Build Environment' tab is selected. It contains checkboxes for 'Inject environment variables to the build process' and 'Inject passwords to the build as environment variables'. The 'General' and 'Source Code Management' tabs are also visible at the top of the configuration panel.

The screenshot shows the Jenkins job configuration page for 'mail_service_job_stop_and_remove_container-job'. The 'Build' tab is selected. A dropdown menu titled 'Add build step' is open, showing various options such as 'Conditional step (single)', 'Invoke Gradle script' (selected), and 'Trigger all builds on other projects'. Other options include 'Conditional steps (multiple)', 'Execute Windows batch command', 'Execute shell', 'Inject environment variables', 'Invoke Ant', 'Invoke top-level Maven targets', and 'Trigger all builds on other projects'. The 'General' and 'Source Code Management' tabs are also visible at the top of the configuration panel.

The screenshot shows the Jenkins job configuration interface for a job named 'mail_service_job_stop_and_remove_container-job'. The 'Build' tab is active, displaying the following settings:

- From Root Build Script Dir: checked
- Build step description: empty
- Switches: empty
- Tasks: stopContainer, removeContainer
- Root Build script: empty
- Build File: empty
- Specify Gradle build file to run. Also, some environment variables are available to the build script: empty
- Force GRADLE_USER_HOME to use workspace: unchecked

At the bottom, there are 'Add build step' and 'Post-build Actions' buttons, along with 'Save' and 'Apply' buttons.

8.3. Job to run integration test-

Follow snapshots below-

The screenshot shows the Jenkins 'New Item' creation page. The 'Integration Test Run Job' name has been entered into the 'Enter an item name' field. Below the field, there are several project types listed:

- Freestyle project**: This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Maven project**: Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- Multi-configuration project**: Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- MultiJob Project**: MultiJob Project, suitable for running other jobs
- External Job**: This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system. See [the documentation for more details](#).

At the bottom, there is an 'OK' button and a note: 'If you want to create a new item from other existing, you can use this option.'

Screenshot of the Jenkins job configuration page for "integration_test_run_job". The "Source Code Management" tab is selected.

Source Code Management

Repository URL: `https://ssn55@bitbucket.org/ca-labs/gis-services-integration-tests.git`

Credentials: `ssn55/***** (bitbucket username and password)`

Branch Specifier (blank for 'any'): `*GSS-4B-POC-RestAssured`

Additional Behaviours: `Add`

Buttons: `Save`, `Apply`

Screenshot of the Jenkins job configuration page for "integration_test_run_job". The "Build Triggers" tab is selected.

Build Triggers

- Trigger builds remotely (e.g., from scripts)
- Build after other projects are built
- Build periodically
- Poll SCM

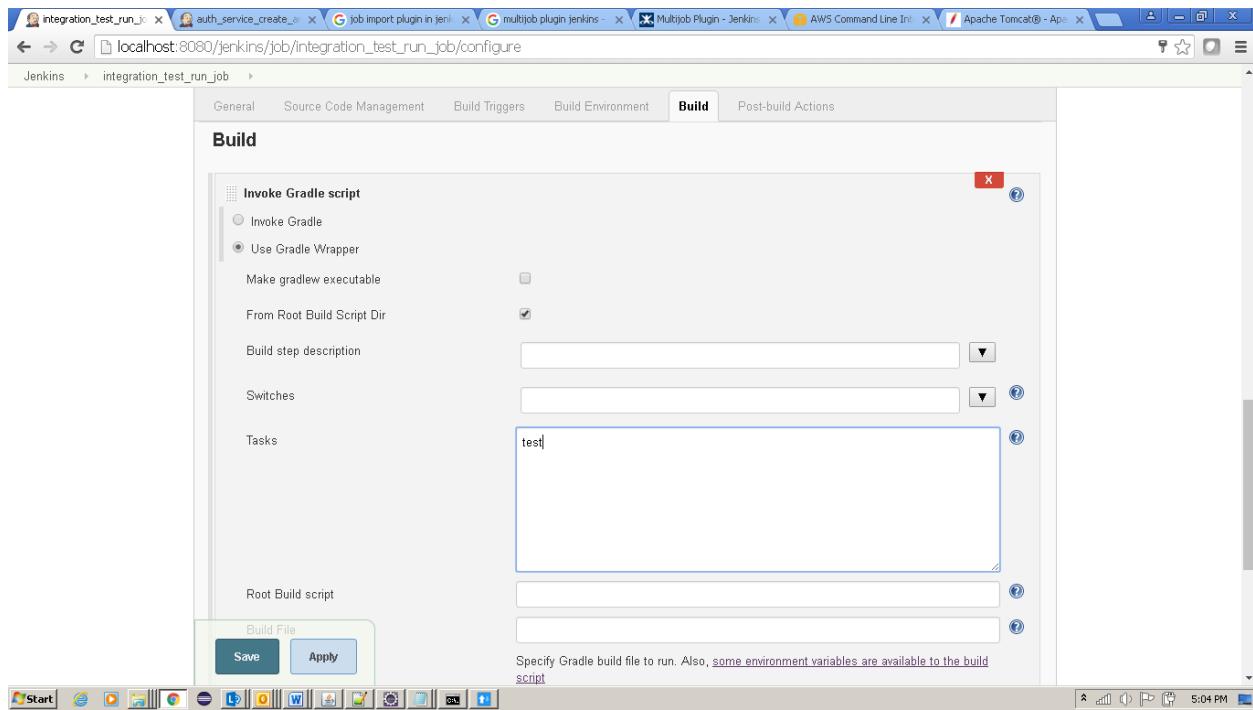
Build Environment

- Inject environment variables to the build process
- Inject passwords to the build as environment variables

Build

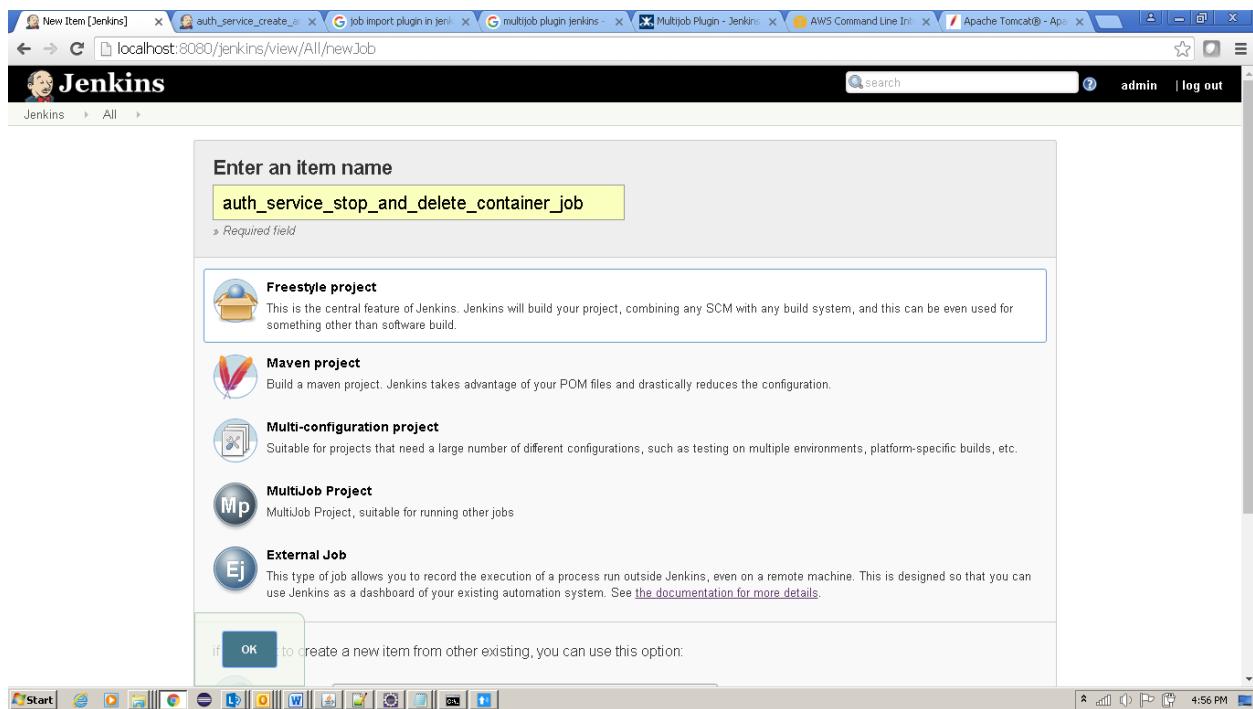
Add build step ▾

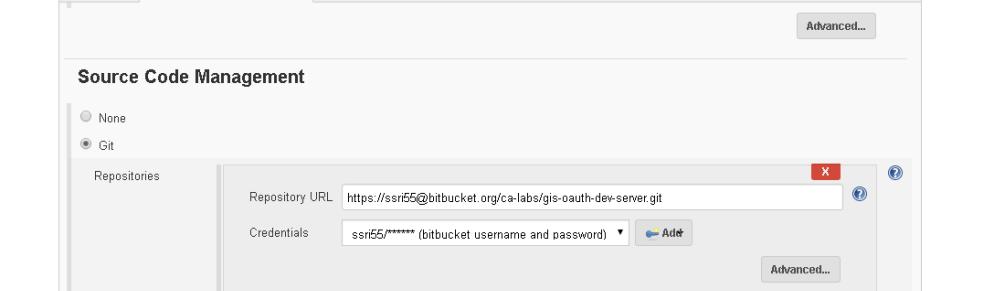
- Conditional step (single)
- Conditional steps (multiple)
- Execute Windows batch command
- Execute shell
- Inject environment variables
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Trigger/call builds on other projects



8.4. Job to stop and remove auth container

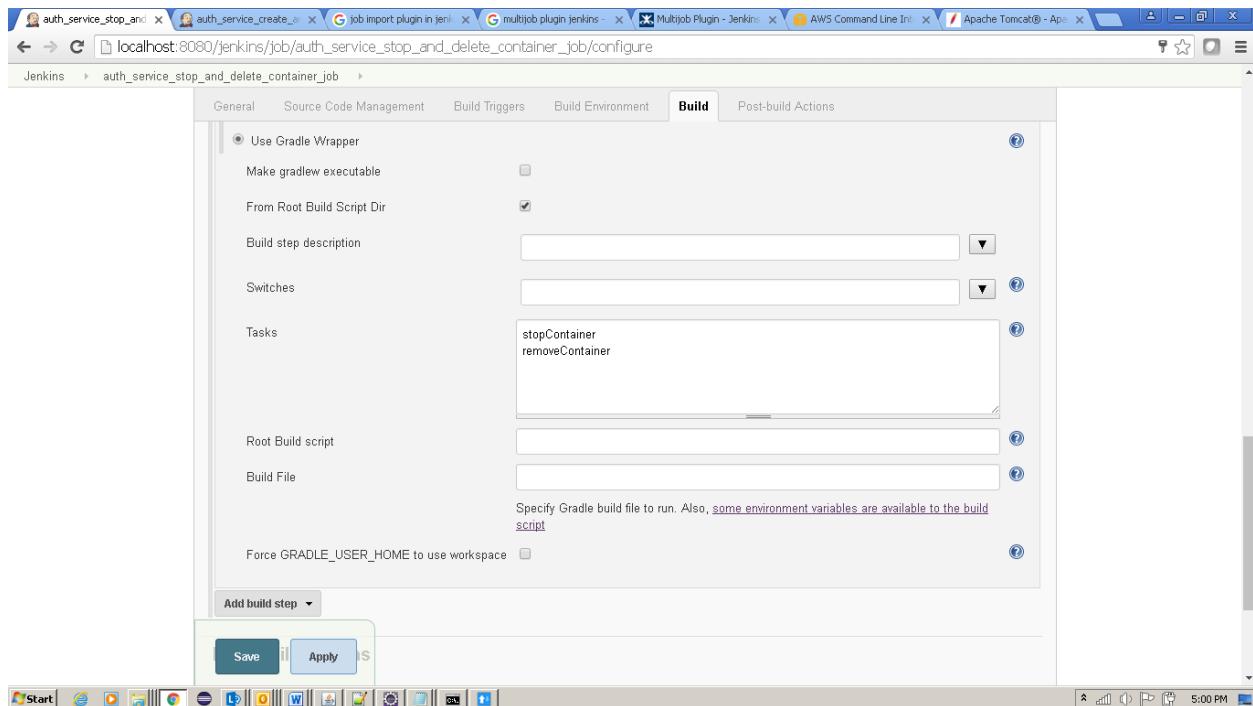
Follow snapshot below-





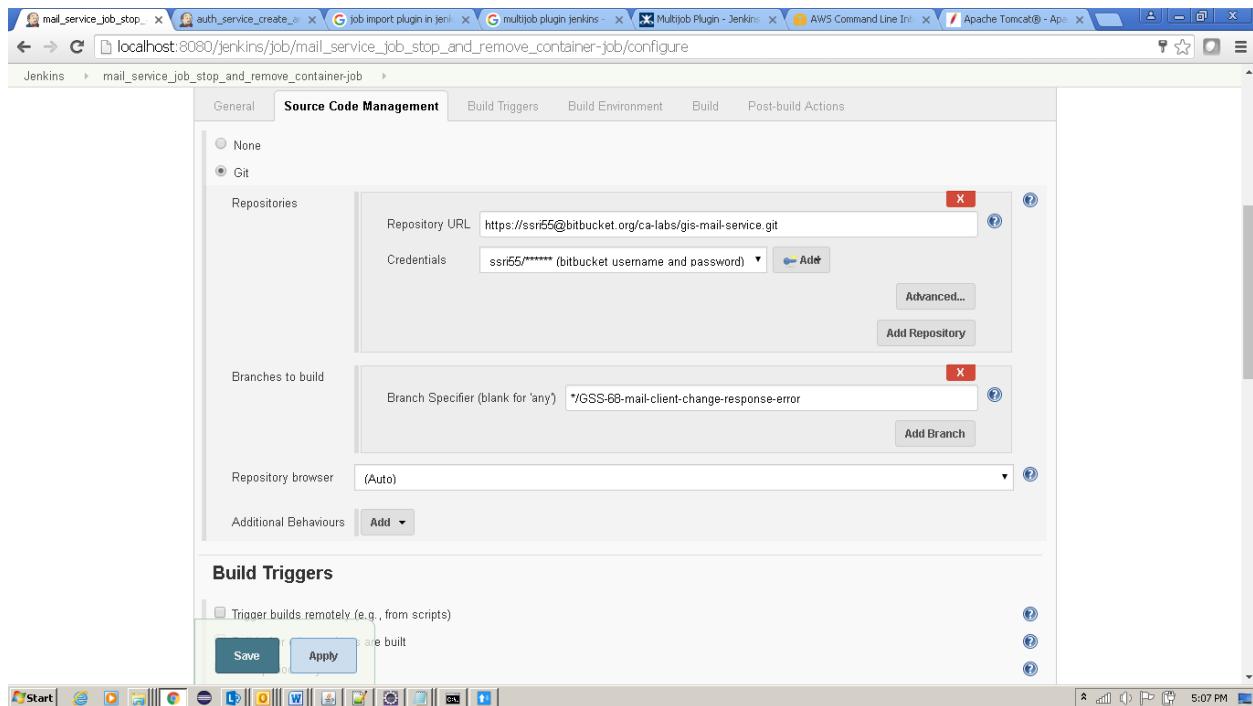
The screenshot shows the Jenkins job configuration page for the job "auth_service_stop_and_delete_container_job". The "Source Code Management" tab is selected. Under "Repositories", a single Git repository is configured with the URL `https://ssri55@bitbucket.org/ca-labs/gis-oauth-dev-server.git` and credentials `ssri55*****`. The "Branches to build" section contains a branch specifier `*/GSS-40-ConfigureFileUploadAuthorization`. The "Additional Behaviours" section is currently empty. At the bottom, there are "Save" and "Apply" buttons.

The screenshot shows the Jenkins job configuration page for 'auth_service_stop_and_delete_container_job'. The 'Build Triggers' tab is active. Below it, the 'Build Environment' and 'Build' sections are visible. In the 'Build' section, a dropdown menu titled 'Add build step' is open, showing various options: Conditional step (single), Conditional steps (multiple), Execute Windows batch command, Execute shell, Inject environment variables, Invoke Ant, Invoke Gradle script (which is highlighted with a blue selection bar), Invoke top-level Maven targets, and Trigger/call builds on other projects.



8.5. Job to stop and remove mail container

Similar to above job this can also be created. Some snapshots are below-



The screenshot shows the Jenkins job configuration interface for a job named 'mail_service_job_stop_and_remove_container-job'. The 'Build' tab is active. In the 'Invoke Gradle script' section, the 'Use Gradle Wrapper' option is selected. Under 'Tasks', 'stopContainer' and 'removeContainer' are listed. Other fields like 'From Root Build Script Dir' and 'Build step description' are also visible.

8.6. Job to buildImage and upload to ecr repository

Follow as snapshots below-

The screenshot shows the Jenkins 'New Item' creation page. A 'Freestyle project' is selected, and the item name 'buildAuthImageAndUploadToECR' is entered in the 'Enter an item name' field. Below the field, there are descriptions and icons for other project types: Maven project, Multi-configuration project, MultiJob Project, and External Job.

Screenshot of Jenkins job configuration for "buildAuthImageAndUploadToECR".

General tab:

- "This project is parameterized" checkbox is checked.
- Checkboxes for "Prepare an environment for the run", "Disable this project", and "Execute concurrent builds if necessary" are unchecked.
- "Advanced..." button is present.

Source Code Management tab:

- Radio button for "Git" is selected.
- Repositories** section:
 - Repository URL: <https://ssri55@bitbucket.org/ca-labs/gis-oauth-dev-server.git>
 - Credentials: "ssri55/***** (bitbucket username and password)" dropdown, "Add" button.
 - "Advanced..." button.
 - "Add Repository" button.
- Branches to build** section:
 - Branch Specifier (blank for 'any'): */master
 - "Add Branch" button.
- Save** and **Apply** buttons.

Screenshot of Jenkins job configuration for "buildAuthImageAndUploadToECR".

Build Environment tab:

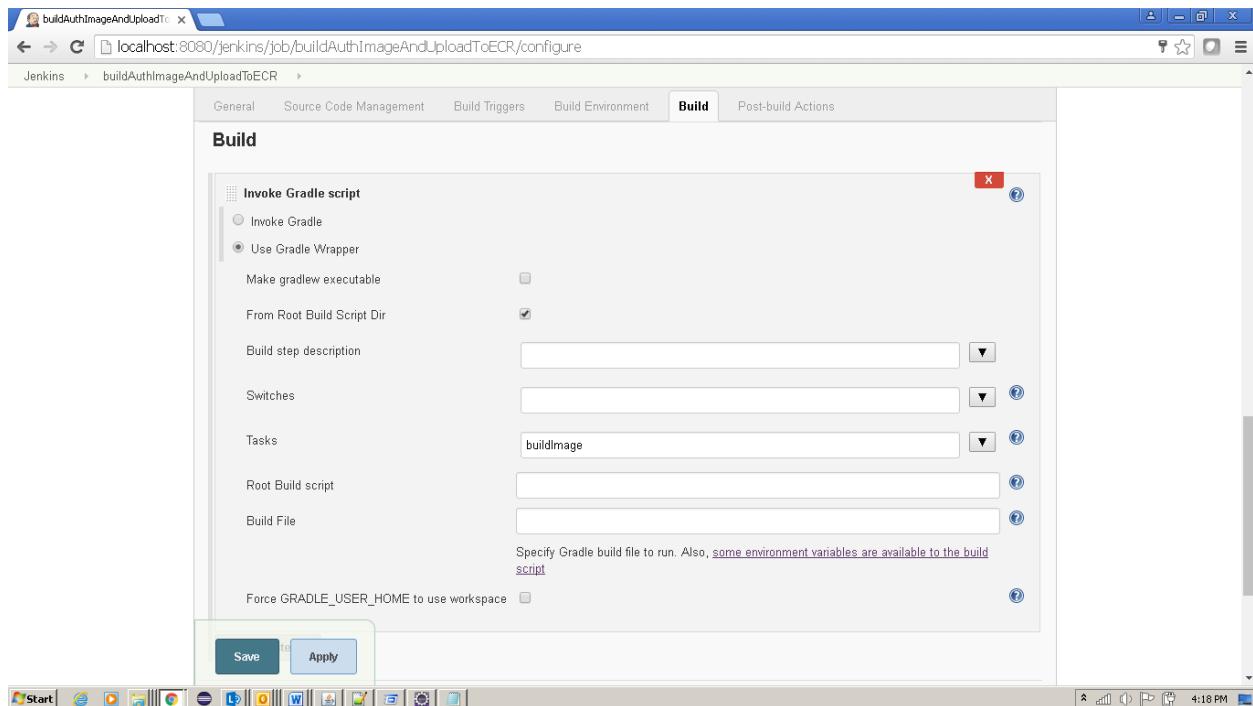
- "Poll SCM" checkbox is checked.

Build tab:

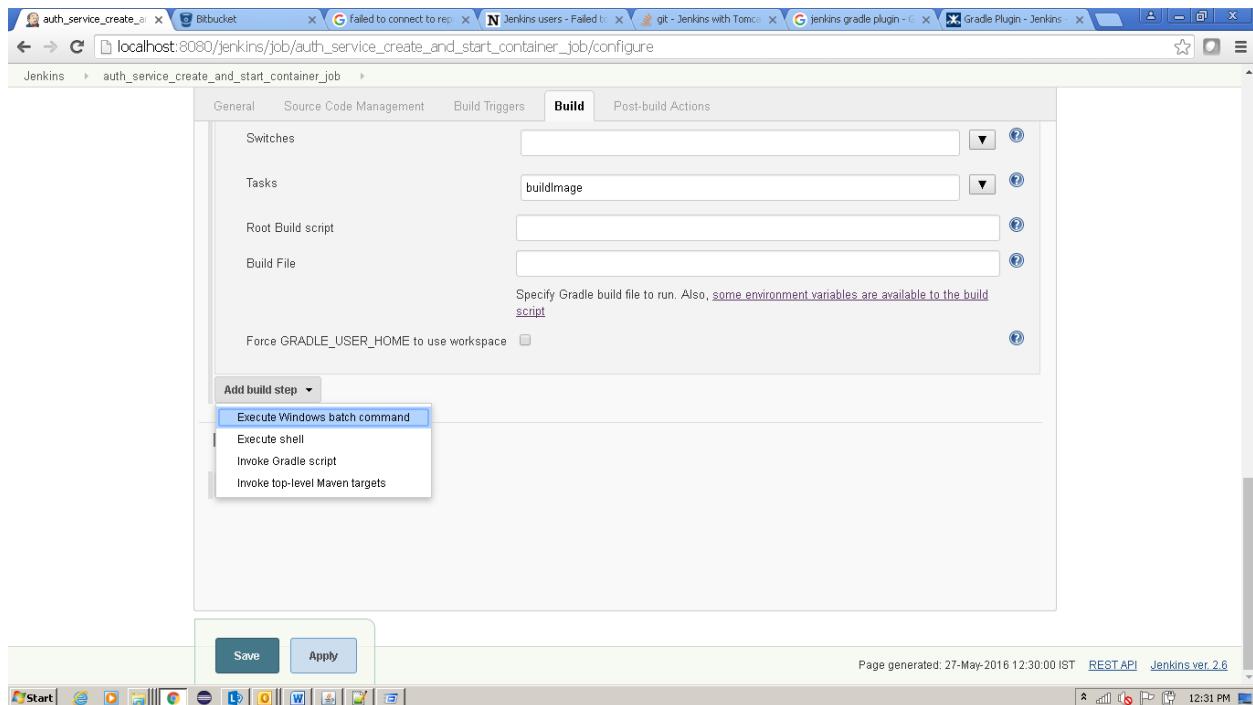
- "Add build step" dropdown menu is open, showing options:
 - Conditional step (single)
 - Conditional steps (multiple)
 - Execute Windows batch command
 - Execute shell
 - Inject environment variables
 - Invoke Ant
 - Invoke Gradle script** (selected)
 - Invoke top-level Maven targets
 - Trigger/call builds on other projects

Page footer:

Page generated: 27-May-2016 16:17:25 IST REST API Jenkins ver. 2.6



Now Select buildstep dropdown of execute Windows batch command-



Now in the window paste following commands and save-

```

set REG_ADDRESS=285637842202.dkr.ecr.us-west-2.amazonaws.com
set REPO=gis-auth-service-ecr
set IMAGE_VERSION=latest
set LOCAL_IMAGE_NAME=gis-oauth-dev-server
set WORKSPACE_PATH=
C:\Users\ssri55\jenkins\workspace\auth_service_create_and_start_container_job

aws configure get default

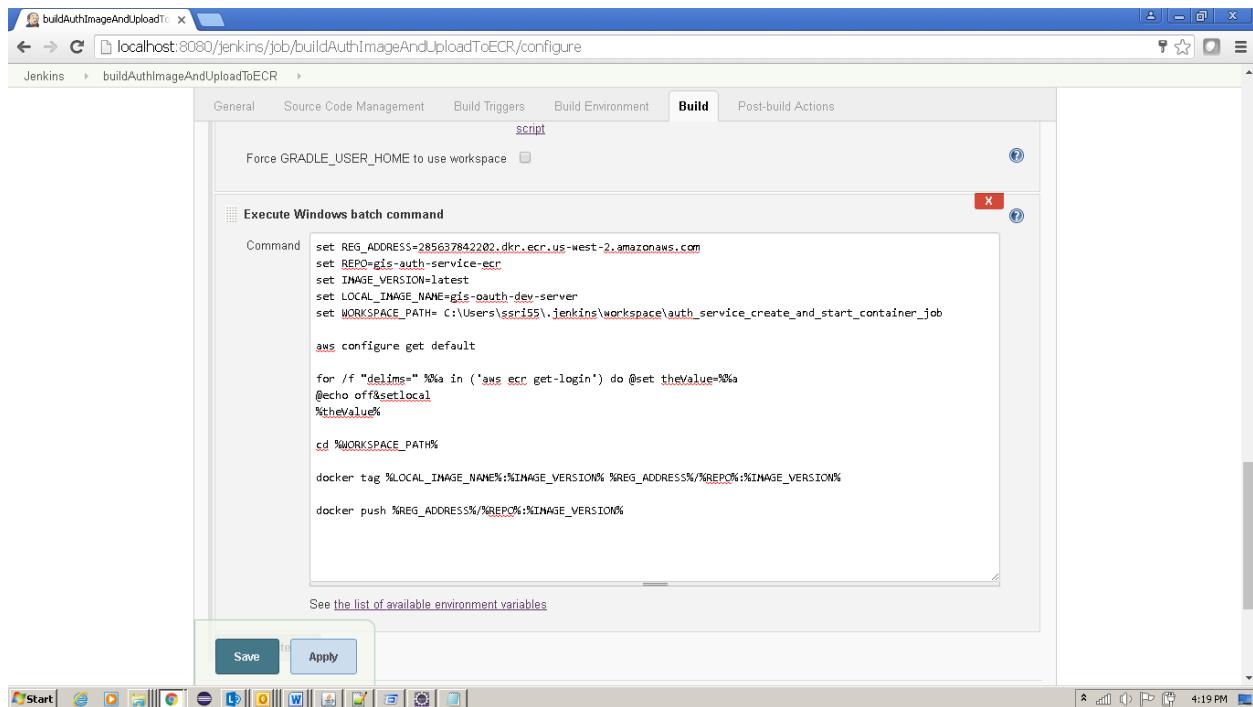
for /f "delims=" %%a in ('aws ecr get-login') do @set theValue=%%a
@echo off&setlocal
%theValue%

cd %WORKSPACE_PATH%

docker tag %LOCAL_IMAGE_NAME%:%IMAGE_VERSION%
%REG_ADDRESS%/%REPO%:%IMAGE_VERSION%

docker push %REG_ADDRESS%/%REPO%:%IMAGE_VERSION%

```



The job is created and can be run now.

8.7.Job to Run Jmeter Job

Follow as snapshots below-

The screenshot shows the Jenkins configuration page for a project named "JMeter Integration with Jenkins". The "General" tab is selected. The "Project name" field contains "JMeter Integration with Jenkins". The "Description" field is empty. Under "Discard old builds", several options are listed: "Discard old builds", "GitHub project", "This project is parameterized", "Throttle builds", "Disable this project", and "Execute concurrent builds if necessary". A "Source Code Management" section is present, showing "None" selected. At the bottom are "Save" and "Apply" buttons.

The screenshot shows the Jenkins configuration page for the same project. The "Source Code Management" tab is selected. Under "Repositories", a "Git" repository is configured with a URL of "https://gmaha@bitbucket.org/ca-labs/gis-services-integration-tests.git" and credentials "gmaha2*****". Under "Branches to build", a branch specifier "*/GSS-67-CreatePerfTest" is defined. The "Build Triggers" section includes "Trigger builds remotely (e.g., from scripts)" and "Build periodically". At the bottom are "Save" and "Apply" buttons.

JMeter Config [Jenkins] x localhost:8080/jenkins/job/JMeter%20Integration%20with%20Jenkins/configure

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Build

Invoke Gradle script

Gradle Version (Default)

Use Gradle Wrapper

Build step description

Switches

Tasks clean jmRun jmReport

Root Build script

Build File C:\Program Files\CA\gis-services-integration-tests\build.gradle

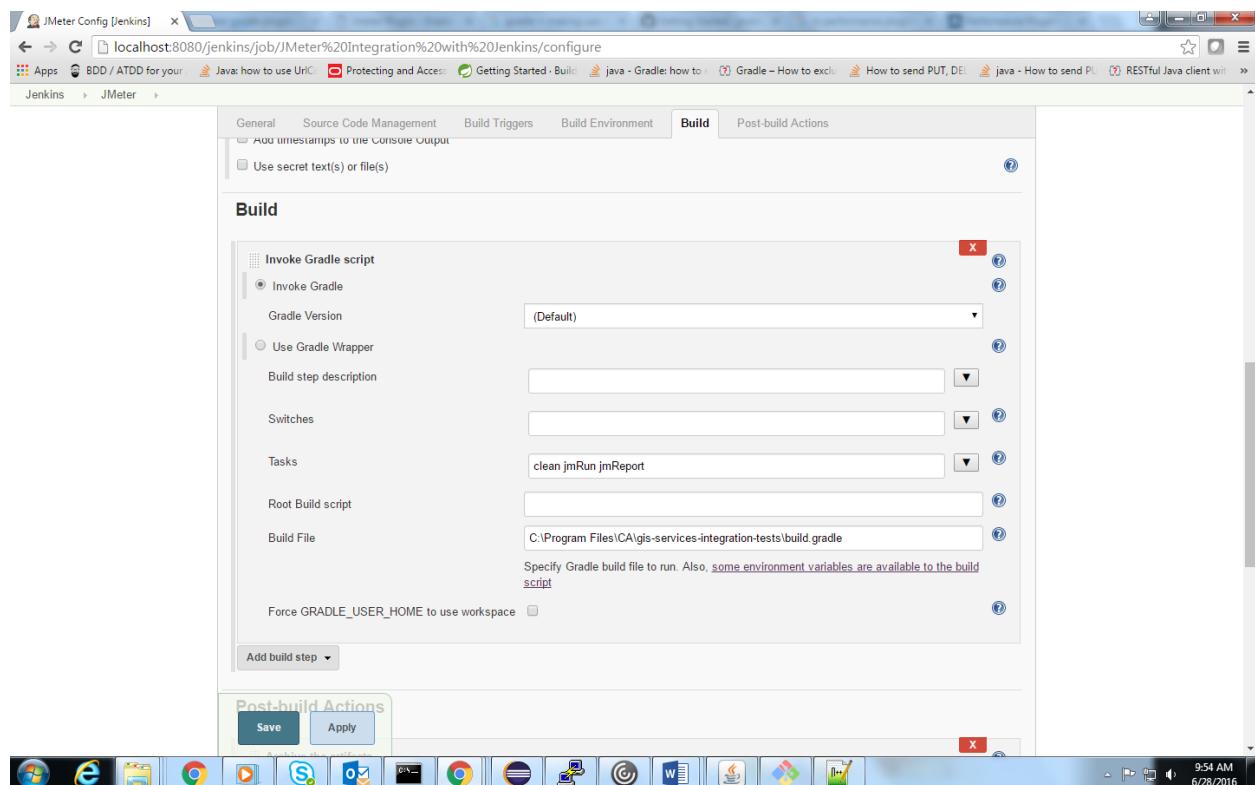
Specify Gradle build file to run. Also, [some environment variables are available to the build script](#)

Force GRADLE_USER_HOME to use workspace

Add build step

Post-build Actions

Save Apply



localhost:8080/jenkins/me x localhost:8080/jenkins/me/my-views/view/All/job/JMeter%20Integration%20with%20Jenkins/HTML_Report_and_Graph_Report/

Back to JMeter Zip

Back to JMeter Zip

Back to JMeter Zip

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Post-build Actions

Publish HTML reports

Reports

HTML directory to archive C:\Program Files\CA\gis-services-integration-tests\build\jmeter-report\

Index page[s] *.jmx* html

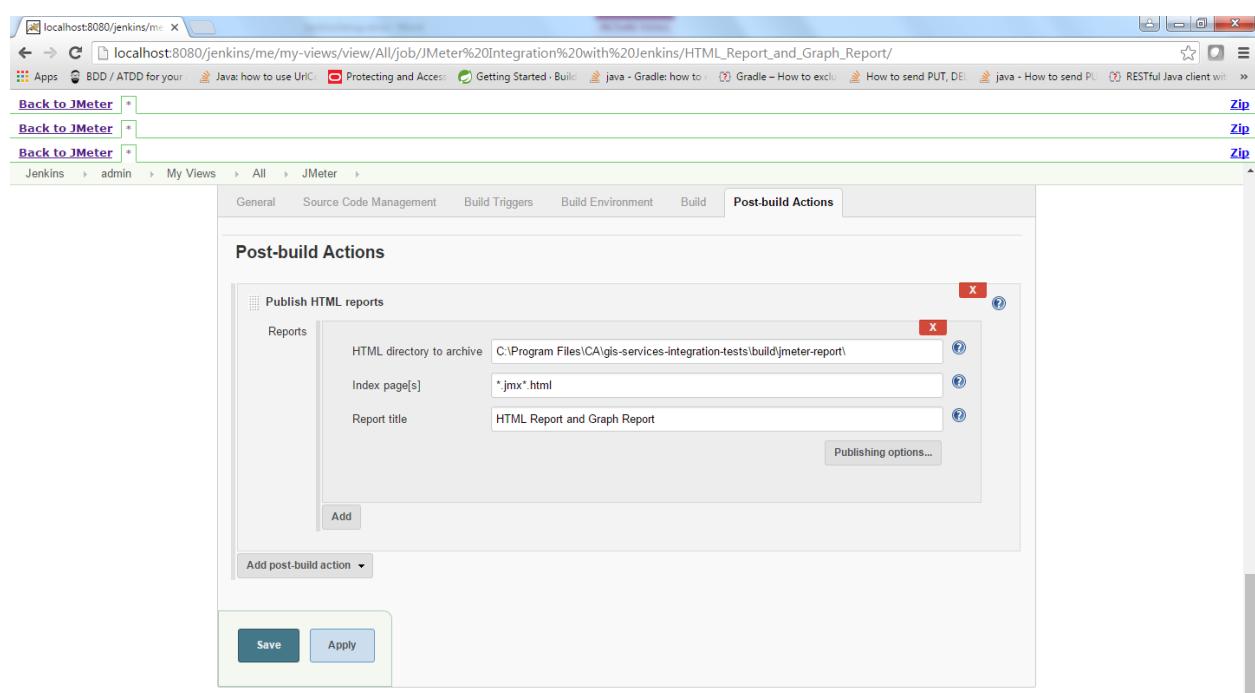
Report title HTML Report and Graph Report

Publishing options...

Add

Add post-build action

Save Apply



8.8.Master Job to call child jobs -

Snapshots below-

The screenshot shows the Jenkins 'New Item' creation interface. In the top search bar, 'MasterJob' is typed. Below the search bar, there's a section titled 'Enter an item name' with 'MasterJob' entered. A list of project types is shown, with 'MultiJob Project' highlighted by a blue border. Other options include 'Freestyle project', 'Maven project', 'Multi-configuration project', and 'External Job'. At the bottom of the list, there's a note about creating a new item from existing ones, with an 'OK' button and a 'Type to autocomplete' input field.

The screenshot shows the Jenkins 'MasterJob' configuration page. The 'Build Triggers' tab is active. Under 'Build Environment', there are checkboxes for 'Inject environment variables to the build process' and 'Inject passwords to the build as environment variables'. The 'Build' section contains a 'Add build step' dropdown menu. The menu lists several options: 'Conditional step (single)', 'Conditional steps (multiple)', 'Execute Windows batch command', 'Execute shell', 'Inject environment variables', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Multijob Phase' (which is currently selected and highlighted in blue), and 'Trigger/call builds on other projects'. The status bar at the bottom indicates the page was generated on May 27, 2016, at 17:10:31 IST.

localhost:8080/jenkins/job/MasterJob/configure

General Source Code Management Multijob specific configuration Build Triggers Build Environment Build Post-build Actions

Inject passwords to the build as environment variables

Build

MultiJob Phase

Phase name CreateAndStartContainers

Phase jobs

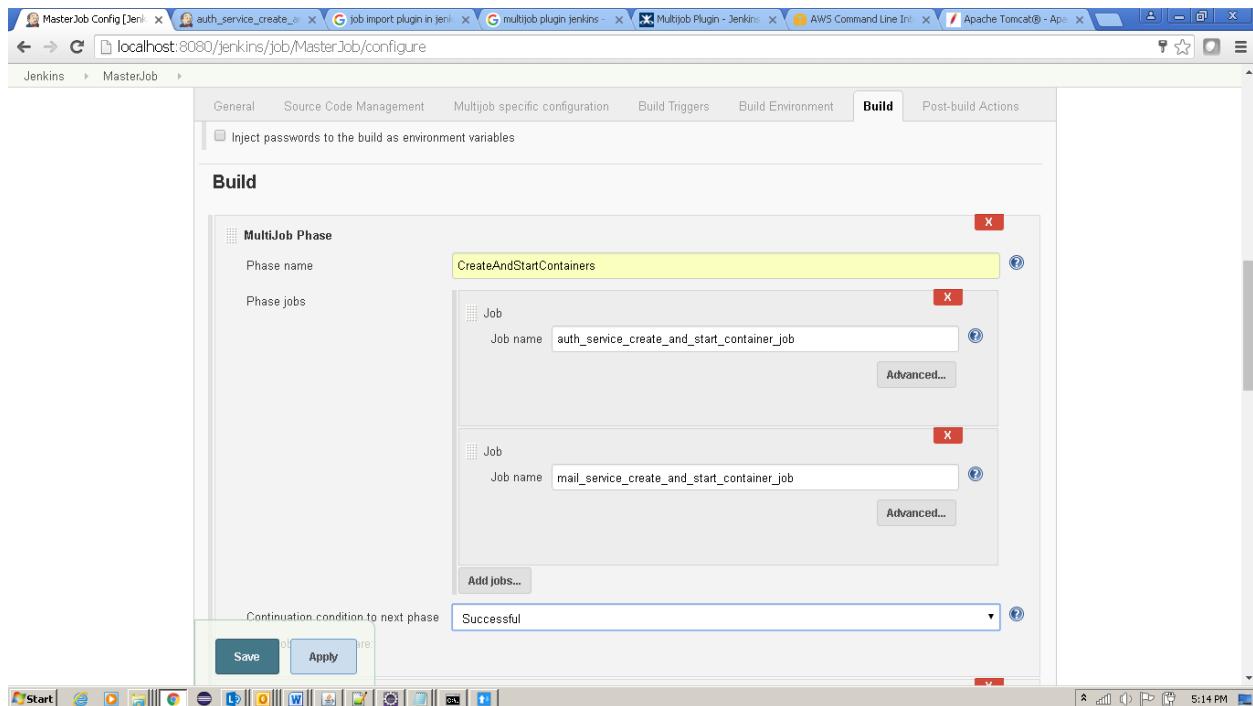
Job Job name auth_service_create_and_start_container_job

Job Job name mail_service_create_and_start_container_job

Add jobs...

Continuation condition to next phase Successful

Save **Apply**



localhost:8080/jenkins/job/MasterJob/configure

General Source Code Management Multijob specific configuration Build Triggers Build Environment Build Post-build Actions

when jobs' statuses are:

MultiJob Phase

Phase name RunTests

Phase jobs

Job Job name integration_test_run_job

Add jobs...

Continuation condition to next phase Always

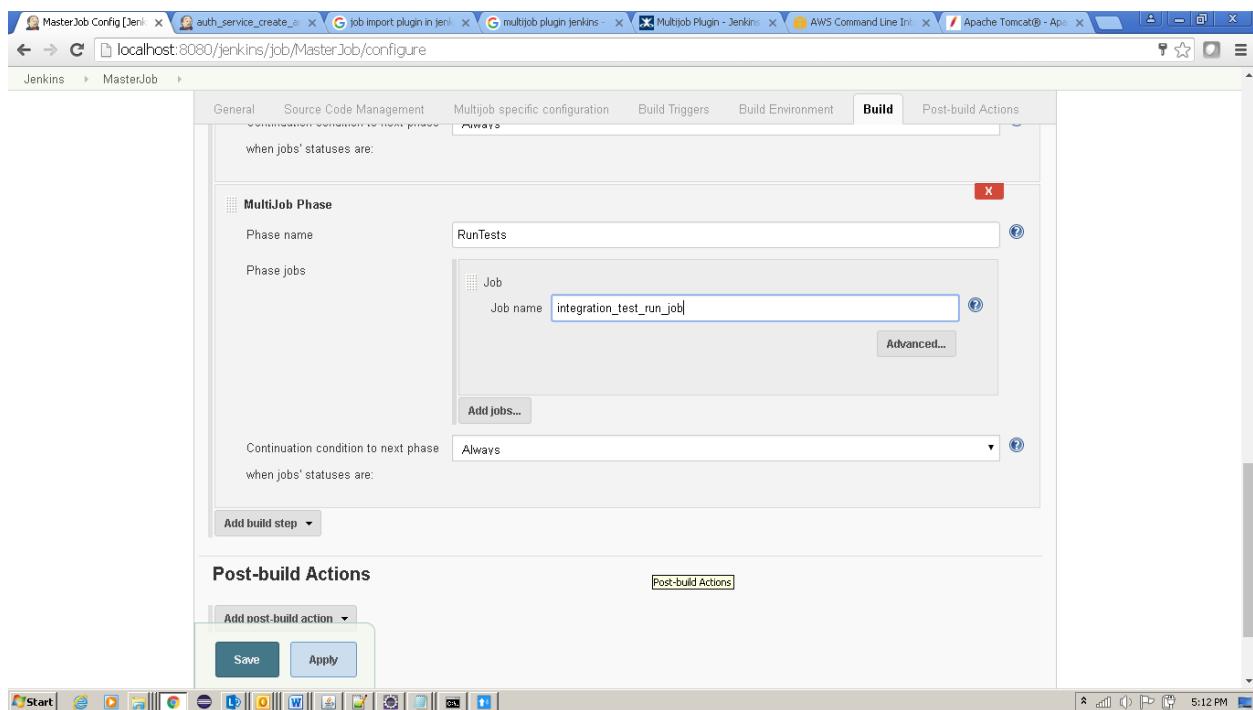
when jobs' statuses are:

Add build step ▾

Post-build Actions

Add post-build action ▾ Post-build Actions

Save **Apply**



Screenshot of Jenkins MasterJob configuration page (localhost:8080/jenkins/job/MasterJob/configure) showing the Build tab.

General tab is selected. Under **Multijob specific configuration**, the **Build Triggers** section shows "when jobs' statuses are:" set to **Always**.

Build tab is selected. Under **MultiJob Phase**, a new phase named **RunTests** is being configured. It contains a single job named **integration_test_run_job**. The continuation condition for the next phase is set to **Always**.

Post-build Actions section is present but currently empty.

Buttons at the bottom: **Save** and **Apply**.

Screenshot of Jenkins MasterJob configuration page (localhost:8080/jenkins/job/MasterJob/configure) showing the Build tab.

General tab is selected. Under **Multijob specific configuration**, the **Build Triggers** section shows "when jobs' statuses are:" set to **Always**.

Build tab is selected. Under **MultiJob Phase**, a new phase named **StopAndRemoveContainers** is being configured. It contains two jobs: **mail_service_job_stop_and_remove_container-job** and **auth_service_stop_and_delete_container_job**. The continuation condition for the next phase is set to **Always**.

Post-build Actions section is present but currently empty.

Buttons at the bottom: **Save** and **Apply**.

MultiJob Project MasterJob

S	W	Job	Last Success	Last Failure	Last Duration	Console
●	○	MasterJob	N/A	N/A	N/A	
<i>CreateAndStartContainers</i>						
●	○	auth_service_create_and_start_container_job	N/A	N/A	N/A	
●	○	mail_service_create_and_start_container_job	N/A	N/A	N/A	
<i>Run Tests</i>						
●	○	integration_test_run_job	N/A	N/A	N/A	
<i>StopAndRemoveContainers</i>						
●	○	mail_service_stop_and_remove_container_job	N/A	N/A	N/A	
●	○	auth_service_stop_and_delete_container_job	N/A	N/A	N/A	

Icon: S M L Legend:

9. Running Job

9.1.MasterJob-

We need to click BuildNow-

MultiJob Project MasterJob

S	W	Job	Last Success	Last Failure	Last Duration	Console
●	○	MasterJob	Building	N/A	N/A	
<i>CreateAndStartContainers</i>						
●	○	auth_service_create_and_start_container_job	N/A	N/A	N/A	
●	○	mail_service_create_and_start_container_job	N/A	N/A	N/A	
<i>Run Tests</i>						
●	○	integration_test_run_job	N/A	N/A	N/A	
<i>StopAndRemoveContainers</i>						
●	○	mail_service_stop_and_remove_container_job	N/A	N/A	N/A	
●	○	auth_service_stop_and_delete_container_job	N/A	N/A	N/A	

Icon: S M L Legend:

Jenkins

MasterJob [Jenkins]

localhost:8080/jenkins/job/MasterJob/

search admin log out DISABLE AUTO REFRESH

Back to Dashboard Status Changes Workspace Build Now Delete MultiJob Project Configure

Build History trend → find #1 27-May-2016 17:16 RSS for all RSS for failures

MultiJob Project MasterJob

add description Disable Project

S	W	Job	Last Success	Last Failure	Last Duration	Console
●	○	MasterJob	N/A	N/A	N/A	
CreateAndStartContainers						
●	○	auth_service_create_and_start_container_job	N/A	N/A	N/A	
●	○	mail_service_create_and_start_container_job	N/A	N/A	N/A	
Run Tests						
●	○	integration_test_run_job	N/A	N/A	N/A	
StopAndRemoveContainers						
●	○	mail_service_stop_and_remove_container_job	N/A	N/A	N/A	
●	○	auth_service_stop_and_delete_container_job	N/A	N/A	N/A	

Icon: S M L Legend RSS for all RSS for failures RSS for just latest builds

Jenkins

MasterJob [Jenkins]

localhost:8080/jenkins/job/MasterJob/

search admin log out DISABLE AUTO REFRESH

Back to Dashboard Status Changes Workspace Build Now Delete MultiJob Project Configure

Build History trend → find #1 27-May-2016 17:16 RSS for all RSS for failures

MultiJob Project MasterJob

add description Disable Project

S	W	Job	Last Success	Last Failure	Last Duration	Console
●	○	MasterJob	N/A	N/A	9.9 sec and counting	
CreateAndStartContainers						
●	○	auth_service_create_and_start_container_job	N/A	N/A	N/A	
●	○	mail_service_create_and_start_container_job	N/A	N/A	3.2 sec and counting	
Run Tests						
●	○	integration_test_run_job	N/A	N/A	N/A	
StopAndRemoveContainers						
●	○	mail_service_stop_and_remove_container_job	N/A	N/A	N/A	
●	○	auth_service_stop_and_delete_container_job	N/A	N/A	N/A	

Icon: S M L Legend RSS for all RSS for failures RSS for just latest builds

Jenkins > MasterJob > MasterJob

MasterJob

Changes **Workspace** **Build Now** **Delete MultiJob Project** **Configure**

Build History **trend**

#1 27-May-2016 17:16

- Changes**
- Console Output** **Build Information**
- Edit Build Information**
- Environment Variables**

S	W	Job	Last Success	Last Failure	Last Duration	Console
●	☀	MasterJob	N/A	N/A	31 sec and counting	
●	☀	CreateAndStartContainers				
●	☁	auth_service_create_and_start_container_job	N/A	N/A	N/A	
●	☀	mail_service_create_and_start_container_job	N/A	N/A	25 sec and counting	
●	☀	Run Tests				
●	☀	integration_test_run_job	N/A	N/A	N/A	
●	☀	StopAndRemoveContainers				
●	☀	mail_service_stop_and_remove_container_job	N/A	N/A	N/A	
●	☀	auth_service_stop_and_delete_container_job	N/A	N/A	N/A	

Icon: S M L Legend: RSS for all RSS for failures RSS for just latest builds

Workspace **Recent Changes**

9.2. Build and upload image to ECR-

Click BuildNow on the job of buildImage.

localhost:8080/jenkins/job/buildAuthImageAndUploadToECR/

Jenkins

Project buildAuthImageAndUploadToECR

Build History **trend**

#3 28-May-2016 00:57
#2 28-May-2016 00:47
#1 27-May-2016 19:48

Permalinks

- Last build (#3), 8 min 16 sec ago
- Last failed build (#3), 8 min 16 sec ago
- Last unsuccessful build (#3), 8 min 16 sec ago
- Last completed build (#0), 8 min 16 sec ago

RSS for all **RSS for failures**

localhost:8080/jenkins/job/buildAuthImageAndUploadToECR/build?delay=0sec

Page generated: 28-May-2016 01:05:27 IST REST API Jenkins ver. 2.6

buildAuthImageAndUploadToECR

localhost:8080/jenkins/job/buildAuthImageAndUploadToECR/

Jenkins

Jenkins > buildAuthImageAndUploadToECR >

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

Project buildAuthImageAndUploadToECR

[add description](#)

[Disable Project](#)

Build History

trend [trend](#)

find

#4 28-May-2016 01:05

#3

#2

#1

Changes

Console Output

Edit Build Information

Delete Build

Environment Variables

Git Build Data

No Tags

Recent Changes

Permalinks

- [Last build \(#4\), 1 min 11 sec ago](#)
- [Last failed build \(#4\), 1 min 11 sec ago](#)
- [Last unsuccessful build \(#4\), 1 min 11 sec ago](#)
- [Last completed build \(#4\), 1 min 11 sec ago](#)

localhost:8080/jenkins/job/buildAuthImageAndUploadToECR/4/console

Page generated: 28-May-2016 01:06:46 IST REST API Jenkins ver. 2.6

Start

1:06 AM

