

# DEVOPS PROFESSIONAL CERTIFICATION PROGRAM

## Lab – 3: Getting Started with Jenkins

- Start Jenkins Service & Setup basic tools
- Manage Jenkins – Plugins, Proxy Setting & Tool Configurations (Java & Git)
- Create Jenkins Job (Freestyle)
  - Integrate SCM (Git/Gitlab)
  - Perform build process using shell commands
  - Explore configure one or more Repositories/branches

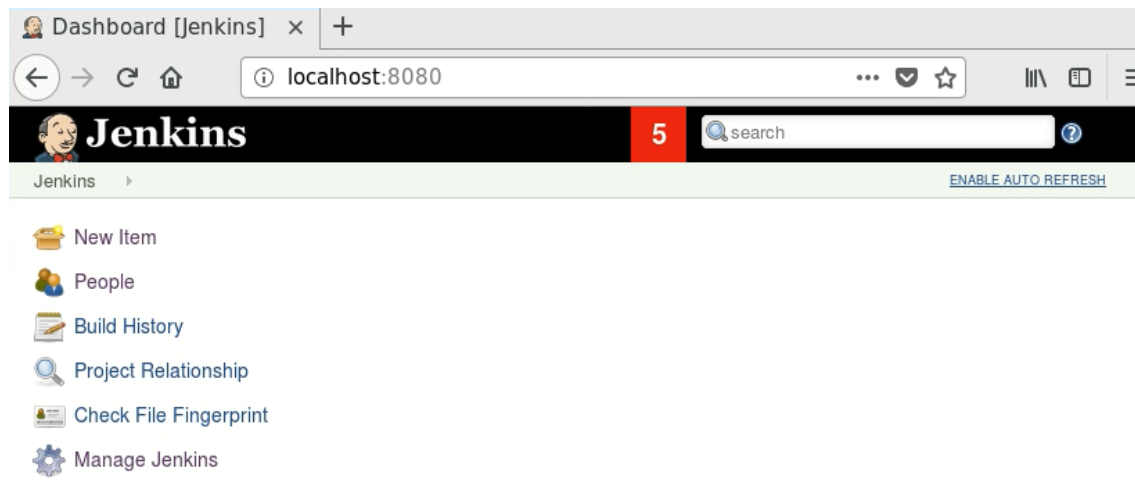
Prepared By:  
avinash.patel@wipro.com



## 1. Start Jenkins Service & Setup basic tools










```
osgdev@TG-DevOps-OS004:~$ service jenkins status
â–¶ jenkins.service - LSB: Start Jenkins at boot time
Loaded: loaded (/etc/init.d/jenkins; bad; vendor preset: enabled)
Active: active (exited) since Thu 2018-03-29 13:21:06 IST; 2 weeks 6 days ago
Docs: man:systemd-sysv-generator(8)
Process: 1665 ExecStart=/etc/init.d/jenkins start (code=exited, status=0/SUCCESS)
Tasks: 0
Memory: 0B
CPU: 0
```

You can access Jenkins on the browser using the URL: <http://localhost:8080/>



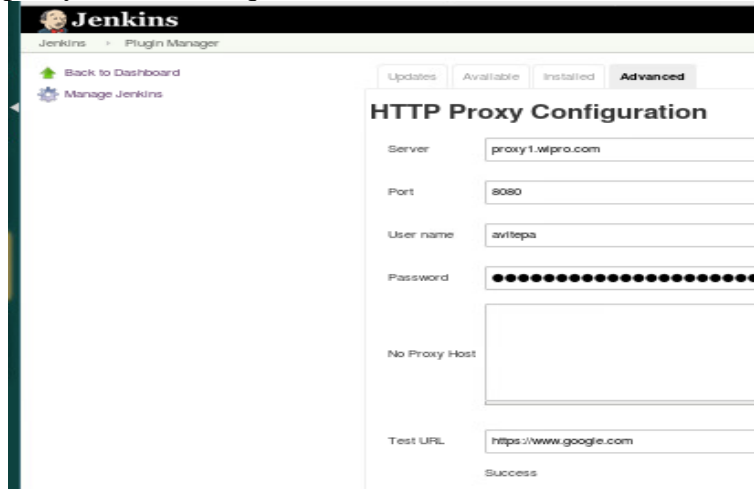
## 2. Manage Jenkins – Plugins, Proxy Setting & Tool Configurations (Java & Git)

Restore the previous version of Jenkins [Downgrade to 2.89.2](#)

-  [Configure System](#)  
Configure global settings and paths.
-  [Configure Global Security](#)  
Secure Jenkins; define who is allowed to access/use the system.
-  [Configure Credentials](#)  
Configure the credential providers and types
-  [Global Tool Configuration](#)  
Configure tools, their locations and automatic installers.
-  [Reload Configuration from Disk](#)  
Discard 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. (updates available)
-  [System Information](#)  
Displays various environmental information to assist trouble-shooting.
-  [System Log](#)  
System log captures output from java.util.logging output related to Jenkins.
-  [Load Statistics](#)

✓ **Manage Plugins (Updates | installed | Available | Advanced)**

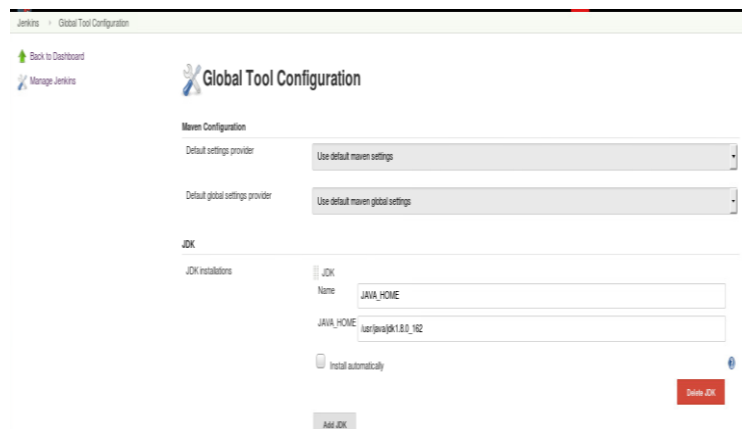
- Select required plugins and install
- Set proxy details (if required)



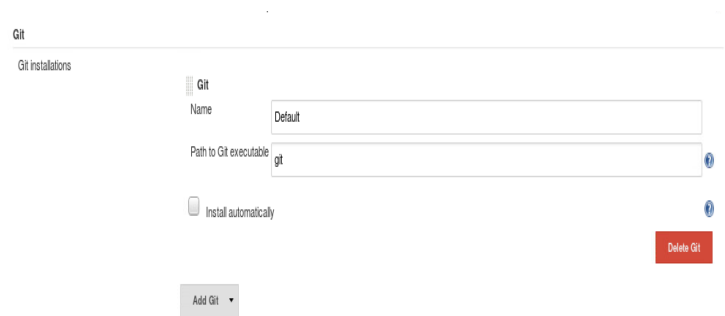
The screenshot shows the Jenkins 'HTTP Proxy Configuration' page. The 'Advanced' tab is selected. The 'Server' field is set to 'proxy1.wipro.com', 'Port' is '8080', 'User name' is 'avilopa', and 'Password' is masked with dots. The 'Test URL' field is 'https://www.google.com'. A 'Success' message is displayed at the bottom.

✓ **Global Tool Configuration**

- **Configure Git, JDK, Maven, Sonar Server...**

1. **JDK Configuration**

The screenshot shows the 'Global Tool Configuration' page in Jenkins, specifically the 'JDK' section. Under 'JDK installations', there is a table with one entry: 'Name' is 'JAVA\_HOME' and 'JDK HOME' is 'usr/javajdk1.8.0\_162'. There is an 'Add JDK' button at the bottom left and a 'Delete JDK' button at the bottom right.

2. **Git Configuration**

The screenshot shows the 'Global Tool Configuration' page in Jenkins, specifically the 'Git' section. Under 'Git installations', there is a table with one entry: 'Name' is 'Default' and 'Path to Git executable' is 'git'. There is an 'Add Git' button at the bottom left and a 'Delete Git' button at the bottom right.

### 3. Create Jenkins Job (Freestyle)

Once Jenkins is Up & Running

- ✓ Create Freestyle Jenkins Job
- ✓ Configure SCM (Git/Gitlab)
- ✓ Construct Build processing steps within "Build" section
  - Construct shell commands to do build process (Java project)
- ✓ Run with "Build Now"
- ✓ Observe Build History and console output

The screenshot shows the 'Enter an item name' form in Jenkins. At the top is a text input field for the item name, with a 'Required field' label below it. Below the input field are several project type options, each with an icon and a description:

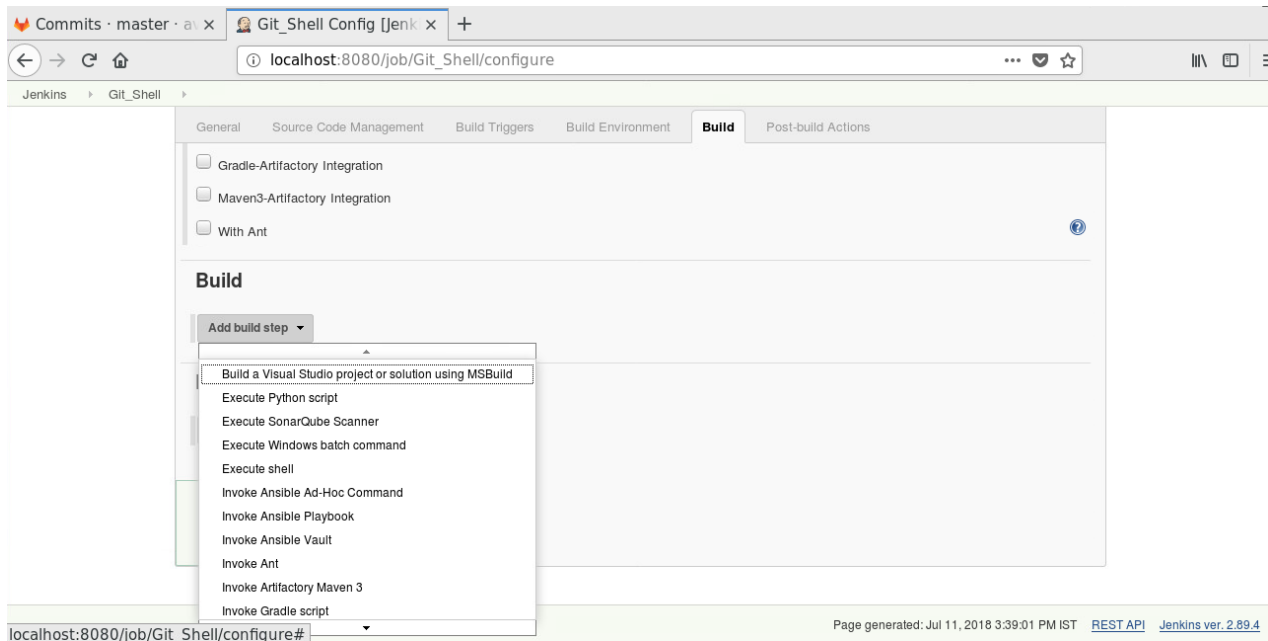
- Freestyle project**: This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Maven project**: Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.
- Pipeline**: Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**: Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**: Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- GitHub Organization**: Scans a GitHub organization (or user account) for all repositories matching some defined markers.

At the bottom, there is a 'Multibranch Pipeline' option which is partially obscured by a green 'OK' button.

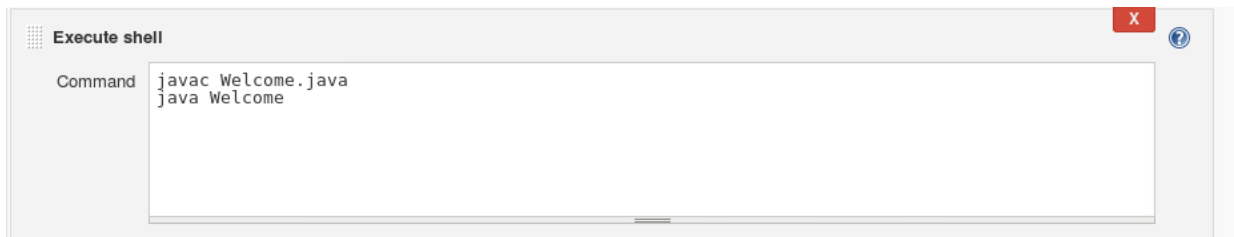
### Configure Git Repository Path

The screenshot shows the 'Source Code Management' configuration page in Jenkins. The 'Git' radio button is selected under the 'Source Code Management' section. The 'Repositories' section contains a 'Repository URL' field with the value '/home/osgdev/R2/L2R'. Below this is a red error message: 'Please enter Git repository.' The 'Credentials' dropdown menu is set to '- none -'. There is an 'Add' button next to the credentials dropdown. The 'Branches to build' section has a 'Branch Specifier (blank for \'any\')' field with the value '\*/master'. There is a red 'X' icon next to the branch specifier field. At the bottom, there are 'Save' and 'Apply' buttons, and a dropdown menu set to '(Auto)'.

## Configure Build – Execute Shell with commands – Apply and Save

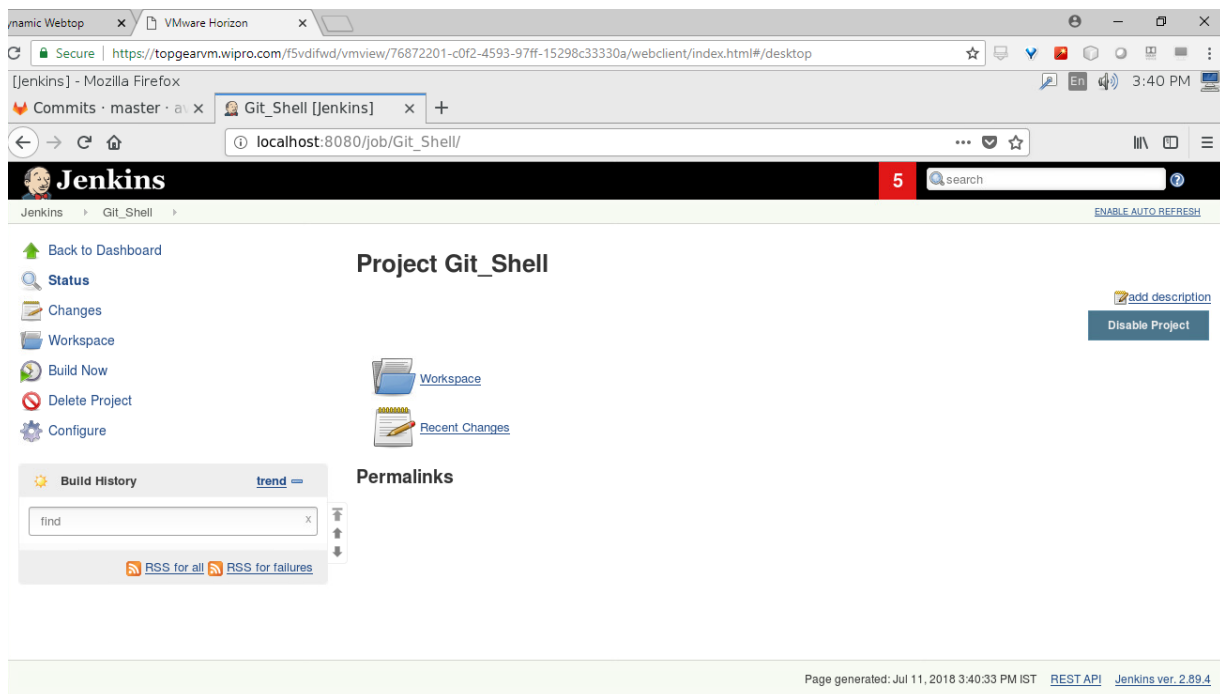


The screenshot shows the Jenkins 'Git\_Shell Config' page for a job named 'Git\_Shell'. The 'Build' tab is selected, and a dropdown menu is open under 'Add build step'. The menu lists various build actions, including 'Execute shell', which is highlighted. The page footer indicates it was generated on Jul 11, 2018, at 3:39:01 PM IST, using Jenkins ver. 2.89.4.

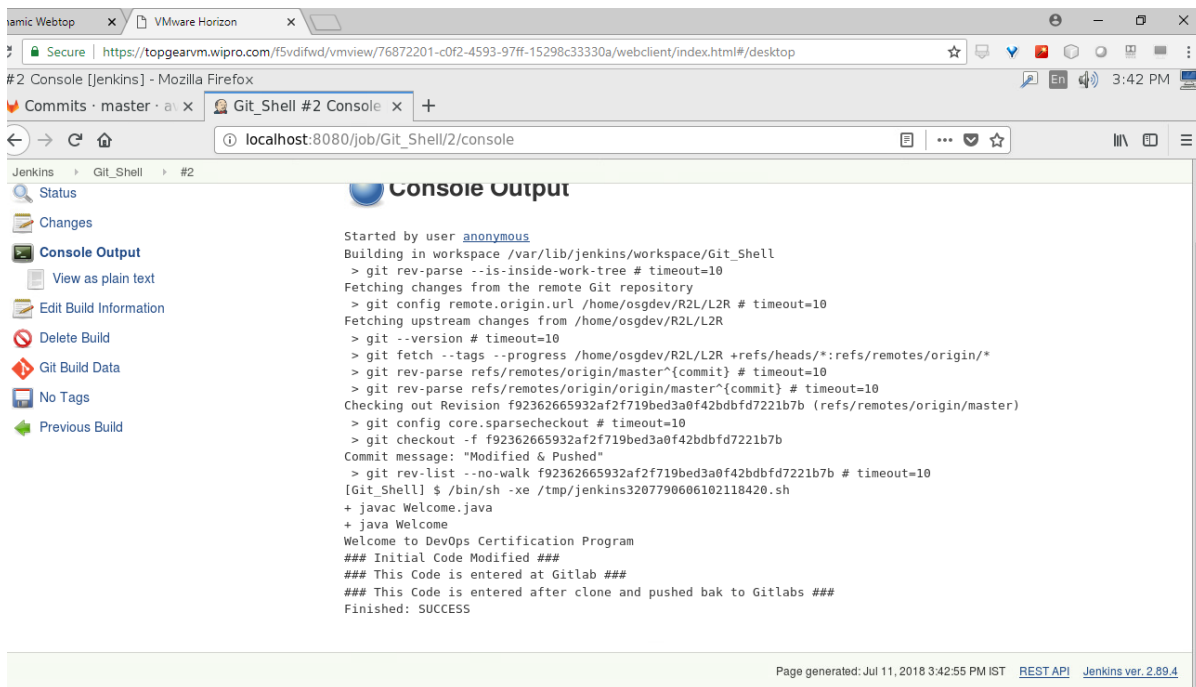
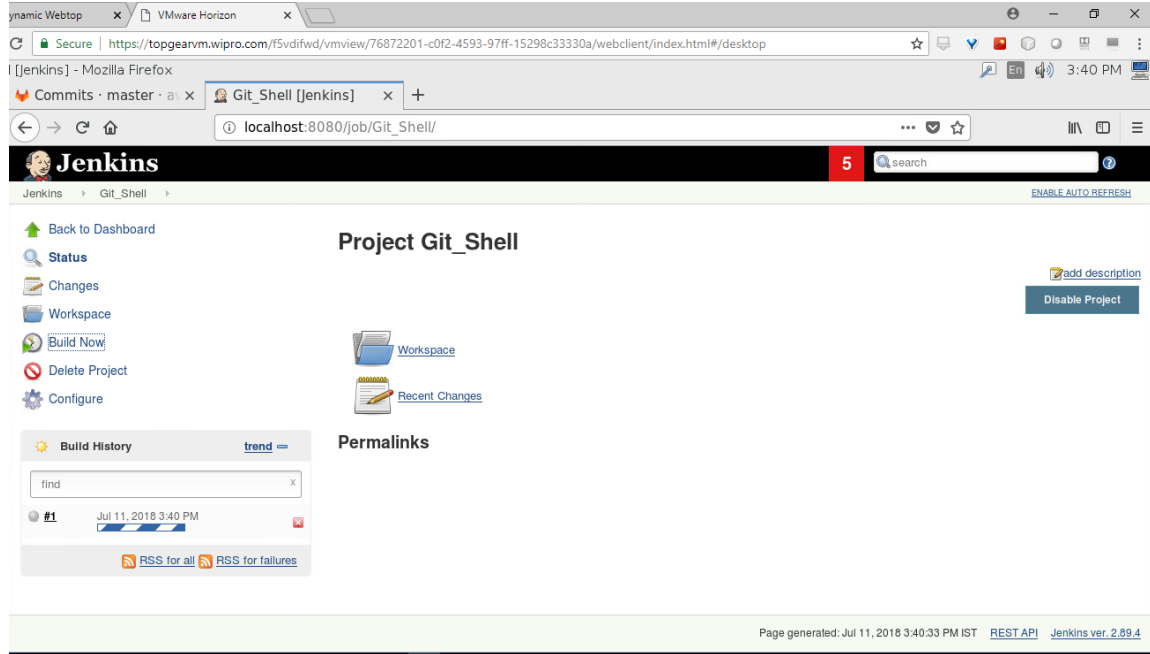


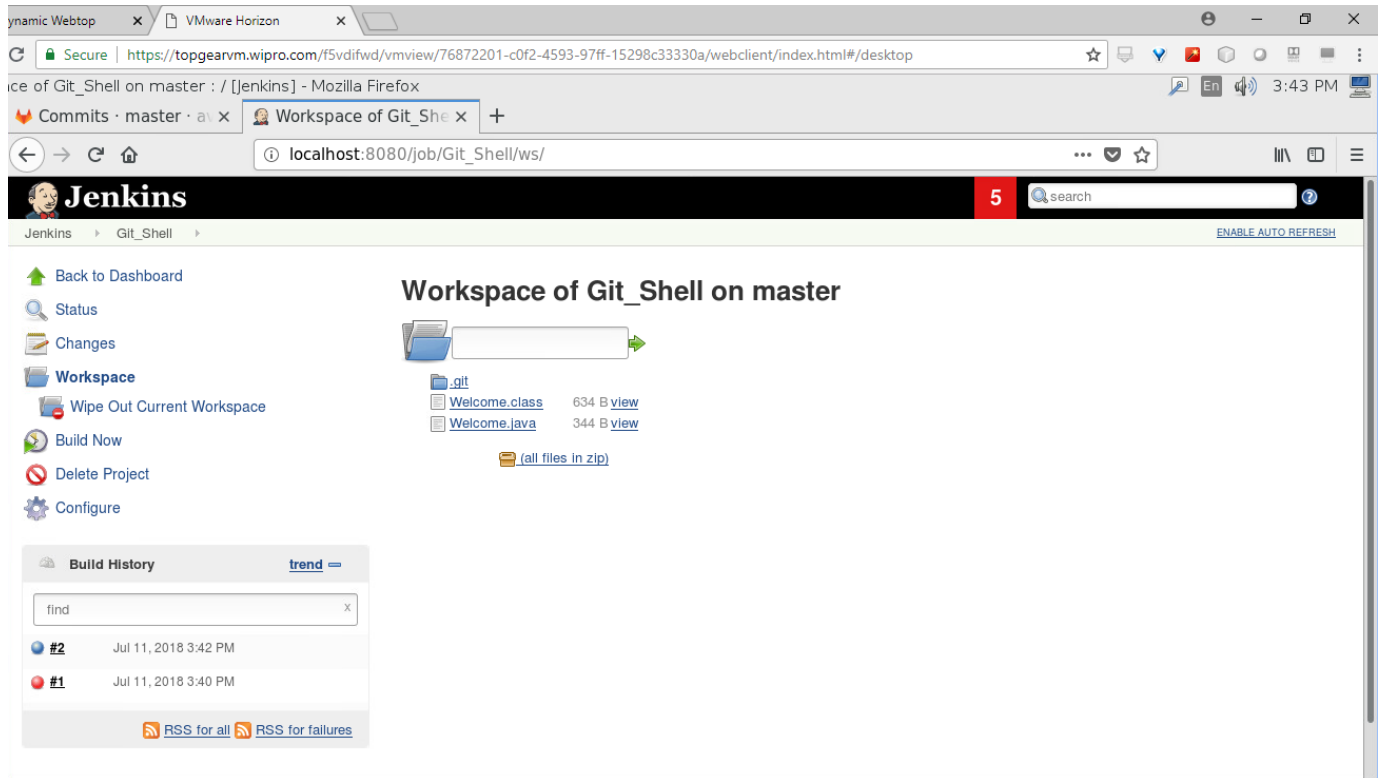
The screenshot shows the 'Execute shell' configuration step. The 'Command' field contains the following text:

```
javac Welcome.java  
java Welcome
```



The screenshot shows the Jenkins 'Project Git\_Shell' dashboard. The page includes a sidebar with navigation links: Back to Dashboard, Status, Changes, Workspace, Build Now, Delete Project, and Configure. The main content area displays the project name 'Project Git\_Shell', a 'Workspace' icon, and a 'Recent Changes' icon. Below these, there is a 'Build History' section with a search bar and a 'Permalinks' section. The page footer indicates it was generated on Jul 11, 2018, at 3:40:33 PM IST, using Jenkins ver. 2.89.4.

**Select Build Now to initiate the process – View Console Output & Workspace**

**Note:**

- Change SCM configuration to access central repository URL (Gitlab) and observe execution
- Configure more than one Repository and construct build command accordingly to observe execution