AVF Jenkins Setup

Knowledge Transfer Document

4/13/2018

[Type the company name]

Ajoy Sinha

AVF Setup Document

Document identifier:

Location:

Editor:

Contributors:

Abstract:

Status:

Table of Contents

[1 Introduction 4](#_Toc511664412)

[1.1 Terminology 4](#_Toc511664413)

[2 Jenkins Setup 5](#_Toc511664414)

[2.1 Setup Jenkins 5](#_Toc511664415)

[ Login into AMI instance using Putty 5](#_Toc511664416)

[ Run all command using sudo 5](#_Toc511664417)

[ Run yum update 5](#_Toc511664418)

[ Get Jenkins repository using below command 5](#_Toc511664419)

[ Get Jenkins repository key 5](#_Toc511664420)

[ Install jenkins package 5](#_Toc511664421)

[ Start Jenkins and register it as service 5](#_Toc511664422)

[ Open your browser and navigate to http://IP:8080. You will see Jenkins dashboard. 6](#_Toc511664423)

[ You will find the initialAdminPassword under 6](#_Toc511664424)

[ /var/lib/jenkins/secrets/initialAdminPassword 6](#_Toc511664425)

[ Open the file and copy the password and pest it under Administrator Password field 6](#_Toc511664426)

[ Select continue 6](#_Toc511664427)

[2.2 Setup User 6](#_Toc511664428)

[ Login into Jenkins using admin userid and password 6](#_Toc511664429)

[ Follow this path Manage Jenkins > Manage Users > Create User 6](#_Toc511664430)

[ Add user details and save the user 6](#_Toc511664431)

[ Add user name and password to login 6](#_Toc511664432)

[2.3 Setup Plugins 7](#_Toc511664433)

[ Login into Jenkins using admin userid and password 7](#_Toc511664434)

[ Follow this path Manage Jenkins > Manage Plugins 7](#_Toc511664435)

[ Search plugin using filter 7](#_Toc511664436)

[ Check necessary plugins 7](#_Toc511664437)

[ Click “install Without Restart” 7](#_Toc511664438)

[ List of Plugin to be installed 7](#_Toc511664439)

[2.4 Setup Distributed Build (Master Slave) Configuration 9](#_Toc511664440)

[3 Create AVF Application Project 15](#_Toc511664441)

[3.1 Create Project 15](#_Toc511664442)

[4 Create AVF Integration Test Project 17](#_Toc511664443)

[4.1 Create Project 17](#_Toc511664444)

[5 Build Projects 18](#_Toc511664445)

[5.1 Build AVF Application Project 18](#_Toc511664446)

[5.2 Build AVF Integration Test Project 19](#_Toc511664447)

[5.3 See Test Result 20](#_Toc511664448)

[6 References 21](#_Toc511664449)

[6.1 Normative 21](#_Toc511664450)

[Appendix A. 22](#_Toc511664451)

# Introduction

## Terminology

# Jenkins Setup

## Setup Jenkins

* Launch an Amazon Linux instance using Amazon Linux AMI (use console to login and start the instance)

### Login into AMI instance using Putty

### Run all command using sudo

### Run yum update

### Get Jenkins repository using below command

sudo wget -O /etc/yum.repos.d/jenkins.repo <http://pkg.jenkins-ci.org/redhat-stable/jenkins.repo>

### Get Jenkins repository key

sudo rpm --import [http://pkg.jenkins-ci.org/redhat-stable/jenkins-ci.org.key](https://t.umblr.com/redirect?z=http%3A%2F%2Fpkg.jenkins-ci.org%2Fredhat-stable%2Fjenkins-ci.org.key&t=MjU0MzM4OTZhYWE3NTFkNGU4ZmUxNDIzYmRiYWUyNGZiYTNmYTdjMSw3UE5YQVZNeg%3D%3D&b=t%3AfVnVieO0zHmDiIU7XeIv6w&p=http%3A%2F%2Fsanketdangi.com%2Fpost%2F62715793234%2Finstall-configure-jenkins-on-amazon-linux&m=1)

### Install jenkins package

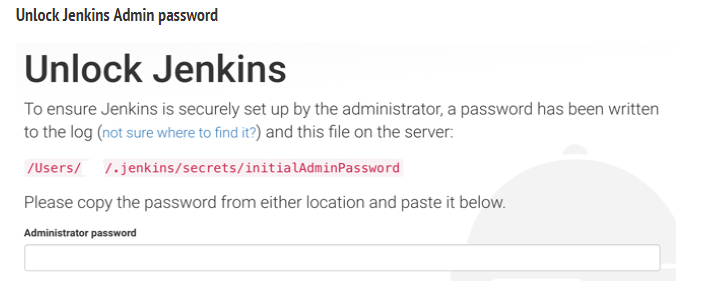
sudo yum install jenkins

### Start Jenkins and register it as service

sudo service jenkins start

sudo chkconfig jenkins on

### Open your browser and navigate to http://IP:8080. You will see Jenkins dashboard.



### You will find the initialAdminPassword under

### /var/lib/jenkins/secrets/initialAdminPassword

### Open the file and copy the password and pest it under Administrator Password field

### Select continue

## Setup User

### Login into Jenkins using admin userid and password

### Follow this path Manage Jenkins > Manage Users > Create User

### Add user details and save the user

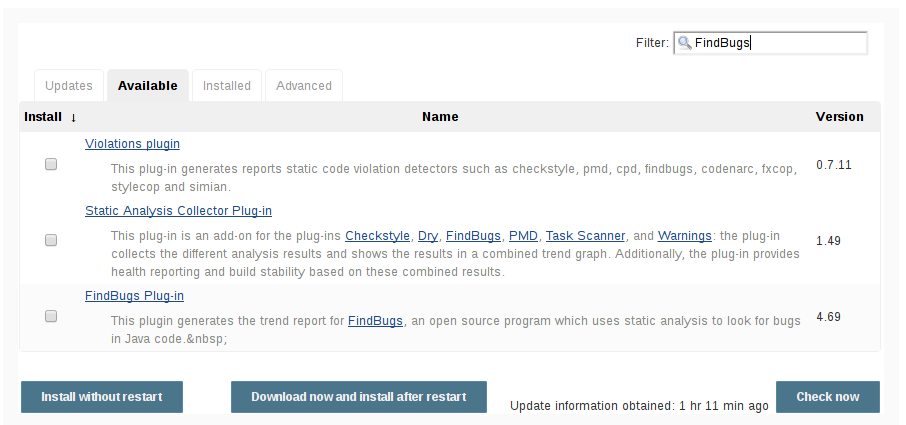
### Add user name and password to login

## Setup Plugins

### Login into Jenkins using admin userid and password

### Follow this path Manage Jenkins > Manage Plugins

Under the Available tab, plugins available for download from the configured Update Center can be searched and considered:

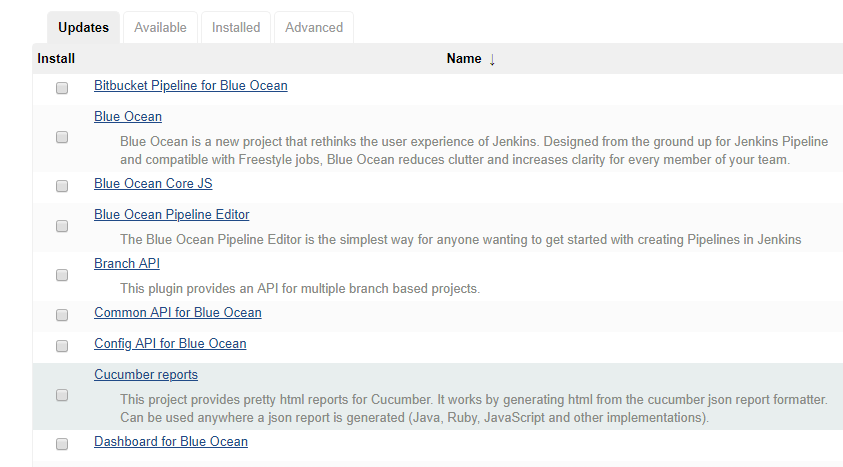


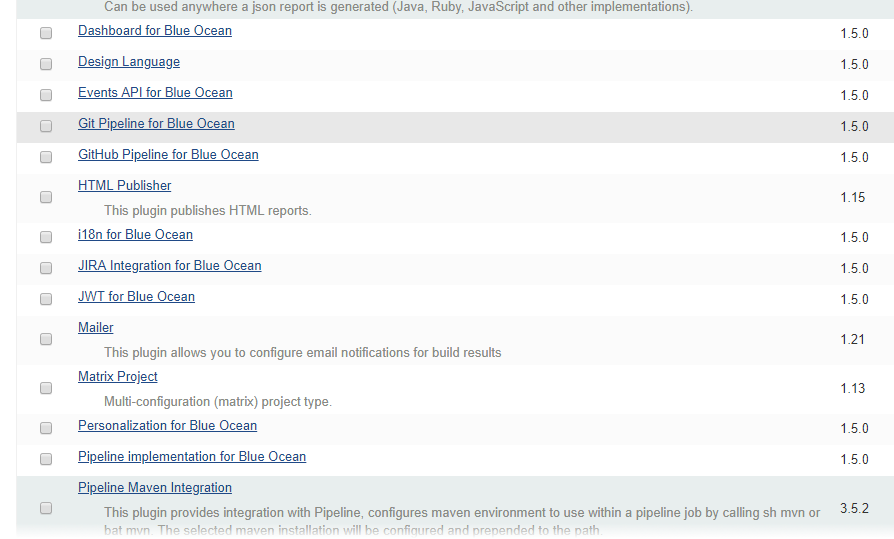
### Search plugin using filter

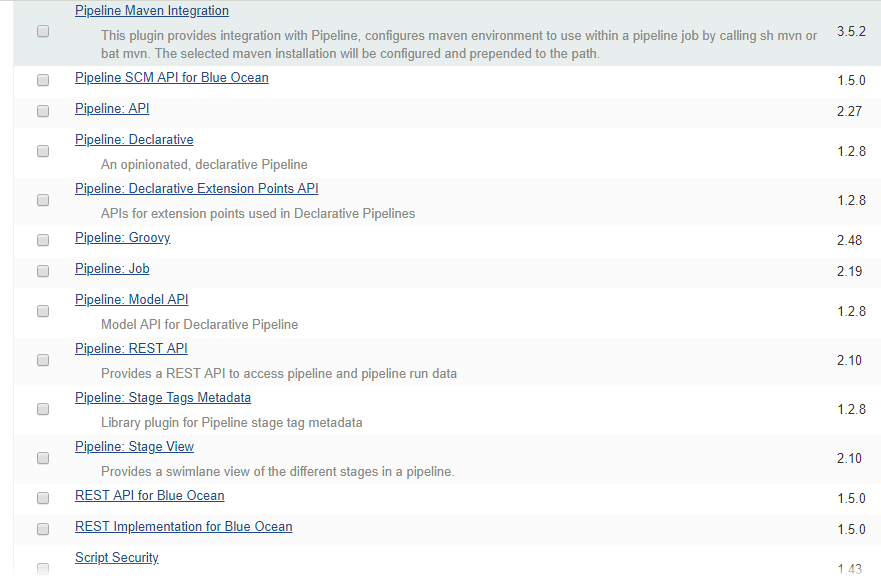
### Check necessary plugins

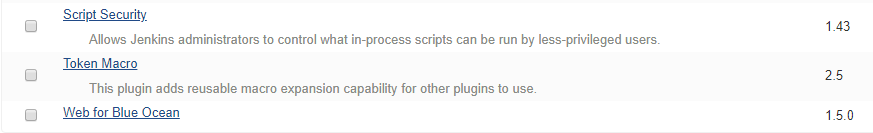
### Click “install Without Restart”

### List of Plugin to be installed









## Setup Distributed Build (Master Slave) Configuration



Setup SSH Key to provide access

Create Public / Private key from Jenkins Master as below

Run sudo ssh-keygen -t rsa -C [ajoy.kumarsinha@atos.net](mailto:ajoy.kumarsinha@atos.net) from command prompt

<< Use email id as required >>

Following will be displayed

Generating public/private rsa key pair.

Enter file in which to save the key (/root/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /root/.ssh/id\_rsa.

Your public key has been saved in /root/.ssh/id\_rsa.pub.

The key fingerprint is:

<< Digest>>

The key's randomart image is:

+---[RSA 2048]----+

| |

| |

| . . |

| a . \* + |

| S o P E .|

| + B B =.|

| . B.O = \*|

| . +.O=B \*.|

| o .\*B== .|

+----[SHA256]-----+

You will find the generated keys under (/root/.ssh)

Once complete, To provide access you can either manually add the contents of the ~/.ssh/rsa.pub file to the file ~/.ssh/authorized\_keys file on the remote server (In Jenkins Slave)

Setup Jenkins Slave Folders

1. Login into Jenkins Slave (Application Server in our case) using Putty.
2. Create a folder called jenkins\_slave under /opt [This Folder will be place of execution of jobs]
3. Provide full access to this folder [Or you can provide specific access if required, But this need to be access and writable from Jenkins Master Node]
4. Login into Jenkins Server using web interface

Create following files and save under /opt/scripts/ of Jenkins Slave

1. deployBlueMobileCore.sh

Content of following file

* 1. /opt/EAP-7.1.0/bin/jboss-cli.sh --connect --controller=<<App Server IP>>:<<Port>> --user=<<UserName>> --password=<<Password>> --command="deploy --force /opt/jenkins\_slave/workspace/AVFApplications/BlueMobileCore/BlueMobileEAR/target/BlueMobile.ear"

1. deployBlueMobileExtranet.sh

Content of following file

* 1. /opt/EAP-7.1.0/bin/jboss-cli.sh --connect --controller=<<App Server IP>>:<<Port>> --user=<<UserName>> --password=<<Password>> --command="deploy --force /opt/jenkins\_slave/workspace/AVFApplications/ AVFExtranet/AVFExtranetEAR/target/AVFExtranet.ear"

1. deployBlueMobileTestHarness.sh

Content of following file

* 1. /opt/EAP-7.1.0/bin/jboss-cli.sh --connect --controller=<<App Server IP>>:<<Port>> --user=<<UserName>> --password=<<Password>> --command="deploy --force /opt/jenkins\_slave/workspace/AVFApplications/ TestHarness/TestHarnessEAR/target/BlueMobileTestHarness.ear"

1. deployBlueMobileAdminConsole.sh

Content of following file

* 1. /opt/EAP-7.1.0/bin/jboss-cli.sh --connect --controller=<<App Server IP>>:<<Port>> --user=<<UserName>> --password=<<Password>> --command="deploy --force /opt/jenkins\_slave/workspace/AVFApplications/ BlueMobileAdminConsole/BlueMobileAdminEAR/target/BlueMobileAdminConsole.ear"

1. deployBlueMobileScheduler.sh

Content of following file

* 1. /opt/EAP-7.1.0/bin/jboss-cli.sh --connect --controller=<<App Server IP>>:<<Port>> --user=<<UserName>> --password=<<Password>> --command="deploy --force /opt/jenkins\_slave/workspace/AVFApplications/ Scheduler/SchedulerEAR/target/BlueMobileScheduler.ear"

1. Setup Node in Jenkins Master
2. Go to Manage Jenkins > Manage Nodes.
3. New Node --> Enter Node Name.
4. Select Permanent Agent
5. Click OK
6. On Details page enter following information as below

Name: Node Name

Description: Some Specific Description for the Slave

# of executors: How many Executors you want to set (For SIT 1 is fine)

Remote root directory: Path mentioned in step 2 [/opt/jenkins\_slave]

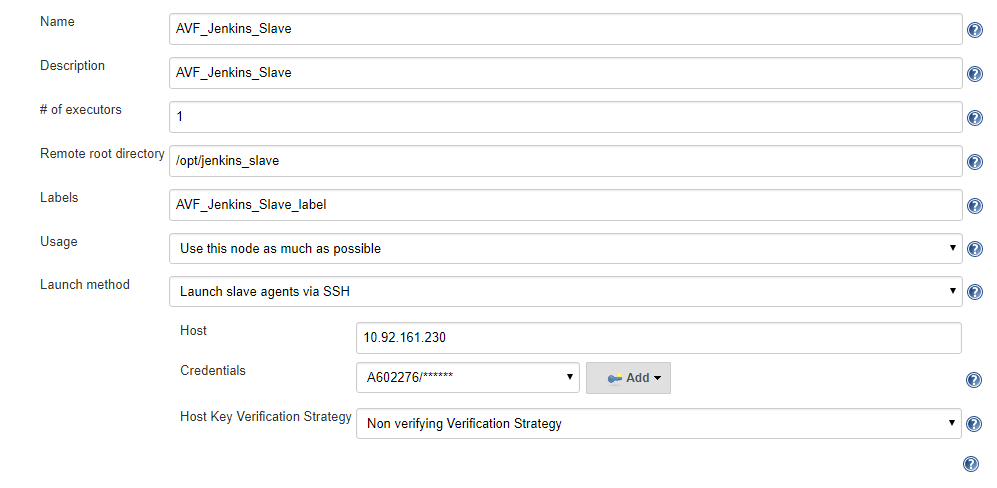
Lables: Add Label to define identification group

Usage: Use this node as much as possible

Launch Method: Launch slave agent via ssh

Host: App Server IP

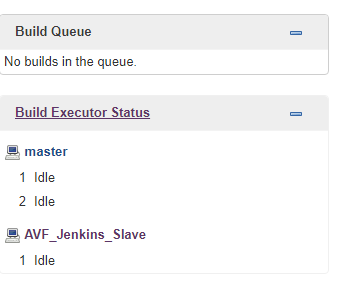
Credential : Userid / Password to connect



Add Node Properties



1. Save Node Details
2. Click on the Launch button to launch agent
3. On successful connection and Launch you will find following node as below



# Create AVF Application Project

## Create Project

1. Login into Jenkins Portal
2. Click on Jenkins > New Item
3. Enter Project Name “AVFApplications” [Avoid to enter Name With Space]
4. Select “Maven Project” and click “OK”

This will create a Jenkins Project called “AVFApplications”

1. Enter Description as “Build and Deploy AVF Application in JBOSS Server”
2. In Section “Maven Info Plugin Configuration”

Add “GitHub Project” value: [**https://github.com/atosorigin/AVF2.0/**](https://github.com/atosorigin/AVF2.0/)

1. In Section “Maven Info Plugin Configuration”

Add “Restrict where this project can be run” value as AVF\_Jenkins\_Slave\_label in Label Expression. This label name is the same name of we have given for Node

1. In Section “Source Code Management” select « Git »

Add values to Repositories as below

1. Repository URL : <https://github.com/atosorigin/AVF2.0.git>
2. Credential : <<As useful, which has access on Repository>>
3. Branch Specifier : \*/master
4. Repository browser : Auto
5. No Additional Behaviours
6. In Section “Build”
7. Root POM will be pom.xml
8. Goals and options will be “clean install -pl !BlueMobileIntegrationTest”
   1. This will ensure not to build IntegrationTest project
9. In Section “Post Steps”

**Select “**Execute Shell**”**

Enter following command

/opt/scripts/deployBlueMobileCore.sh

/opt/scripts/deployBlueMobileExtranet.sh

/opt/scripts/deployBlueMobileTestHarness.sh

/opt/scripts/deployBlueMobileAdminConsole.sh

/opt/scripts/deployBlueMobileScheduler.sh

1. Click Save

# Create AVF Integration Test Project

## Create Project

1. Login into Jenkins Portal
2. Click on Jenkins > New Item
3. Enter Project Name “AVFIntegrationTest” [Avoid to enter Name With Space]
4. Select “Maven Project” and click “OK”

This will create a Jenkins Project called “AVFIntegrationTest”

1. **Enter Description as “**AVF Integration Test : Used to run integration test on System Test Environment**”**
2. In Section “Maven Info Plugin Configuration”

Add “GitHub Project” value: [**https://github.com/atosorigin/AVF2.0/**](https://github.com/atosorigin/AVF2.0/)

1. In Section “Maven Info Plugin Configuration”

Add “Restrict where this project can be run” value as AVF\_Jenkins\_Slave\_label in Label Expression. This label name is the same name of we have given for Node

1. In Section “Source Code Management” select « Git »

Add values to Repositories as below

1. Repository URL : <https://github.com/atosorigin/AVF2.0.git>
2. Credential : <<As useful, which has access on Repository>>
3. Branch Specifier : \*/master
4. Repository browser : Auto
5. No Additional Behaviours
6. In Section “Build”
7. Root POM will be pom.xml
8. Goals and options will be “clean install -pl BlueMobileIntegrationTest”
   1. This will ensure to build IntegrationTest project only
9. Click Save

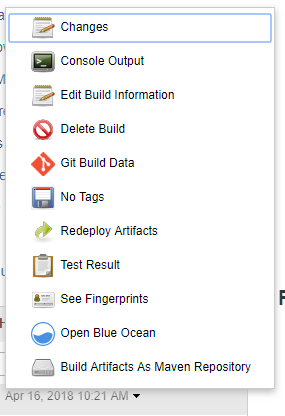
# Build Projects

## Build AVF Application Project

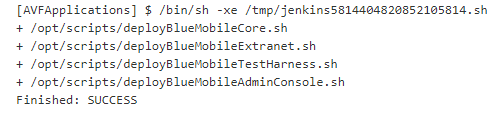
1. Login into Jenkins Portal
2. Click on Jenkins >
3. Click  button of following



1. Project Will Start Building
2. And log can been seen on “Console Output”



1. On Success of Build and Deploy Console will display following



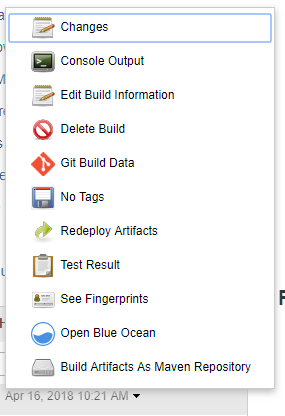
1. And log can been seen on “Console Output”

## Build AVF Integration Test Project

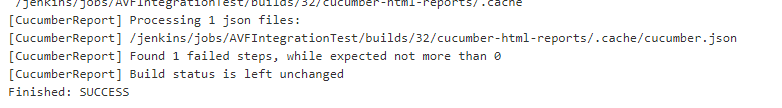
1. Login into Jenkins Portal
2. Click on Jenkins >
3. Click  button of following



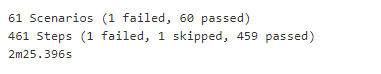
1. Project Will Start Building
2. And log can been seen on “Console Output”



1. On Success of Build Console will display following



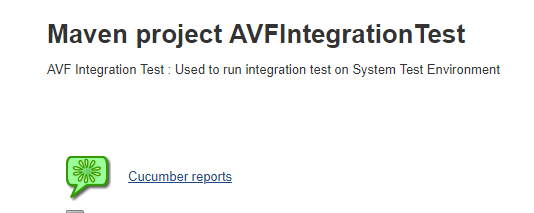
1. On Success of Test Console will display following



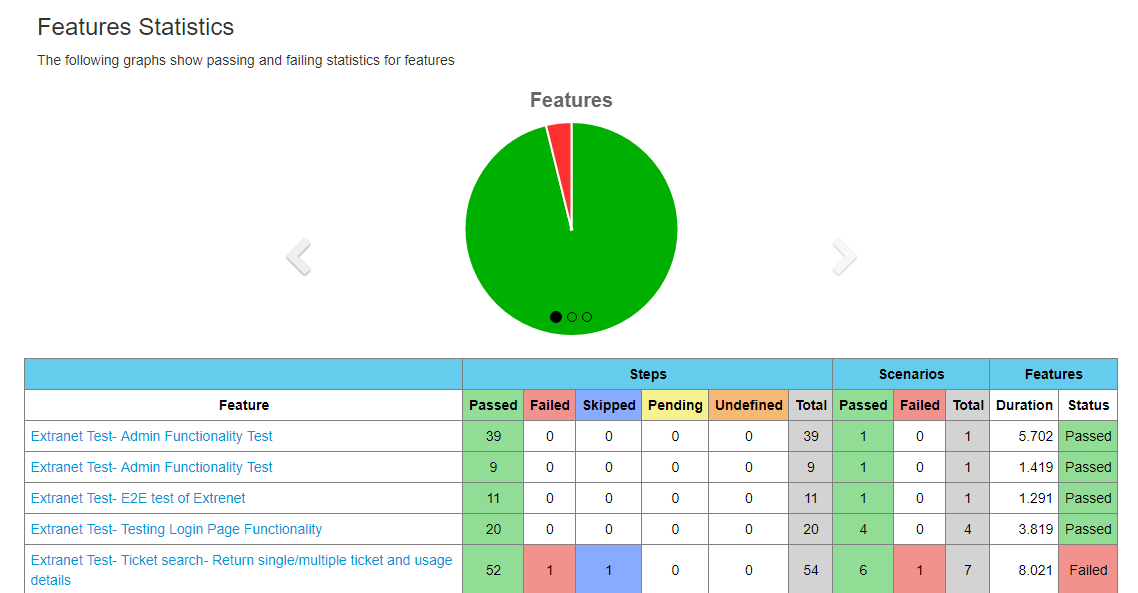
1. Test Result can be seen in Cucumber Reports also

## See Test Result

1. Login into Jenkins Portal
2. Click on Jenkins > AVF Integration Test
3. Click on Cucumber reports as shown below



1. Report will be displayed as below



# References

## Normative