

Module 4: Setting Up Your Build Jobs and Security

Case Study Solution

edureka!

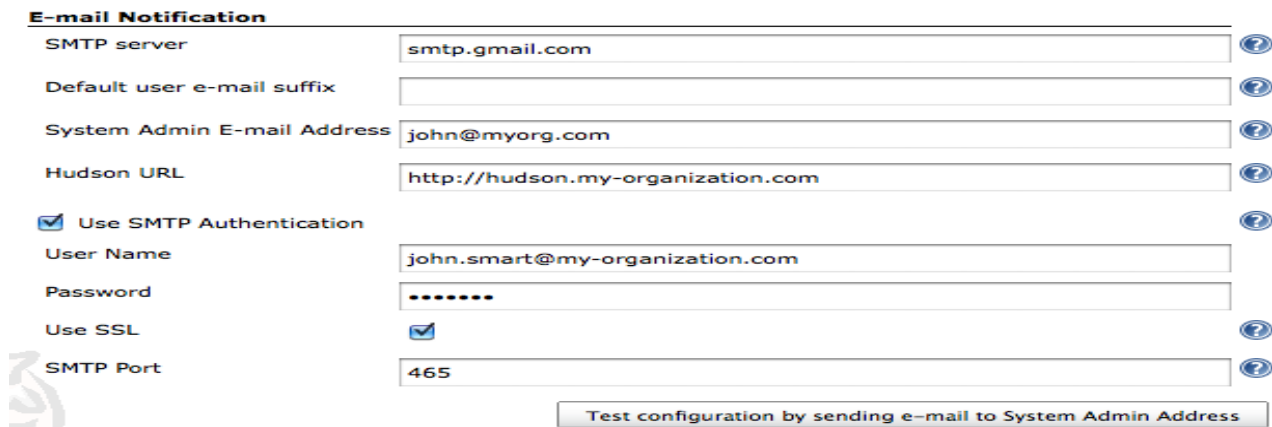
edureka!

© Brain4ce Education Solutions Pvt. Ltd.

Case study - Solution

Step 1: Install “Email Extension Plugin” and “Extensible Choice Parameters Plugin” by navigating Manage Jenkins -> Advanced.

Step 2: Configure the Extended Email notification under Manage Jenkins -> Configure System.



E-mail Notification

SMTP server: smtp.gmail.com

Default user e-mail suffix:

System Admin E-mail Address: john@myorg.com

Hudson URL: http://hudson.my-organization.com

☒ Use SMTP Authentication

User Name: john.smart@my-organization.com

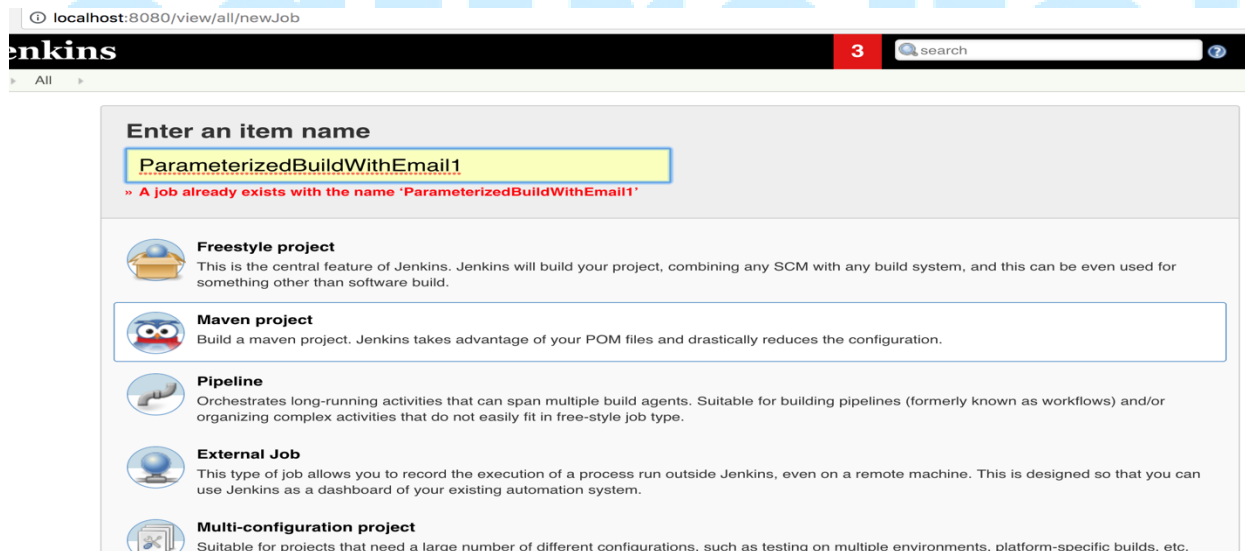
Password:

Use SSL: ☒

SMTP Port: 465

Test configuration by sending e-mail to System Admin Address

Step 3: Create a Jenkins “Maven Project” and name the project.



localhost:8080/view/all/newJob

jenkins

3 search

Enter an item name

ParameterizedBuildWithEmail1

» A job already exists with the name 'ParameterizedBuildWithEmail1'

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 agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Step 4: Enable “This Project is parameterized”, choose “string” parameter and update all the details.

Module 4: Setting Up Your Build Jobs and Security

The screenshot shows the Jenkins job configuration page for 'ParameterizedBuildWithEmail' in the 'General' tab. The browser address bar shows 'localhost:8080/job/ParameterizedBuildWithEmail/configure'. The 'Description' field contains the text: 'This Job takes a parameter from the user, runs an other build before and passes the parameters to the current job.' Below this, there are checkboxes for 'Discard old builds', 'GitHub project', and 'Permission to Copy Artifact'. Under 'Rebuild options', there are checkboxes for 'Rebuild Without Asking For Parameters', 'Disable Rebuilding for this job', and 'This project is parameterised' (which is checked). A 'String Parameter' section is visible with the following details: Name: 'Version', Default Value: '1', and Description: 'The version of the build'. There are 'Save' and 'Apply' buttons at the bottom left.

Step 5: Enable “Git” in Source Code Management and Provide the repository url (<https://github.com/edureka-devops/jenkins-demo.git>)

The screenshot shows the Jenkins job configuration page for 'ParameterizedBuildWithEmail' in the 'Source Code Management' tab. The browser address bar shows 'localhost:8080/job/ParameterizedBuildWithEmail/configure'. The 'Source Code Management' section has 'Git' selected. Under 'Repositories', the 'Repository URL' is 'https://github.com/vathsalahn/jenkins-demo.git' and 'Credentials' is set to '- none -'. There is an 'Add Repository' button. Under 'Branches to build', the 'Branch Specifier (blank for \'any\')' is set to '*/master' and there is an 'Add Branch' button. The 'Repository browser' is set to '(Auto)'. There is an 'Add' button for 'Additional Behaviours'. At the bottom, there are 'Save' and 'Apply' buttons. A note at the bottom says 'Trigger builds remotely (e.g., from scripts)'.

Step 6: Enable the Build Environment to “Delete workspace before build starts” and “Abort the build if it's stuck”

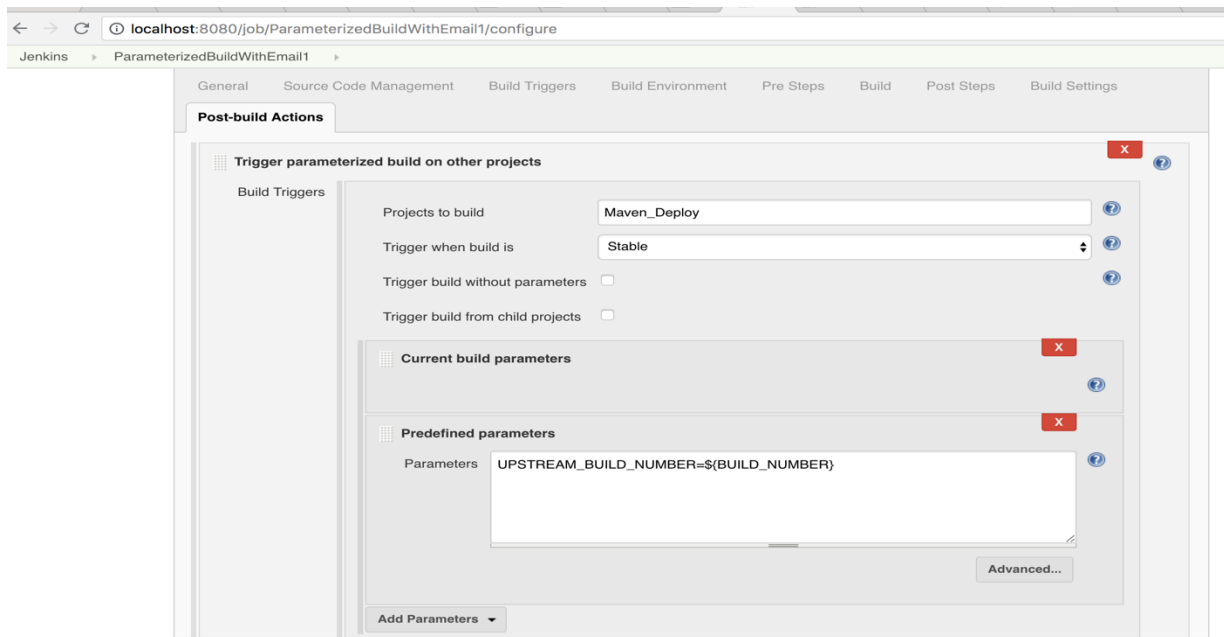
The screenshot shows the Jenkins configuration page for a job named 'Jenkins-Fibonacci'. The 'Build Environment' tab is selected. Under 'Build Environment', the options 'Delete workspace before build starts' and 'Abort the build if it's stuck' are checked. The 'Time-out strategy' is set to 'Absolute' with a 'Timeout minutes' of 3. There are also checkboxes for 'Add timestamps to the Console Output' and 'With Ant'. The 'Build' section shows a single step 'Execute shell' with the command: `chmod +x Fibonacci.sh` and `$(WORKSPACE)/Fibonacci.sh > output.txt`. 'Save' and 'Apply' buttons are at the bottom.

Step 7: In Build section execute the maven script to build the artifact and update the goals to “clean install”.

The screenshot shows the Jenkins configuration page for a job named 'ParameterizedBuildWithEmail1'. The 'Build' tab is selected. Under 'Build', the 'Root POM' is set to 'MavenProject/pom.xml' and the 'Goals and options' are set to 'clean install'. There are 'Advanced...' and 'Add pre-build step' buttons. The 'Post Steps' section has radio buttons for 'Run only if build succeeds', 'Run only if build succeeds or is unstable', and 'Run regardless of build result', with the last one selected. There is also an 'Add post-build step' button. The 'Build Settings' section is partially visible at the bottom.

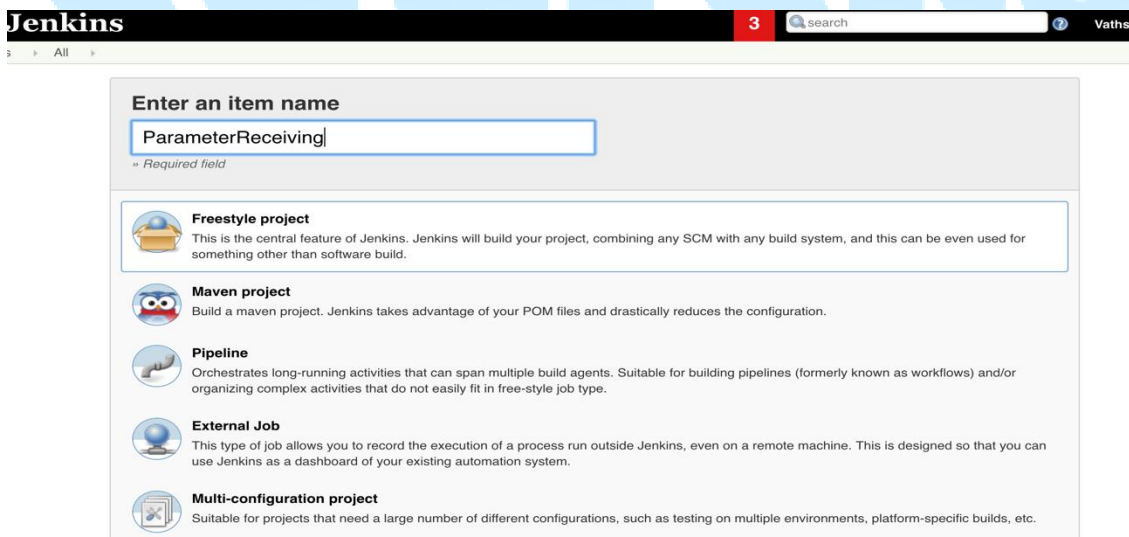
Step 8: Update the Post-build Actions to send these parameters to another downstream jobs. Install “Trigger parameters” plugin in Jenkins and update the parameters to be passed on to the

downstream project. Send the current build parameters (Version) defined in the previous step and also the other environment variables using Predefined parameters.



Step 9: Apply and save the job.

Step 10: Create a downstream Job to accept the parameters from the upstream job. Create a freestyle project.



Step 11: Enable “This project is parameterized” and access the variables to be used from the upstream build.

Module 4: Setting Up Your Build Jobs and Security

The screenshot shows the Jenkins configuration page for a job named 'Maven_Deploy'. The 'General' tab is selected. At the top, there is a checkbox 'This project is parameterised' which is checked. Below this, there are two 'String Parameter' sections. The first parameter has the name 'UPSTREAM_BUILD_NUMBER', an empty 'Default Value' field, and an empty 'Description' field. The second parameter has the name 'Version', an empty 'Default Value' field, and an empty 'Description' field. Each parameter section includes a 'Preview' link and a 'Trim the string' checkbox.

Step 12: Execute a python script to access these parameters and trigger an email notification to the users. Use the repository url given below:

<https://github.com/edureka-devops/jenkins-demo.git>

The screenshot shows the Jenkins configuration page for the same job, now with the 'Source Code Management' tab selected. At the top, there is a checkbox 'Restrict where this project can be run' which is unchecked. Below this, the 'Source Code Management' section is expanded. Under 'Repositories', the 'Git' option is selected. The 'Repository URL' field contains 'https://github.com/vathsalahn/jenkins-demo.git'. The 'Credentials' dropdown is set to '- none -'. Below this, the 'Branches to build' section has a 'Branch Specifier (blank for \'any\')' field set to '*/master'. The 'Repository browser' dropdown is set to '(Auto)'. At the bottom, there is an 'Additional Behaviours' section with an 'Add' button. The 'Subversion' option is also visible but not selected.

Module 4: Setting Up Your Build Jobs and Security

The screenshot shows the 'Maven_Deploy' configuration page in the Jenkins web interface. The 'General' tab is selected. The 'Editable Email Notification' section is expanded, showing various settings for email notifications. The 'Project From' and 'Project Recipient List' fields are both set to 'vathsala.hn22@gmail.com'. The 'Project Reply-To List' field is set to '\$DEFAULT_REPLYTO'. The 'Content Type' is set to 'Default Content Type'. The 'Default Subject' is set to '\$DEFAULT_SUBJECT'. The 'Default Content' is set to '\$DEFAULT_CONTENT'. There are help icons (question marks) next to several fields.

localhost:8080/job/Maven_Deploy/configure

Maven_Deploy

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

Editable Email Notification

Disable Extended Email Publisher ☐

Allows the user to disable the publisher, while maintaining the settings

Project From vathsala.hn22@gmail.com

Project Recipient List vathsala.hn22@gmail.com

Project Reply-To List \$DEFAULT_REPLYTO

Comma-separated list of email address that should receive notifications for this project.

Content Type Default Content Type

Default Subject \$DEFAULT_SUBJECT

Default Content \$DEFAULT_CONTENT

The screenshot shows the 'Maven_Deploy' configuration page in the Jenkins web interface. The 'General' tab is selected. The 'Triggers' section is expanded, showing a list of triggers. The first trigger is 'Always'. The 'Send To' field is set to 'Recipient List'. There is an 'Add' button next to the 'Recipient List' field. There is an 'Advanced...' button. There is an 'Add Trigger' button. There is an 'Add post-build action' button. There are help icons (question marks) next to several fields.

1s Maven_Deploy

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

Post-send Script \$DEFAULT_POSTSEND_SCRIPT

Additional groovy classpath Add

Save to Workspace ☐

Triggers

Always

Send To Recipient List

Add

Advanced...

Add Trigger

Add post-build action

Step 13: Trigger the main upstream job and Upon triggering the job, the execution has to be success, downstream job has to be initialized and an email has to be triggered to the recipient list.

Module 4: Setting Up Your Build Jobs and Security

```
localhost:8080/job/ParameterizedBuildWithEmail1/8/console
jenkins > ParameterizedBuildWithEmail1 > #8

[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.439 s
[INFO] Finished at: 2018-07-31T10:10:35+05:30
[INFO] -----
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /Users/vatnaray/.jenkins/workspace/ParameterizedBuildWithEmail1/MavenProject/pom.xml to
org.jfrog.test/multi/3.7-SNAPSHOT/multi-3.7-SNAPSHOT.pom
[JENKINS] Archiving /Users/vatnaray/.jenkins/workspace/ParameterizedBuildWithEmail1/MavenProject/multi3/pom.xml
to org.jfrog.test/multi3/3.7-SNAPSHOT/multi3-3.7-SNAPSHOT.pom
[JENKINS] Archiving
/Users/vatnaray/.jenkins/workspace/ParameterizedBuildWithEmail1/MavenProject/multi3/target/multi3-3.7-
SNAPSHOT.war to org.jfrog.test/multi3/3.7-SNAPSHOT/multi3-3.7-SNAPSHOT.war
[JENKINS] Archiving /Users/vatnaray/.jenkins/workspace/ParameterizedBuildWithEmail1/MavenProject/multi1/pom.xml
to org.jfrog.test/multi1/3.7-SNAPSHOT/multi1-3.7-SNAPSHOT.pom
[JENKINS] Archiving
/Users/vatnaray/.jenkins/workspace/ParameterizedBuildWithEmail1/MavenProject/multi1/target/multi1-3.7-
SNAPSHOT.jar to org.jfrog.test/multi1/3.7-SNAPSHOT/multi1-3.7-SNAPSHOT.jar
[JENKINS] Archiving
/Users/vatnaray/.jenkins/workspace/ParameterizedBuildWithEmail1/MavenProject/multi1/target/multi1-3.7-SNAPSHOT-
sources.jar to org.jfrog.test/multi1/3.7-SNAPSHOT/multi1-3.7-SNAPSHOT-sources.jar
[JENKINS] Archiving
/Users/vatnaray/.jenkins/workspace/ParameterizedBuildWithEmail1/MavenProject/multi1/target/multi1-3.7-SNAPSHOT-
tests.jar to org.jfrog.test/multi1/3.7-SNAPSHOT/multi1-3.7-SNAPSHOT-tests.jar
[JENKINS] Archiving /Users/vatnaray/.jenkins/workspace/ParameterizedBuildWithEmail1/MavenProject/multi2/pom.xml
to org.jfrog.test/multi2/3.7-SNAPSHOT/multi2-3.7-SNAPSHOT.pom
[JENKINS] Archiving
/Users/vatnaray/.jenkins/workspace/ParameterizedBuildWithEmail1/MavenProject/multi2/target/multi2-3.7-
SNAPSHOT.jar to org.jfrog.test/multi2/3.7-SNAPSHOT/multi2-3.7-SNAPSHOT.jar
channel stopped
Started calculate disk usage of build
Finished Calculation of disk usage of build in 0 seconds
Started calculate disk usage of workspace
Finished Calculation of disk usage of workspace in 0 seconds
Triggering a new build of Maven_Deploy
Finished: SUCCESS
```

Changes

Console Output

View as plain text

Edit Build Information

Delete Build

Parameters

Git Build Data

No Tags

Aggregated Test Result

Rebuild

Build Artifacts As Maven Repository

Previous Build

Next Build

```
Started by upstream project "ParameterizedBuildWithEmail1" build number 7
originally caused by:
  Started by user Vathsala
Building on master in workspace /Users/vatnaray/.jenkins/workspace/Maven_Deploy
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/vathsalahn/jenkins-demo.git # timeout=10
Fetching upstream changes from https://github.com/vathsalahn/jenkins-demo.git
> git --version # timeout=10
> git fetch --tags --progress https://github.com/vathsalahn/jenkins-demo.git
+refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 3495e61b1275296763260189b1bbf702f4f6bae0 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 3495e61b1275296763260189b1bbf702f4f6bae0
Commit message: "Rename Jenkinsfile to JenkinsFile"
> git rev-list --no-walk 3495e61b1275296763260189b1bbf702f4f6bae0 # timeout=10
No emails were triggered.
[Maven_Deploy] $ /bin/sh -xe /var/folders/64/jgx_kznx47q658s42c88cp340000gn/T/jenkins9098349436253808180.sh
+ echo http://localhost:8080/ 7
http://localhost:8080/ 7
+ python python-example/pythonProj/pythonProj.py 11 http://localhost:8080/ 7
Hello from Python Project
The parameterized output is 11
The project url is http://localhost:8080/
the current build id is 7
Started calculate disk usage of build
Finished Calculation of disk usage of build in 0 seconds
Started calculate disk usage of workspace
Finished Calculation of disk usage of workspace in 0 seconds
Email was triggered for: Always
Sending email for trigger: Always
Sending email to: vathsala.hn22@gmail.com
Finished: SUCCESS
```