Vidyalankar Institute of Technology
Accredited A+ by NAAC

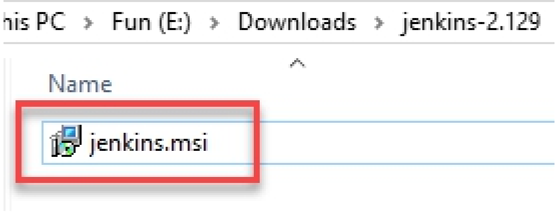
DEPARTMENT OF INFORMATION TECHNOLOGY Experiment No5

Semester	Semester VIII	
Subject	DevOps Lab	
Subject Professor In-charge	Prof. Yash Shah	
Laboratory	L11B	
Student Name	Ashwini Jadhav	
Roll Number	17101B0038	
Grade and Subject Teacher's Signature		

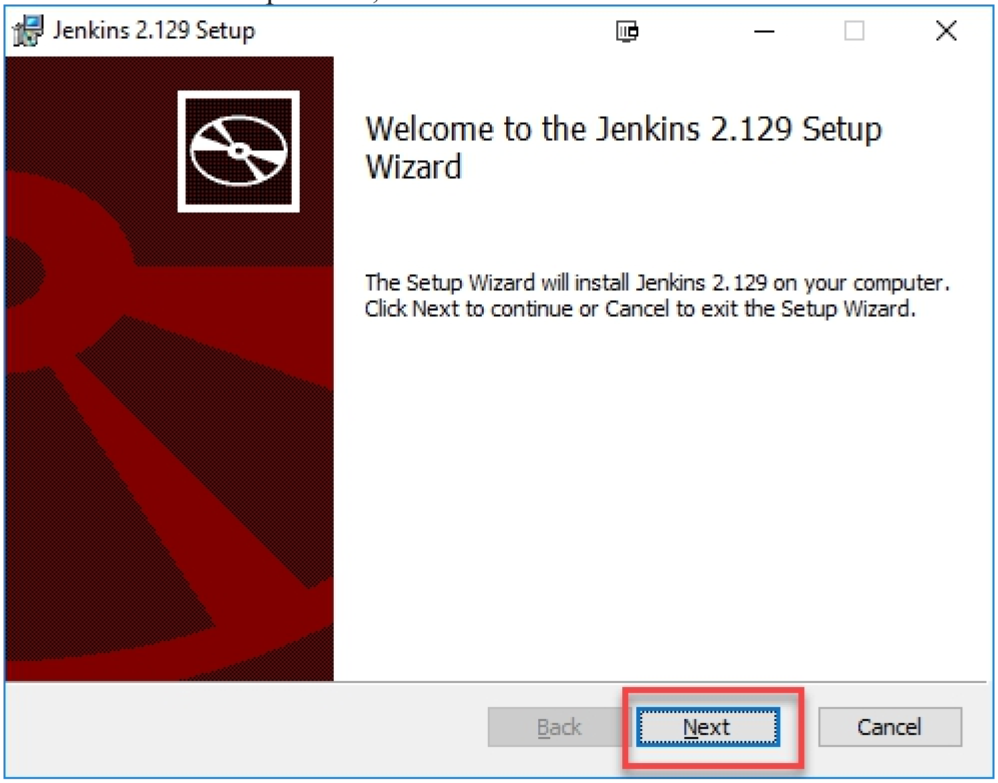
Experiment Number	5	
Experiment Title	To install Jenkins and create different builds on a single job	
Resources / Apparatus Required	Hardware: Compatible Computer System	Java Development Kit (JDK) v.11
Objectives	Explore and implement builds on Jenkins	
Theory	**What is jenkins?** Jenkins is an open source Continuous Integration server capable of orchestrating a chain of actions that help to achieve the Continuous Integration process (and not only) in an automated fashion. Jenkins is a widely used application around the world that has around 300k installations and growing day by day. **Need of Jenkins:** The reason Jenkins became so popular is that of its monitoring of repeated tasks which arise during the development of a project. For example, if your team is developing a project, Jenkins will continuously test your project builds and show you the errors in early stages of your development. By using Jenkins, software companies can accelerate their software development process, as Jenkins can automate build and test at a rapid rate. Jenkins supports the complete development lifecycle of software from building, testing, documenting the software, deploying and other stages of a software development lifecycle.	

Installation steps:

1. Double-click on unzipped jenkins.msi. You can also Jenkin using a WAR (Web application ARchive) but that is not recommended.



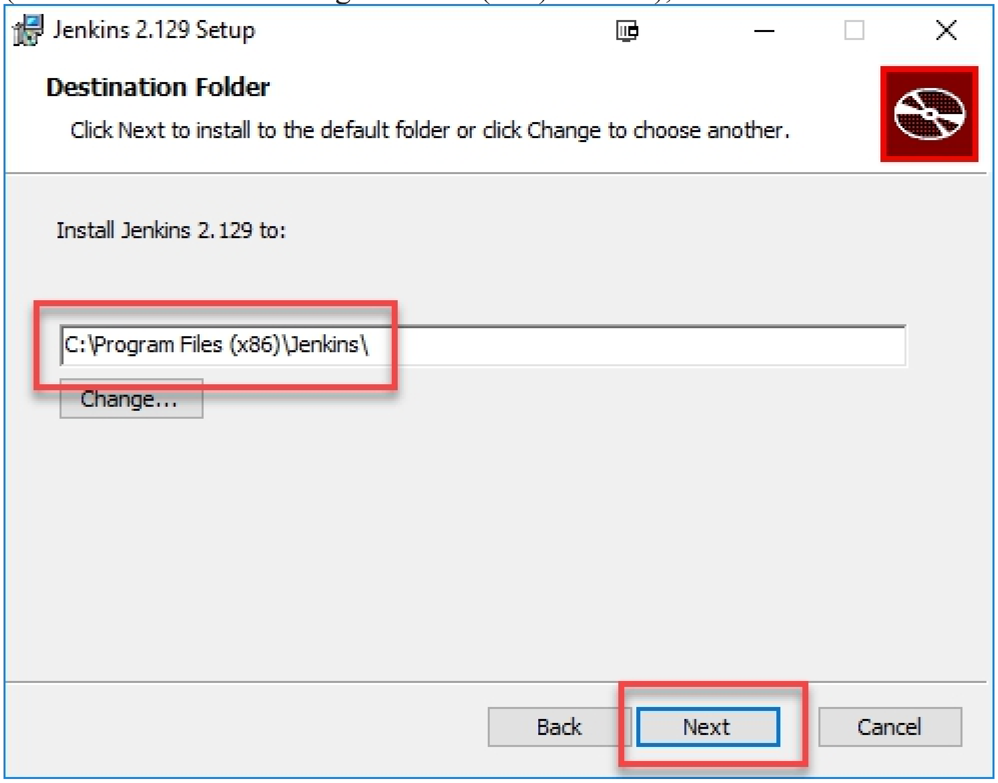
2. In the Jenkin Setup screen, click Next.



Welcome to the Jenkins 2.129 Setup Wizard

The Setup Wizard will install Jenkins 2.129 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

3. Choose the location where you want to have the Jenkins instance installed (default location is C:\Program Files (x86)\Jenkins), then click on **Next** button.



Jenkins 2.129 Setup

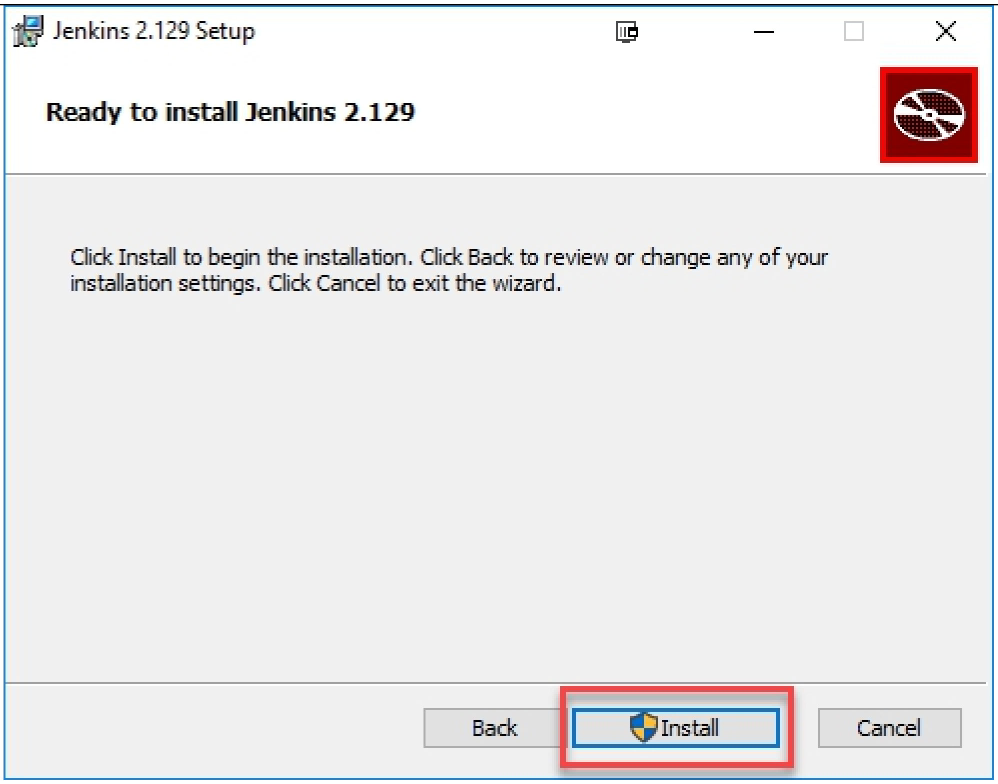
Destination Folder

Click Next to install to the default folder or click Change to choose another.

Install Jenkins 2.129 to:

C:\Program Files (x86)\Jenkins\

4. Click on the Install button.



Jenkins 2.129 Setup

Ready to install Jenkins 2.129

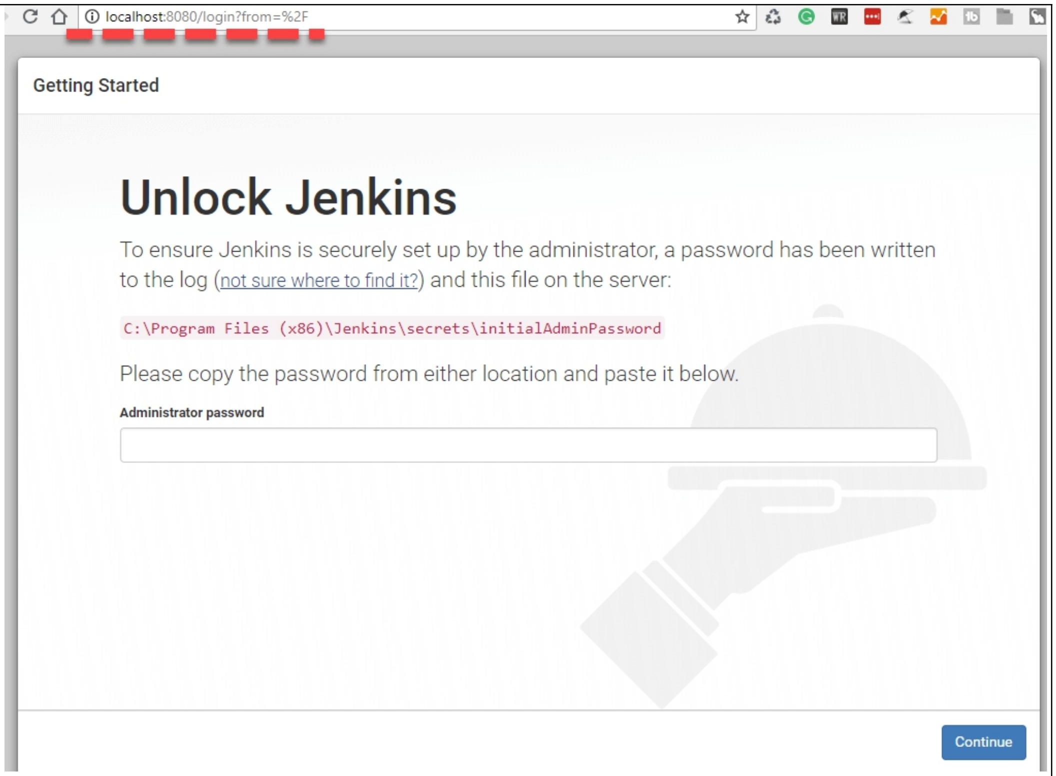
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

5. Once install is complete, click Finish.

6. After completing the Jenkins installation process, a browser tab will pop-up asking for the initial Administrator password. To access Jenkins, you need to go to browse the following path in your web browser.

http://localhost:8080

If you can access the above URL, then it confirms that Jenkins is successfully installed in your system.



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

```
C:\Program Files (x86)\Jenkins\secrets\initialAdminPassword
```

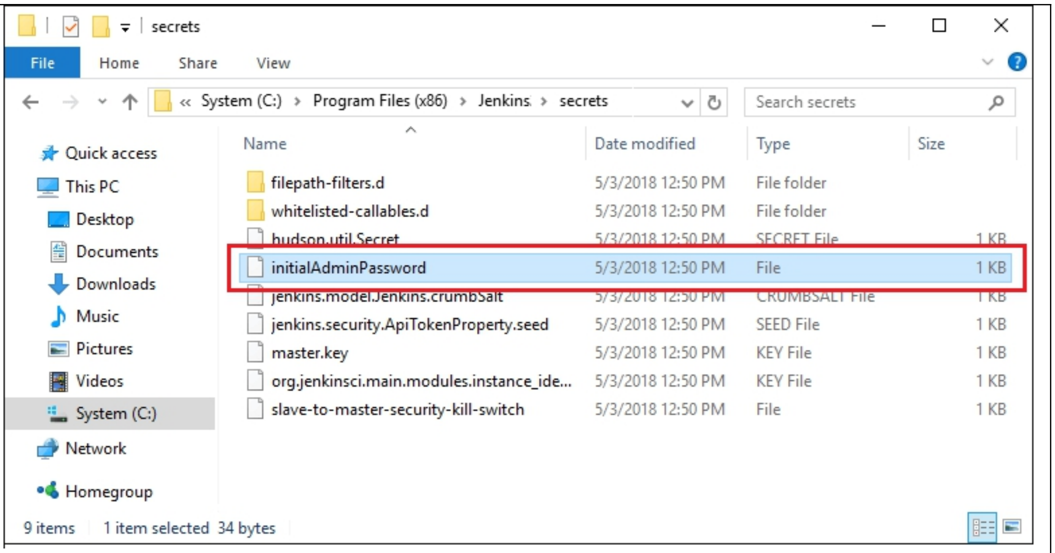
Please copy the password from either location and paste it below.

Administrator password

7. The initial Administrator password should be found under the Jenkins installation path (set at Step 4 in Jenkins Installation).

For default installation location to C:\Program Files (x86)\Jenkins, a file called **initialAdminPassword** can be found under C:\Program Files (x86)\Jenkins\secrets.

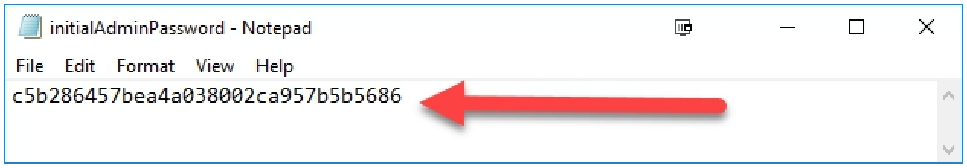
However, If a custom path for Jenkins installation was selected, then you should check that location for **initialAdminPassword** file.



Name	Date modified	Type	Size
filepath-filters.d	5/3/2018 12:50 PM	File folder	
whitelisted-callables.d	5/3/2018 12:50 PM	File folder	
hudson.util.Secret	5/3/2018 12:50 PM	SECRET File	1 KB
initialAdminPassword	5/3/2018 12:50 PM	File	1 KB
jenkins.model.Jenkins.crumbSalt	5/3/2018 12:50 PM	CRUMBSALT File	1 KB
jenkins.security.ApiTokenProperty.seed	5/3/2018 12:50 PM	SEED File	1 KB
master.key	5/3/2018 12:50 PM	KEY File	1 KB
org.jenkinsci.main.modules.instance_ide...	5/3/2018 12:50 PM	KEY File	1 KB
slave-to-master-security-kill-switch	5/3/2018 12:50 PM	File	1 KB

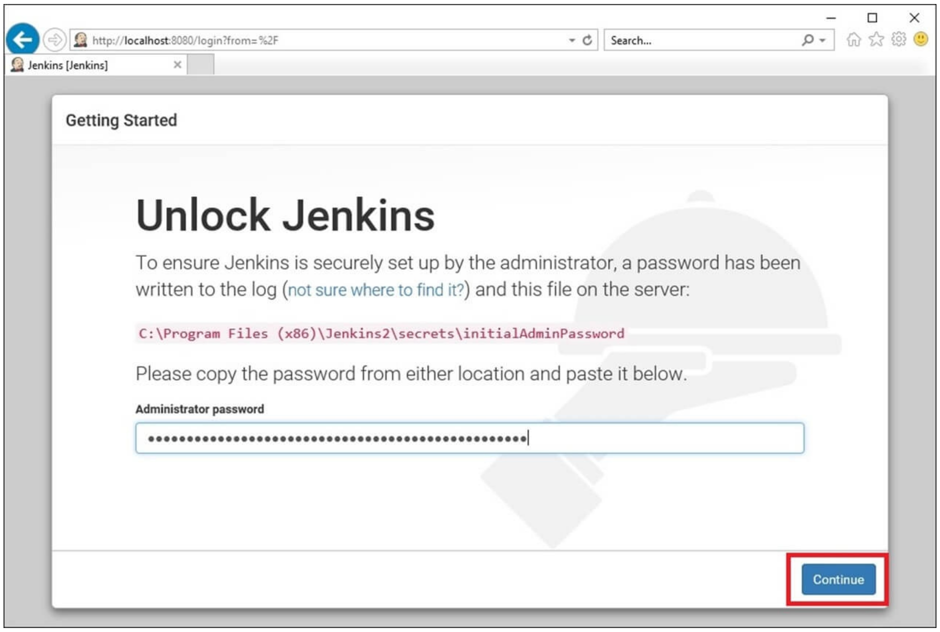
9 items 1 item selected 34 bytes

8. Open the highlighted file and copy the content of the **initialAdminPassword** file.



```
c5b286457bea4a038002ca957b5b5686
```

9. Paste the password it into browser's pop-up tab (http://localhost:8080/login?form=%2F) and click on Continue button.



Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

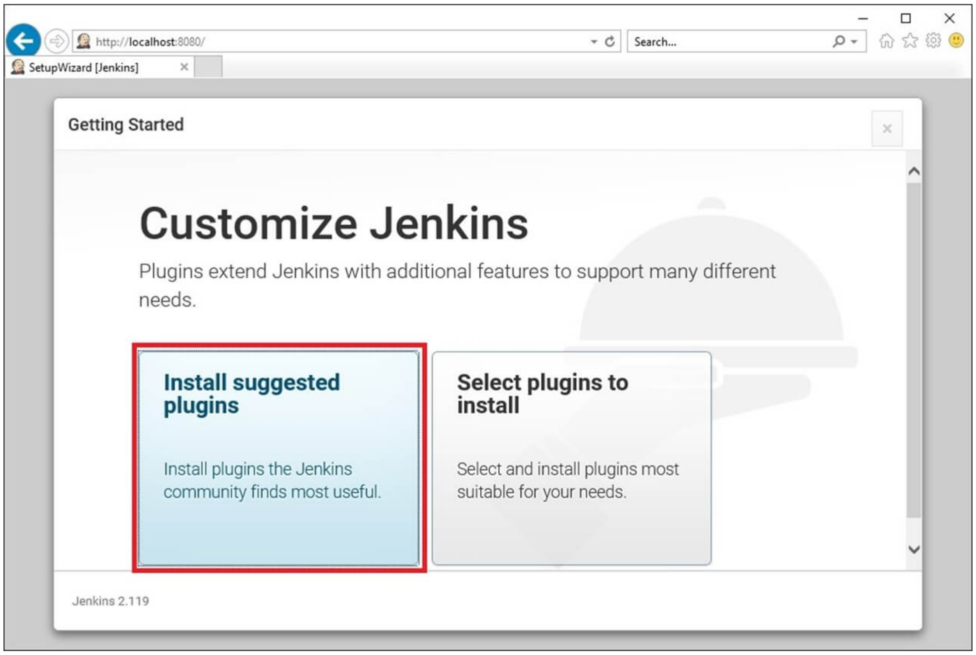
```
C:\Program Files (x86)\Jenkins2\secrets\initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

••

10. Click on the "Install suggested plugins button" so Jenkins will retrieve and install the essential plugins



Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

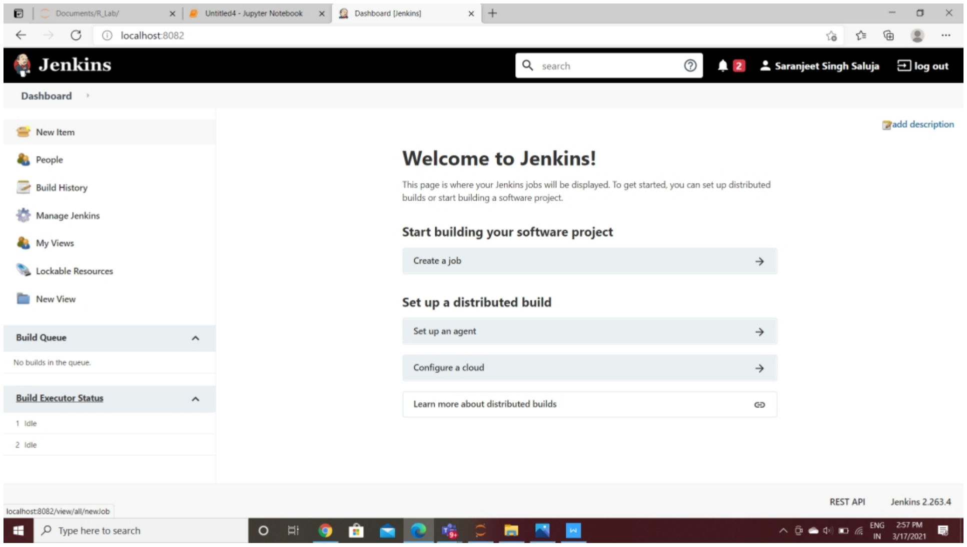
Install plugins the Jenkins community finds most useful.

Select and install plugins most suitable for your needs.

11. Jenkins will start to download and install all the necessary plugins needed to create new Jenkins Jobs.

Output

1. Create new project by clicking on New Item on the Dashboard

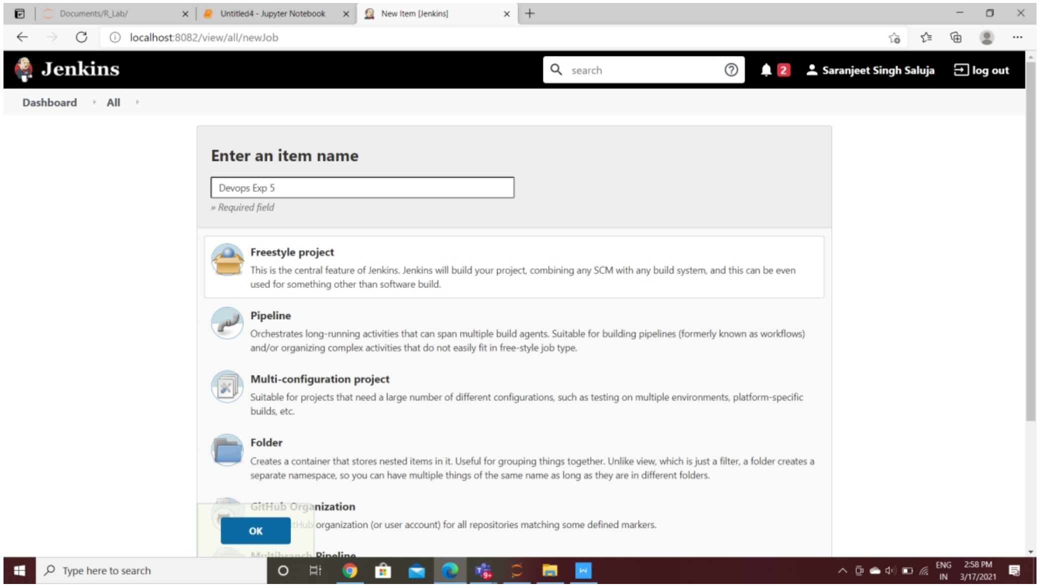


Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

No builds in the queue.

2. In the next screen, enter the Item name, in this case we have named it Devops Exp 5. Choose the ‘Freestyle project option’



Enter an item name

Devops Exp 5

» Required field

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.

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.

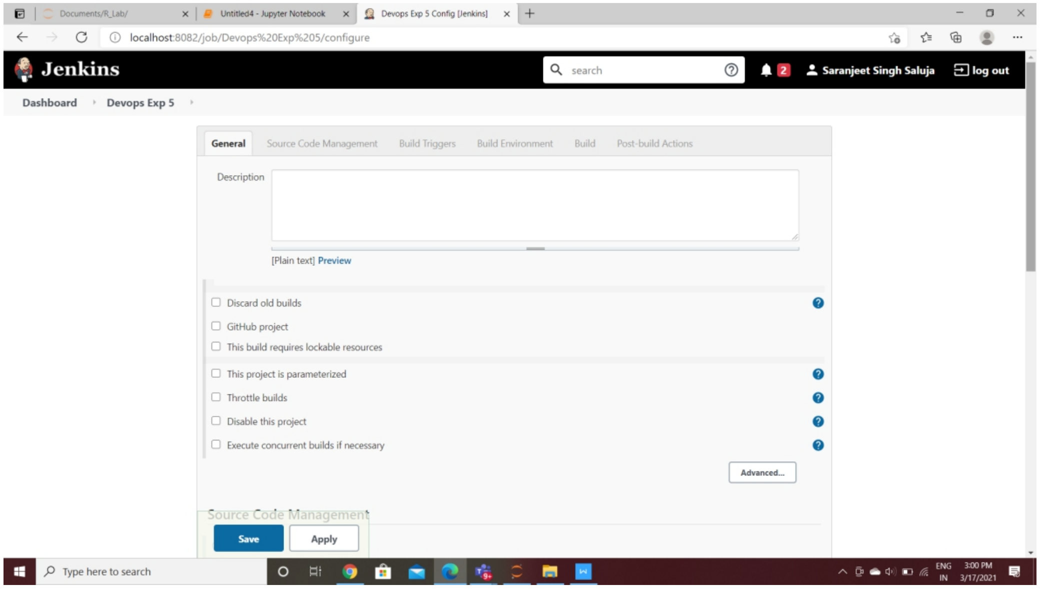
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

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
organization (or user account) for all repositories matching some defined markers.

Multibranch Pipeline

3. The following screen will come up in which you can specify the details of the job.



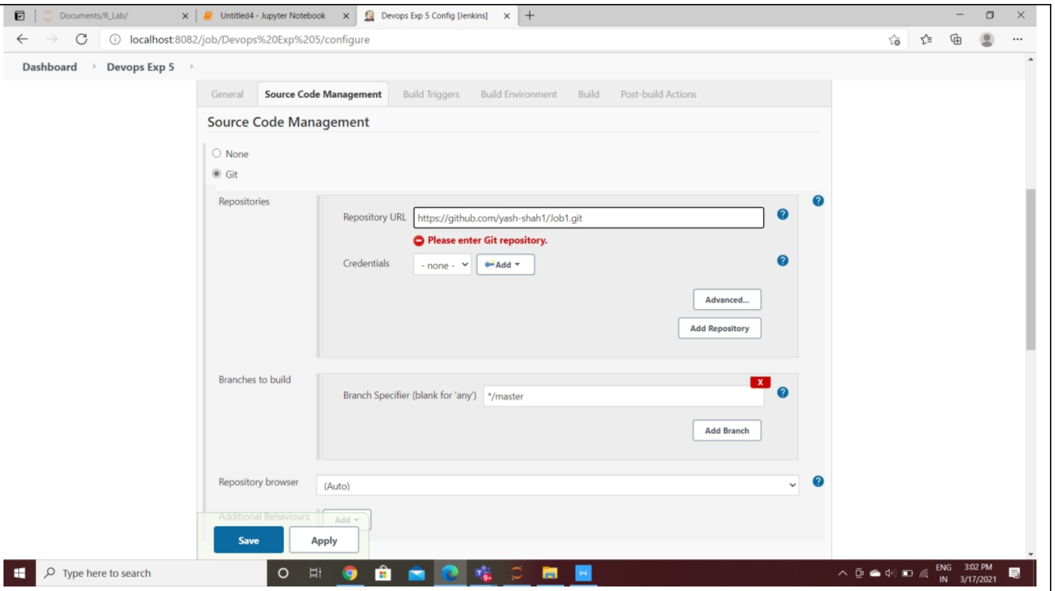
Description

[Plain text] Preview

□ Discard old builds
□ GitHub project
□ This build requires lockable resources
□ This project is parameterized
□ Throttle builds
□ Disable this project
□ Execute concurrent builds if necessary

Source Code Management

4. We need to specify the location of files which need to be built. If you have a repository hosted on Github, you can also enter the url of that repository here. In addition to this, you would need to click on the Add button for the credentials to add a user name and password to the github repository so that the code can be picked up from the remote repository.



Source Code Management

○ None

● Git

Repositories

Repository URL: https://github.com/yash-shah1/Job1.git

Please enter Git repository.

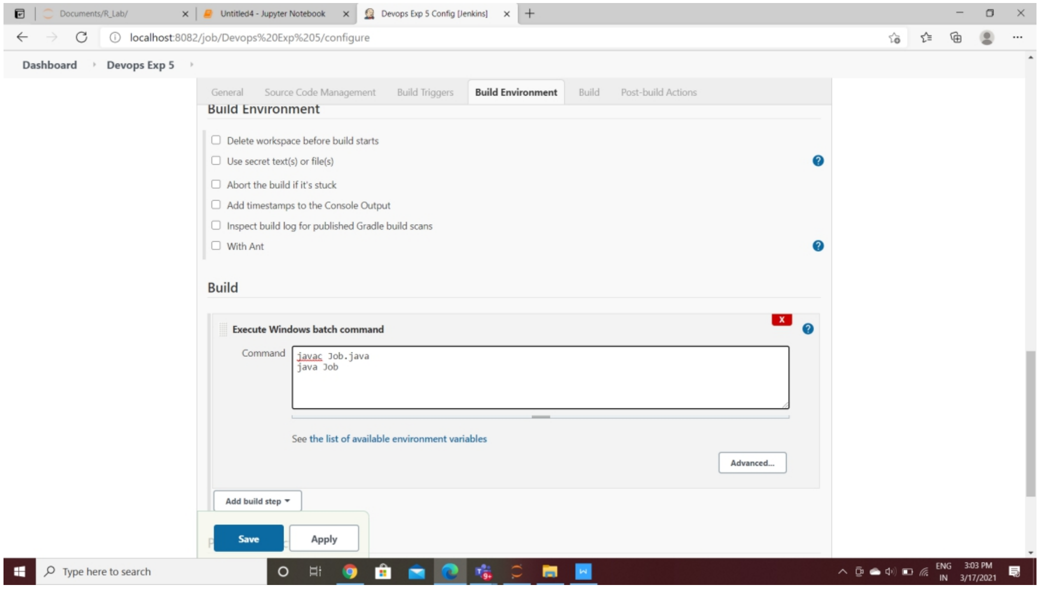
Credentials: - none -

Branches to build

Branch Specifier (blank for 'any'): */master

Repository browser: (Auto)

5. Go to the Build section and click on Add build step → Execute Windows batch command. In the command window, enter the following commands and then click on the Save button.



Build Environment

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s)

☐ Abort the build if it's stuck

☐ Add timestamps to the Console Output

☐ Inspect build log for published Gradle build scans

☐ With Ant

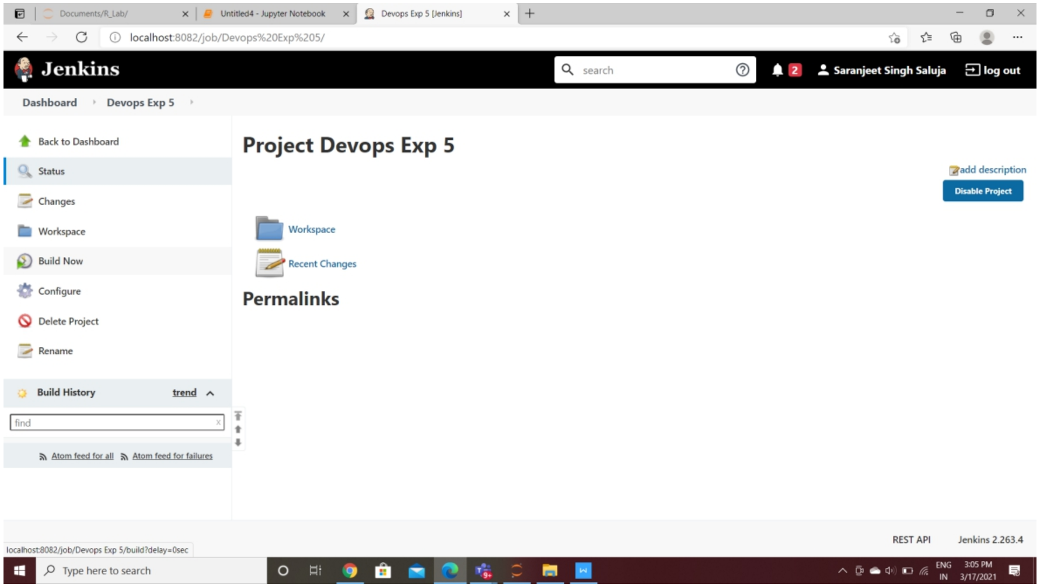
Build

Execute Windows batch command

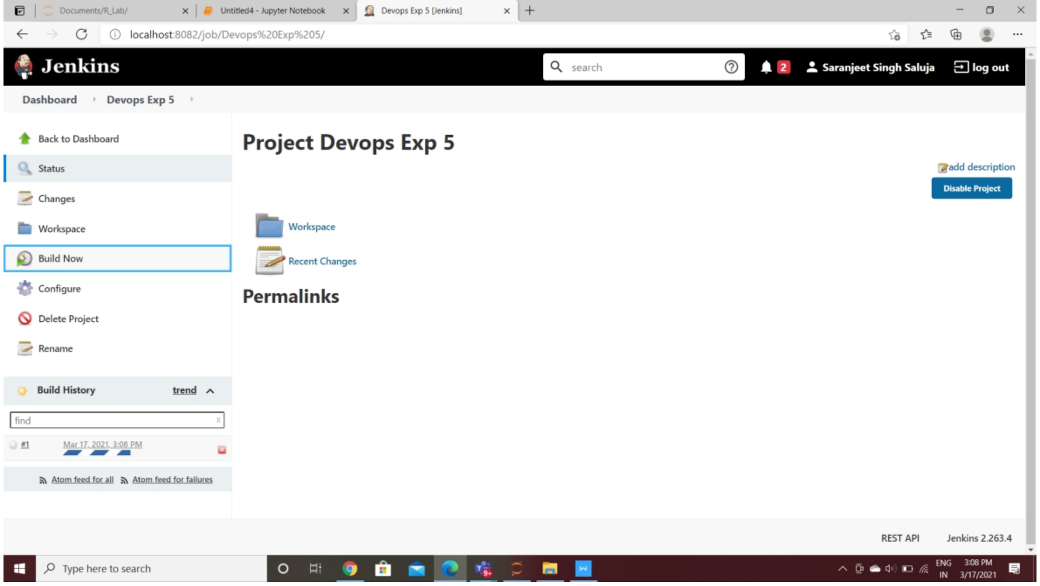
Command:

```
javac Job.java
java Job
```

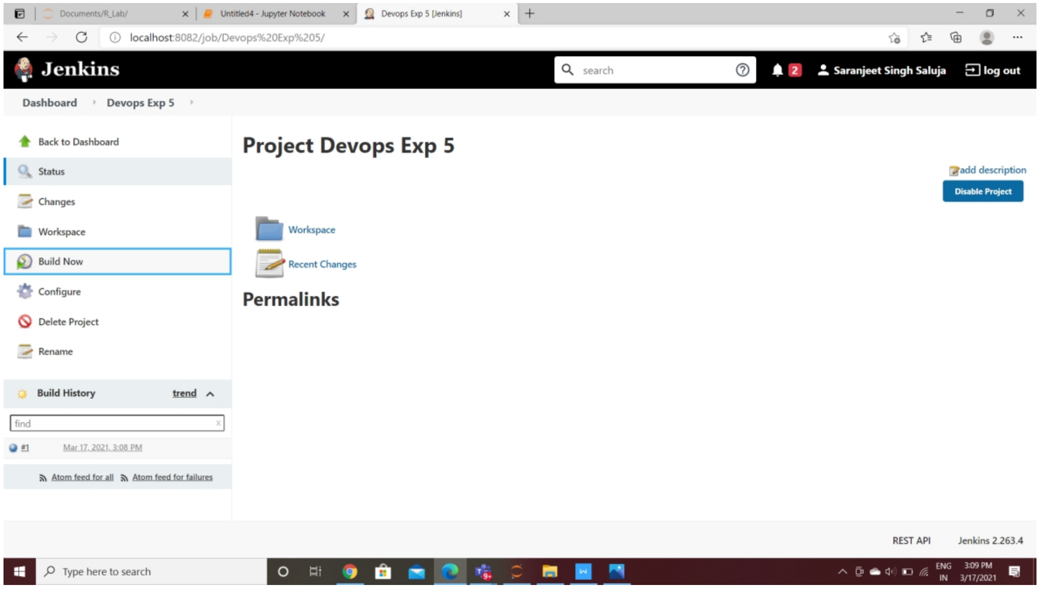
6. Once saved, you can click on the Build Now option to see if you have successfully defined the job.



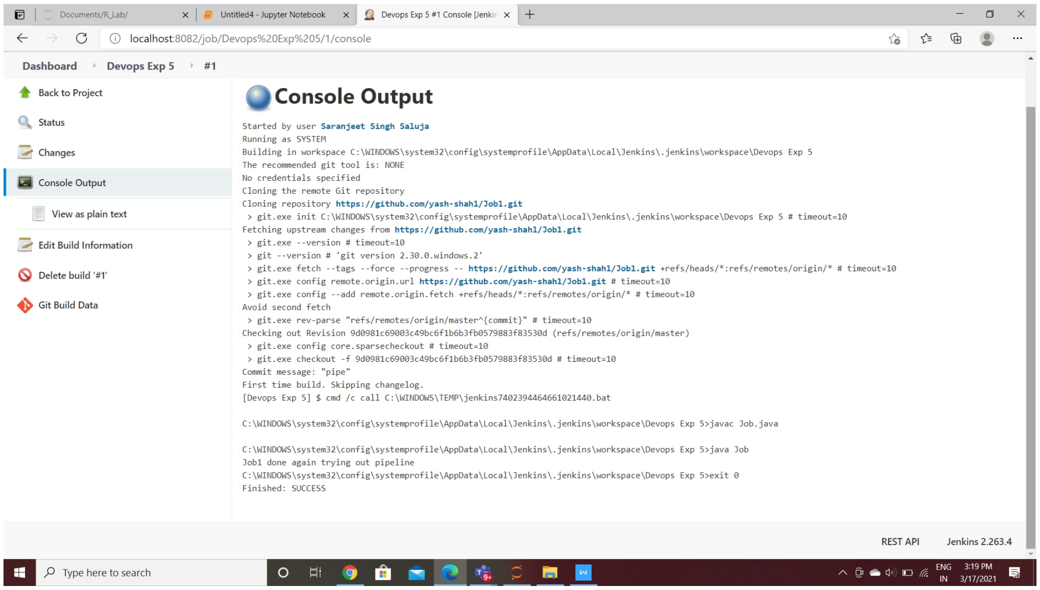
7. Once the build is scheduled, it will run. The following Build history section shows that a build is in progress.



8. Once the build is completed, a status of the build will show if the build was successful or not. In our case, the following build has been executed successfully.



9. Click on the Console Output link to see the details of the build



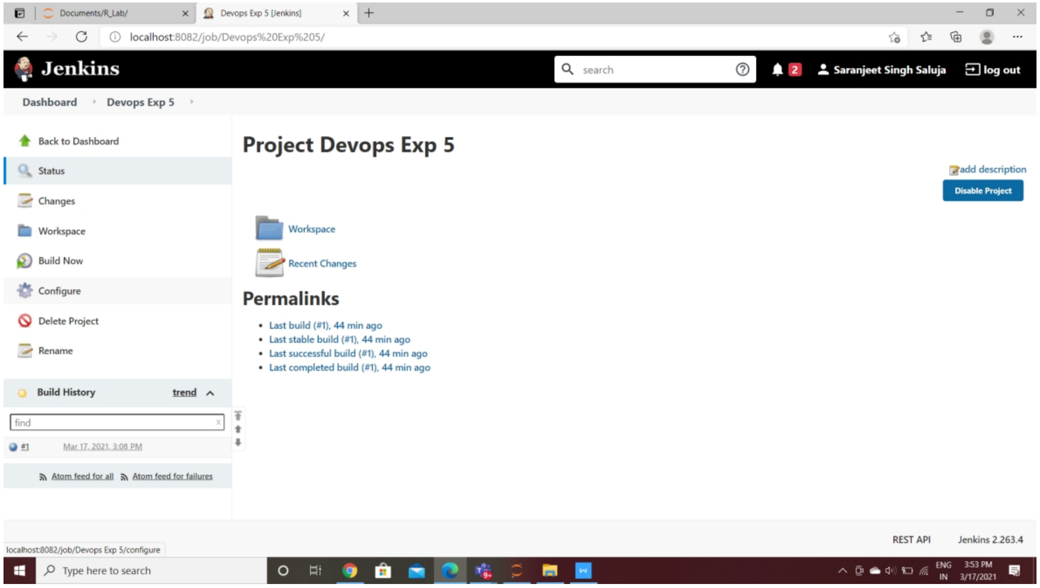
Console Output

```
Started by user Saranjeet Singh Saluja
Running as SYSTEM
Building in workspace C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\Devops Exp 5
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/yash-shah1/Job1.git
 > git.exe init C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\Devops Exp 5 # timeout=10
Fetching upstream changes from https://github.com/yash-shah1/Job1.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.30.0.windows.2'
 > git.exe fetch --tags --force --progress -- https://github.com/yash-shah1/Job1.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe config remote.origin.url https://github.com/yash-shah1/Job1.git # timeout=10
 > git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision 9d0981c69003c49bc6f1b6b3fb0579883f83530d (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f 9d0981c69003c49bc6f1b6b3fb0579883f83530d # timeout=10
Commit message: "pipe"
First time build. Skipping changelog.
[Devops Exp 5] $ cmd /c call C:\WINDOWS\TEMP\jenkins7402394464661021440.bat

C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\Devops Exp 5>javac Job.java

C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\Devops Exp 5>java Job
Job1 done again trying out pipeline
C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\workspace\Devops Exp 5>exit 0
Finished: SUCCESS
```

10. To make changes to your build, click on configure.

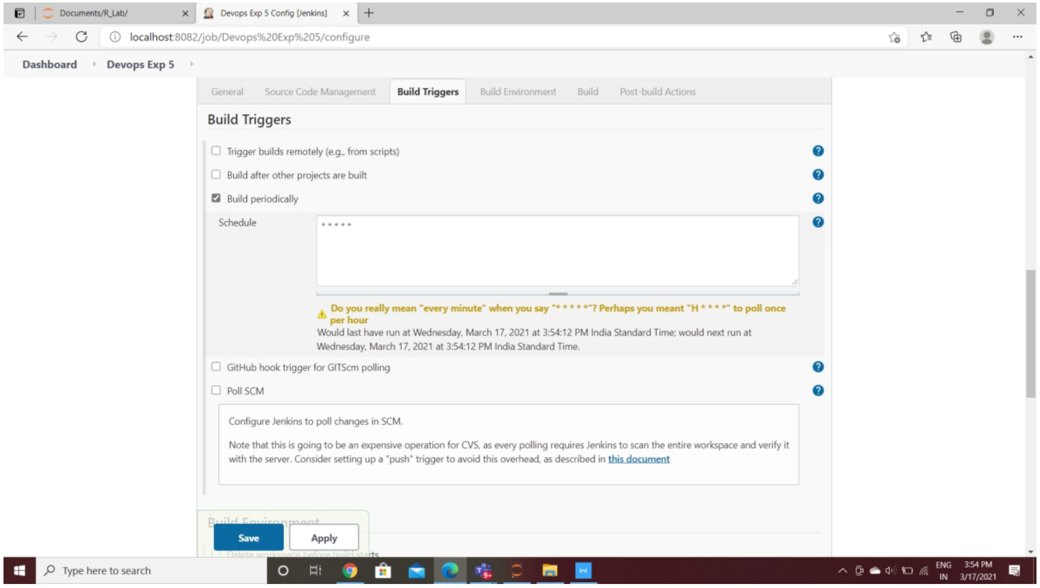


Project Devops Exp 5

Permalinks

- Last build (#1), 44 min ago
- Last stable build (#1), 44 min ago
- Last successful build (#1), 44 min ago
- Last completed build (#1), 44 min ago

11. To carry out triggers, in this case 'build periodically', we go to build triggers.



Build Triggers

☐ Trigger builds remotely (e.g., from scripts)

☐ Build after other projects are built

☑ Build periodically

Schedule

```
* * * * *
```

Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour

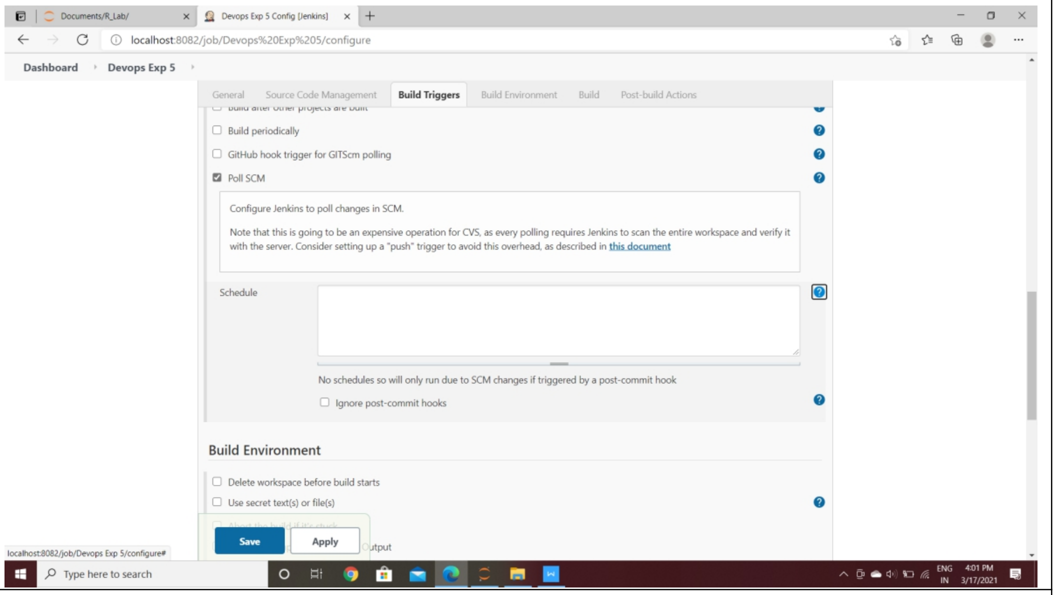
Would last have run at Wednesday, March 17, 2021 at 3:54:12 PM India Standard Time; would next run at Wednesday, March 17, 2021 at 3:54:12 PM India Standard Time.

☐ GitHub hook trigger for GITScm polling

☐ Poll SCM

Configure Jenkins to poll changes in SCM.

Note that this is going to be an expensive operation for CVS, as every polling requires Jenkins to scan the entire workspace and verify it with the server. Consider setting up a "push" trigger to avoid this overhead, as described in this document

	12. In 'Poll SCM', it periodically polls the SCM to check whether changes were made (i.e. new commits) and builds the project if new commits where pushed since the last build  Documents/R_Lab/ Devops Exp 5 Config [Jenkins] localhost:8082/job/Devops%20Exp%205/configure Dashboard Devops Exp 5 General Source Code Management Build Triggers Build Environment Build Post-build Actions □ Build after other projects are built □ Build periodically □ GitHub hook trigger for GITScm polling ☑ Poll SCM Configure Jenkins to poll changes in SCM. Note that this is going to be an expensive operation for CVS, as every polling requires Jenkins to scan the entire workspace and verify it with the server. Consider setting up a "push" trigger to avoid this overhead, as described in this document Schedule No schedules so will only run due to SCM changes if triggered by a post-commit hook □ Ignore post-commit hooks Build Environment □ Delete workspace before build starts □ Use secret text(s) or file(s) Save Apply localhost:8082/job/Devops Exp 5/configure# Type here to search ENG IN 4:01 PM 3/17/2021
Conclusion	Thus, we have implemented builds using jenkins and saved project to jenkins.