

## EXPERIMENT 5

**Aim:** Experiment 5: To Build the pipeline of jobs using Maven / Gradle / Ant in Jenkins, create a pipeline script to Test and deploy an application over the tomcat server

### Programming in Jenkins:

Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible.” In simple way, Continuous integration (CI) is the practice of frequently building and testing each change done to your code automatically.

Jenkins is a self-contained, open-source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.

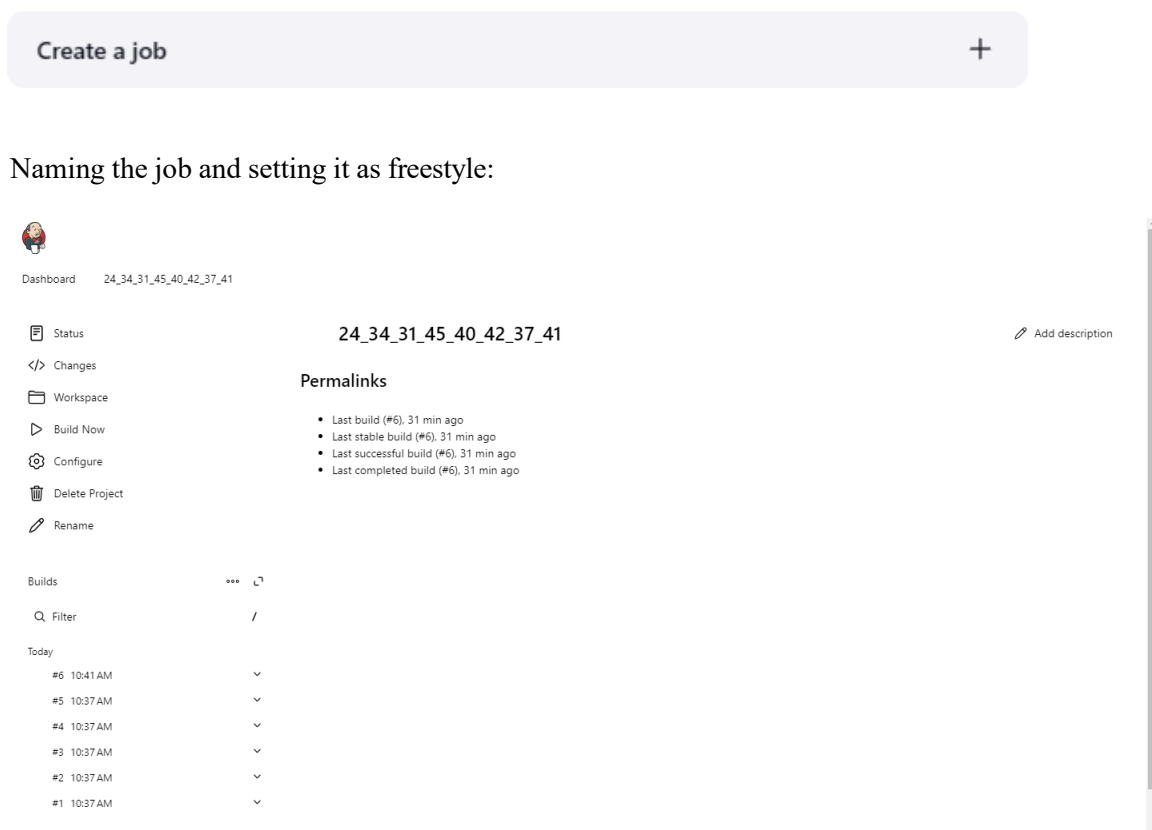
Our first job will execute the shell commands. The freestyle project provides enough options and features to build the complex jobs that you will need in your projects.

### Example 1

#### Example 1.1: Deploying a freestyle app in Jenkins

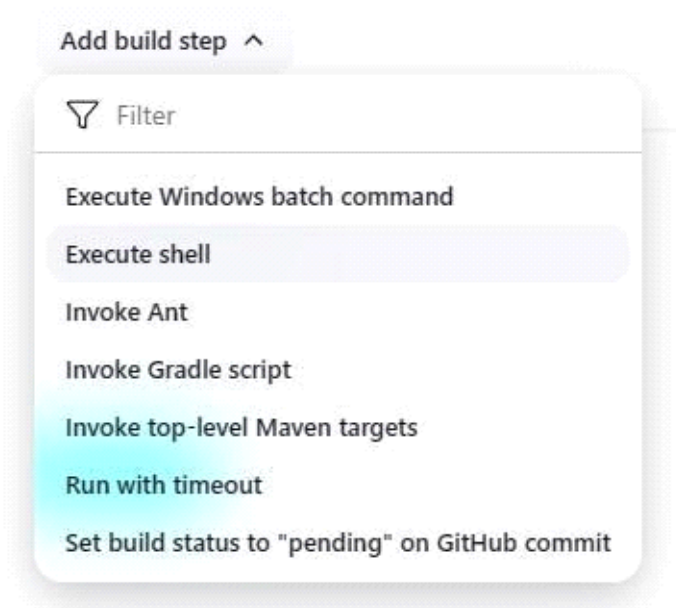
Creating a job:

#### Start building your software project



Selecting build type as “Execute shell”:

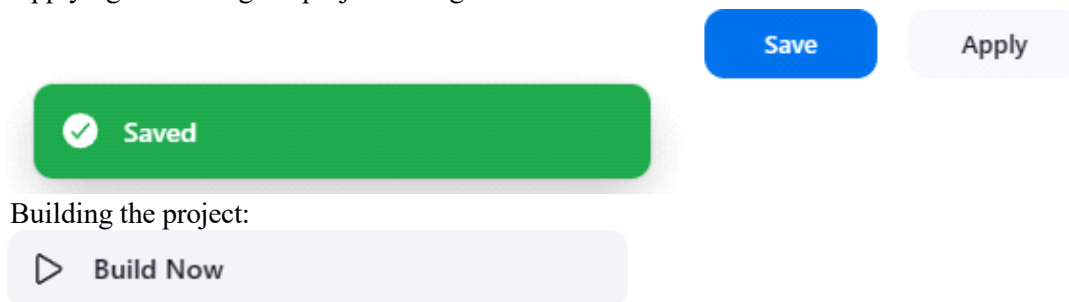
## Build Steps



Entering a simple command for the shell execution:



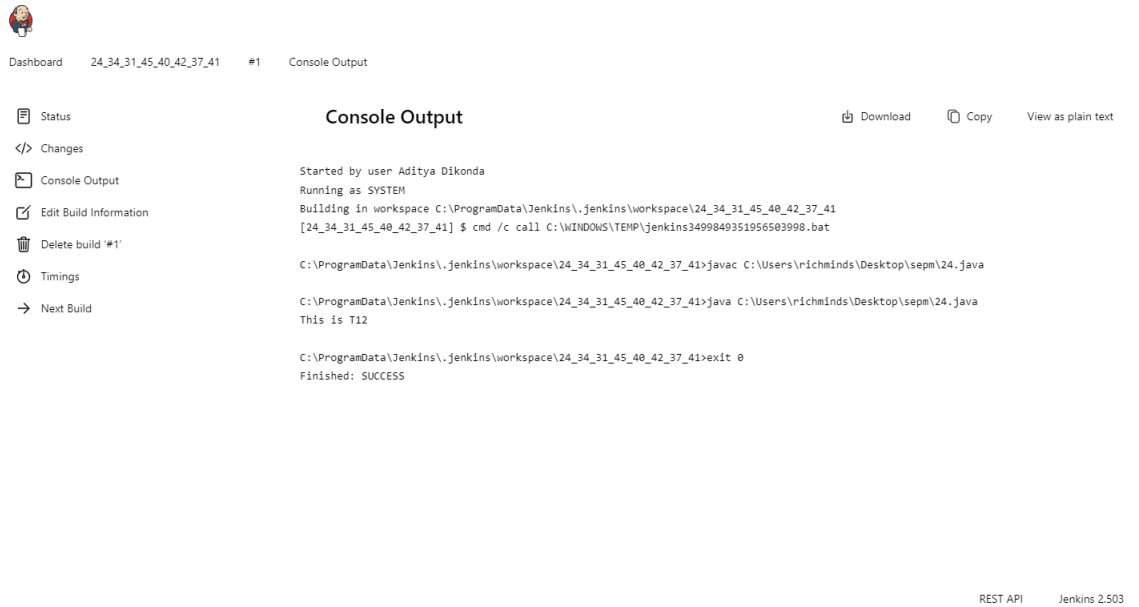
Applying and saving the project configuration:



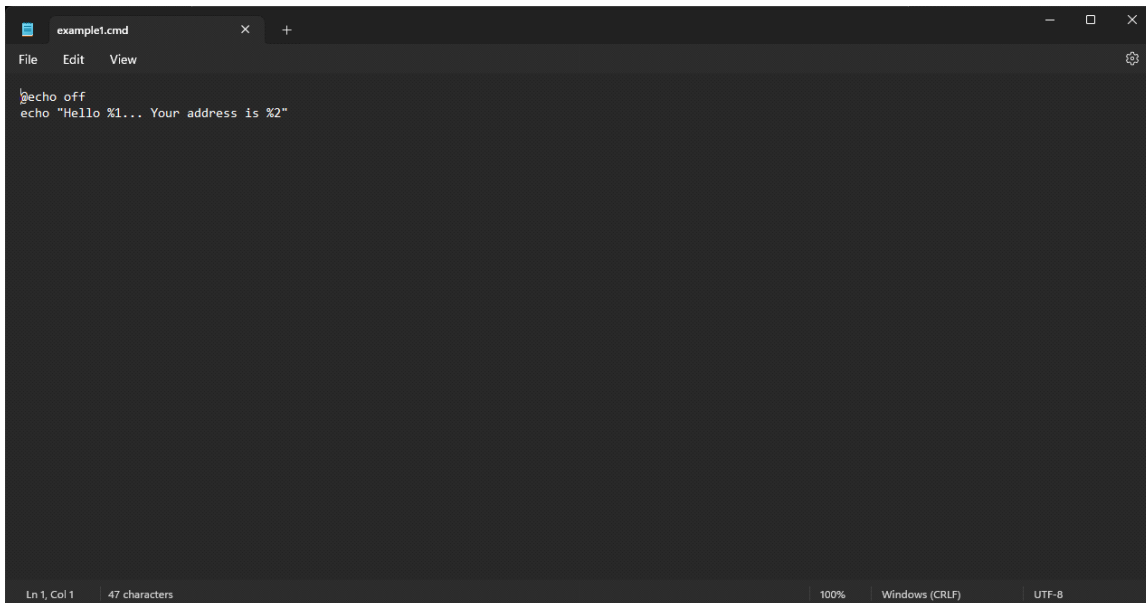
Building the project:

## EXPERIMENT 5

### Console output (after building):



A screenshot of the Jenkins web interface showing the console output of a build. The breadcrumb navigation at the top reads: Dashboard > 24\_34\_31\_45\_40\_42\_37\_41 > #1 > Console Output. On the left sidebar, there are links for Status, Changes, Console Output (active), Edit Build Information, Delete build '#1', Timings, and Next Build. The main area is titled 'Console Output' and contains the following text: Started by user Aditya Dikonda, Running as: SYSTEM, Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\24\_34\_31\_45\_40\_42\_37\_41, [24\_34\_31\_45\_40\_42\_37\_41] \$ cmd /c call C:\WINDOWS\TEMP\jenkins3499849351956503998.bat, C:\ProgramData\Jenkins\jenkins\workspace\24\_34\_31\_45\_40\_42\_37\_41>javac C:\Users\richminds\Desktop\sepm\24.java, C:\ProgramData\Jenkins\jenkins\workspace\24\_34\_31\_45\_40\_42\_37\_41>java C:\Users\richminds\Desktop\sepm\24.java, This is T12, C:\ProgramData\Jenkins\jenkins\workspace\24\_34\_31\_45\_40\_42\_37\_41>exit 0, Finished: SUCCESS. At the bottom right, there are links for REST API and Jenkins 2.503.



A screenshot of a text editor window titled 'example1.cmd'. The menu bar shows 'File', 'Edit', and 'View'. The text content is: `@echo off`  
`echo "Hello %1... Your address is %2"`. The status bar at the bottom indicates 'Ln 1, Col 1', '47 characters', '100%', 'Windows (CRLF)', and 'UTF-8'.

### Example 1.2: Taking parameters through files

Contents of script `example1.cmd`:

Executing script `example1.cmd` on the terminal:

## EXPERIMENT 5

```
Tomcat
12-Feb-2025 11:44:46.716 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -XX:+E
xplicitGCInvokesConcurrent
12-Feb-2025 11:44:46.716 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Xms38
4m
12-Feb-2025 11:44:46.716 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Xmx20
48m
12-Feb-2025 11:44:46.716 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Digno
re.endorsed.dirs=
12-Feb-2025 11:44:46.716 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Dcata
lina.base=C:\Users\15L\Atlassian\Jira
12-Feb-2025 11:44:46.717 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Dcata
lina.home=C:\Users\15L\Atlassian\Jira
12-Feb-2025 11:44:46.717 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djava
.io.tmpdir=C:\Users\15L\Atlassian\Jira\temp
12-Feb-2025 11:44:46.775 INFO [main] org.apache.catalina.core.AprLifecycleListener.lifecycleEvent Loaded Apache Tomcat N
ative library [1.3.1] using APR version [1.7.4].
12-Feb-2025 11:44:46.775 INFO [main] org.apache.catalina.core.AprLifecycleListener.lifecycleEvent APR capabilities: IPv6
[true], sendfile [true], accept filters [false], random [true], UDS [true].
12-Feb-2025 11:44:46.775 INFO [main] org.apache.catalina.core.AprLifecycleListener.lifecycleEvent APR/OpenSSL configurat
ion: useAprConnector [false], useOpenSSL [true]
12-Feb-2025 11:44:46.786 INFO [main] org.apache.catalina.core.AprLifecycleListener.initializeSSL OpenSSL successfully in
itialized [OpenSSL 3.0.14 4 Jun 2024]
12-Feb-2025 11:44:47.005 INFO [main] org.apache.coyote.AbstractProtocol.init Initializing ProtocolHandler ["http-nio-808
0"]
12-Feb-2025 11:44:47.014 INFO [main] org.apache.catalina.startup.Catalina.load Server initialization in [521] millisecon
ds
12-Feb-2025 11:44:47.032 INFO [main] org.apache.catalina.core.StandardService.startInternal Starting service [Catalina]
12-Feb-2025 11:44:47.032 INFO [main] org.apache.catalina.core.StandardEngine.startInternal Starting Servlet engine: [Apa
che Tomcat/9.0.98]
```

Modifying the Jenkins project to execute the script while supplying required parameters:

### Build Steps

Execute Windows batch command ?

Command

See [the list of available environment variables](#)

C:\Admin\Academics\TSEC\Start3\SEPM\example1.cmd Siddhant Goregaon

Advanced ▾

Add build step ▾

Console output after building the modified project:

```
Status
</> Changes
Console Output
View as plain text
Edit Build Information
Delete build "4F"
Previous Build

Console Output
Started by user Siddhant Chetiar
Running as SYSTEM
Building in workspace C:\ProgramData\jenkins\workspace\example1
[example1] $ cd /c call C:\WINDOWS\TEMP\jenkins707509583E16532158.bat
C:\ProgramData\jenkins\workspace\example1>C:\Admin\Academics\TSEC\Start3\SEPM\example1.cmd Siddhant Goregaon
"Hello Siddhant... Your address is Goregaon"
Finished: SUCCESS
```

: Running

### a Java program under Jenkins

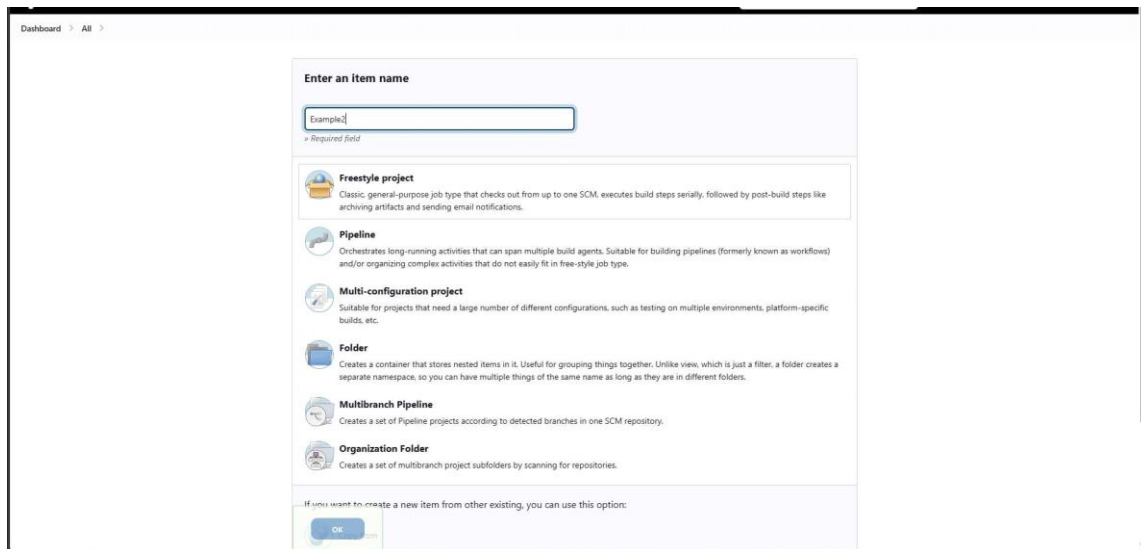
Creating a simple Java program:

Compiling and running the program on the terminal:

```
This is T12
```

Creating a new freestyle project:

## EXPERIMENT 5



Configure new project:

Command

See the list of available environment variables

```
javac C:\Users\richminds\Desktop\sepm\24.java  
java C:\Users\richminds\Desktop\sepm\24.java
```

Console output after building:

### Console Output

[Download](#)

[Copy](#)

[View as plain text](#)

```
Started by user Aditya Dikonda  
Running as SYSTEM  
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\24_34_31_45_40_42_37_41  
[24_34_31_45_40_42_37_41] $ cmd /c call C:\WINDOWS\TEMP\jenkins3970528995341461278.bat  
  
C:\ProgramData\Jenkins\.jenkins\workspace\24_34_31_45_40_42_37_41>javac C:\Users\richminds\Desktop\sepm\24.java  
  
C:\ProgramData\Jenkins\.jenkins\workspace\24_34_31_45_40_42_37_41>java C:\Users\richminds\Desktop\sepm\24.java  
This is T12  
  
C:\ProgramData\Jenkins\.jenkins\workspace\24_34_31_45_40_42_37_41>exit 0  
Finished: SUCCESS
```

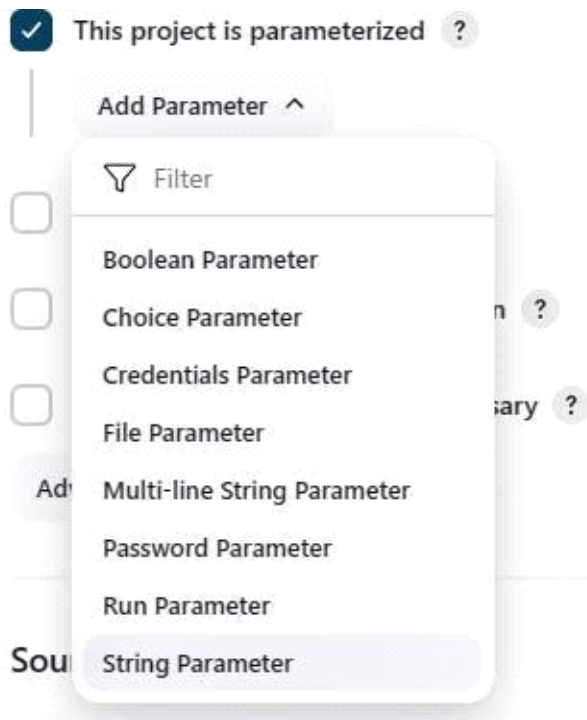
## Example 3

### Example 3.1: Parameterise build

Creating a new freestyle project:

Enabling parameterisation and adding a String parameter:

## EXPERIMENT 5



Configuring the string parameter as Fname:

The image shows a configuration form for a 'String Parameter'. The form has a title bar with a hamburger menu, the text 'String Parameter', and a close button. Below the title bar, there are three input fields: 'Name' (containing 'Fname'), 'Default Value' (empty), and 'Description' (empty). Below the 'Description' field, there is a 'Plain text' label and a 'Preview' link. At the bottom, there is a checkbox labeled 'Trim the string'.

Adding a choice parameter and configuring it as City with the following choices:

The image shows a configuration form for a 'Choice Parameter'. The form has a title bar with a hamburger menu, the text 'Choice Parameter', and a close button. Below the title bar, there are three input fields: 'Name' (containing 'City'), 'Choices' (containing a list of cities: Ambernath, Badlapur, Kalyan, Dombivli), and 'Description' (empty). Below the 'Choices' field, there is a warning icon and the text 'Requires Choices.'.

Configuring build steps:

## EXPERIMENT 5

### Build Steps

≡ Execute Windows batch command ?

Command

See [the list of available environment variables](#)

```
C:\Admin\Academics\TSEC\Start3\SEPM\example3.cmd %Fname% %City%
```

Advanced ▾

Add build step ▾

Entering parameters for build:

### Project Example3

This build requires parameters:

Fname

City

▶ Build

Cancel

Console output after building:

### ✓ Console Output

```
Started by user Siddhant Chetlur
Running as SYSTEM
[EnvInject] - Loading node environment variables.
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\Example3
[Example3] $ cmd /c call C:\WINDOWS\TEMP\jenkins14094536165150986151.bat

C:\ProgramData\Jenkins\jenkins\workspace\Example3>C:\Admin\Academics\TSEC\Start3\SEPM\example3.cmd Siddhant Bandra
Hello your name is Siddhant and your city is Bandra
Finished: SUCCESS
```

## Example 5

```
example5.py
File Edit View
import sys
num = int(sys.argv[1])

print(f"Original number is {num}")
print(f"Binary representation of {num} is {bin(num)}")
print(f"Octal representation of {num} is {oct(num)}")
print(f"Hexadecimal representation of {num} is {hex(num)}")
print(f"Complex representation of {num} is {complex(num)}")

Ln 9, Col 60 299 characters 100% Windows (CRLF) UTF-8
```

## EXPERIMENT 5

### Example 5.1: Running a Python program

Creating a simple Python script:


Running the Python script on the terminal:


```
lumber is 10
```


Creating a new freestyle project:


**Enter an item name**


» Required field


**Freestyle project**  
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

**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.

**Multibranch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.

**Organization Folder**  
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

Parameterising the project with a string parameter as follows:

☒ This project is parameterized [?](#)

**String Parameter** [?](#)

Name [?](#)

Default Value [?](#)

Description [?](#)

Plain text [Preview](#)

☐ Trim the string [?](#)

Add Parameter [v](#)



## EXPERIMENT 5

Configuring the build steps:

Command

See the list of available environment variables

```
python C:\Users\richminds\Desktop\sepm\24.py
```

Setting the parameter for the build:

The screenshot shows the Jenkins web interface for a project named 'Project 24\_34\_31\_45\_40\_42\_37\_41'. The left sidebar contains navigation links: Dashboard, 24\_34\_31\_45\_40\_42\_37\_41, Status, </> Changes, Workspace, Build with Parameters, Configure, Delete Project, and Rename. The main content area shows the 'Configure' page for the project. It indicates 'This build requires parameters:' and shows a parameter named 'num' with a value of '10'. A 'Cancel' button is visible. Below the configuration area, there is a 'Builds' section with a filter and a list of builds. The builds list shows a series of builds numbered #1 to #6, all with a status of 'Success' and a time of 10:37 AM, except for build #6 which is 10:41 AM.

Builds	Status	Time
#6	Success	10:41 AM
#5	Success	10:37 AM
#4	Success	10:37 AM
#3	Success	10:37 AM
#2	Success	10:37 AM
#1	Success	10:37 AM

**Conclusion:** Thus, we have successfully studied Continuous Integration and installed, configured, and understood programming with Jenkins.