

[illegible]

<b>Name</b>	Django WebApp
<b>URL</b>	<a href="https://www.attackdefense.com/challengedetails?cid=2039">https://www.attackdefense.com/challengedetails?cid=2039</a>
<b>Type</b>	DevOps Basics: Continuous Integration

**Important Note:** This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

## Challenge Description

The continuous Integration development process dictates that developers push the code into the master development repository frequently and each (or a small number of) code pushes can trigger the automated build, tests, and deploy the latest build on the test server.

[Jenkins](#) is an open-source automation server that is used widely for continuous integration.

A Jenkins instance and a Gitlab instance are provided to the user. The source code of a Django Webapp is stored on the Gitlab instance.

**Objective:** Create a Jenkins job to migrate and test the web application!

### Instructions:

- The GitLab server is reachable with the name 'gitlab'
- Gitlab credentials:

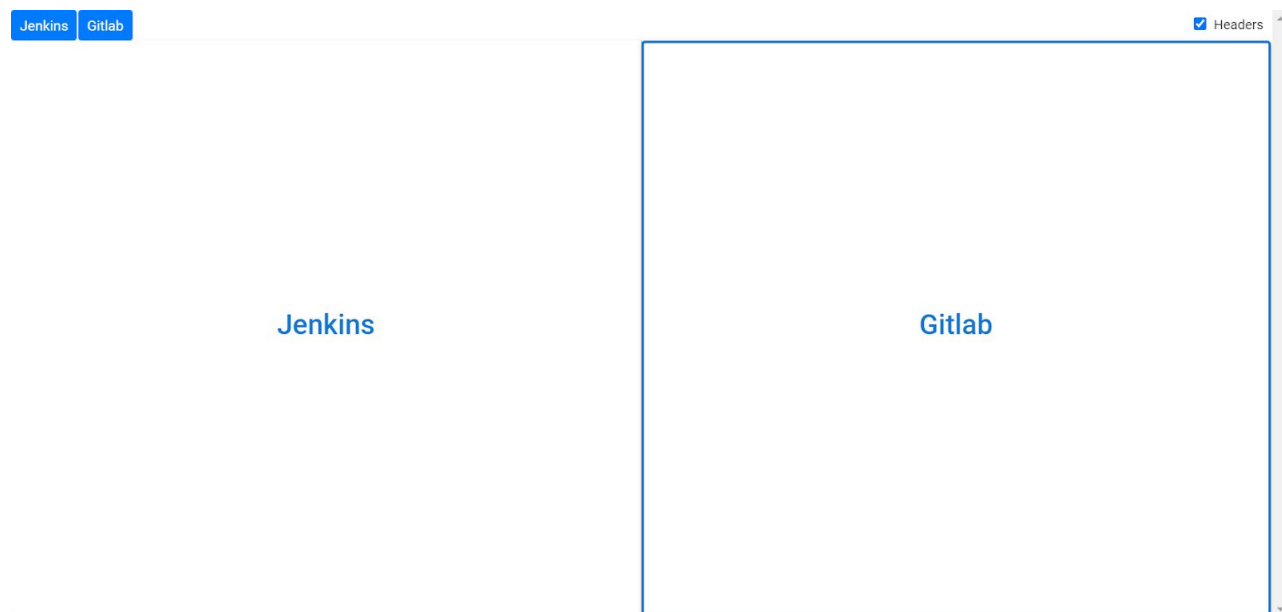
Username	Password
root	welcome123

- The Jenkins server is reachable with the name 'jenkins'
- Jenkins credentials:

Username	Password
admin	welcome123

## Lab Setup

On starting the lab, the following interface will be accessible to the user.



On choosing (clicking the text in the center) left left panel, **Jenkins web UI** will open in a new tab



Welcome to Jenkins!

Sign in

☐ Keep me signed in

On selecting the right panel, a web UI of **Gitlab** will open in a new tab.



## GitLab Community Edition

### Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Sign in	Register
Username or email	
<input type="text"/>	
Password	
<input type="password"/>	
<input type="checkbox"/> Remember me	<a href="#">Forgot your password?</a>
Sign in	

The GitLab instance takes time to function properly. Hence, for some time the following wait page might be visible. This page reloads automatically.

# PENTESTER ACADEMY

WebApp will appear once deployed!

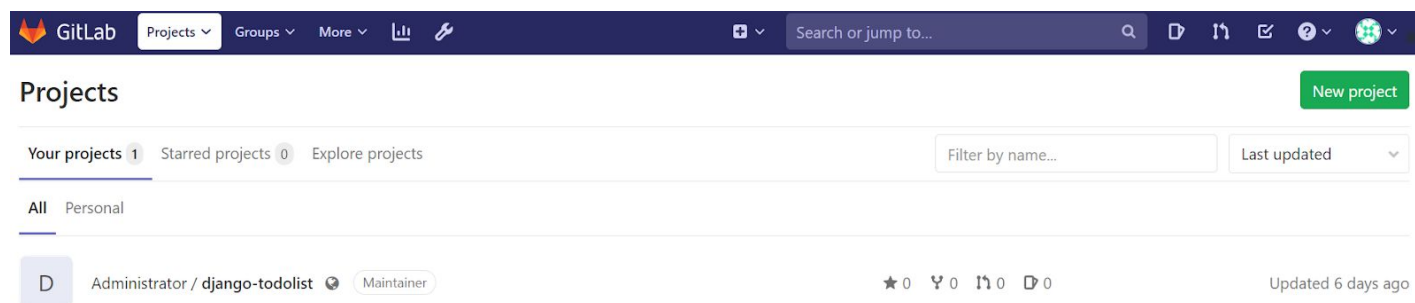
This page refreshes automatically.

## Solution

**Step 1:** Login into GitLab instance using the provided credentials:

Username: root

Password: welcome123



GitLab Projects Groups More Search or jump to...

### Projects

New project

Your projects 1 Starred projects 0 Explore projects Filter by name... Last updated

All Personal

D Administrator / django-todolist Maintainer ★ 0 ♀ 0 🗨 0 📄 0 Updated 6 days ago

There is a repository present in the Administrator's account. Click on the repository link to open the repository page.

GitLab Projects Groups More Search or jump to...

Administrator > django-todolist > Details

Profile was successfully updated

**django-todolist** Project ID: 2

27 Commits 1 Branch 0 Tags 287 KB Files

master django-todolist / + History Find file Web IDE Clone

Update stop-containers.sh Administrator authored 6 days ago 1a0c56a1

README MIT License Add CHANGELOG Add CONTRIBUTING Enable Auto DevOps Add Kubernetes cluster

Set up CI/CD

Name	Last commit	Last update
------	-------------	-------------

**Step 2:** Login into Jenkins instance using the provided credentials:

Username: admin

Password: welcome123

Jenkins search ? monitor 1 admin log out

Jenkins

New Item People Build History Manage Jenkins My Views Lockable Resources New View

**Welcome to Jenkins!**

Please [create new jobs](#) to get started.

**Build Queue**

No builds in the queue.

**Build Executor Status**

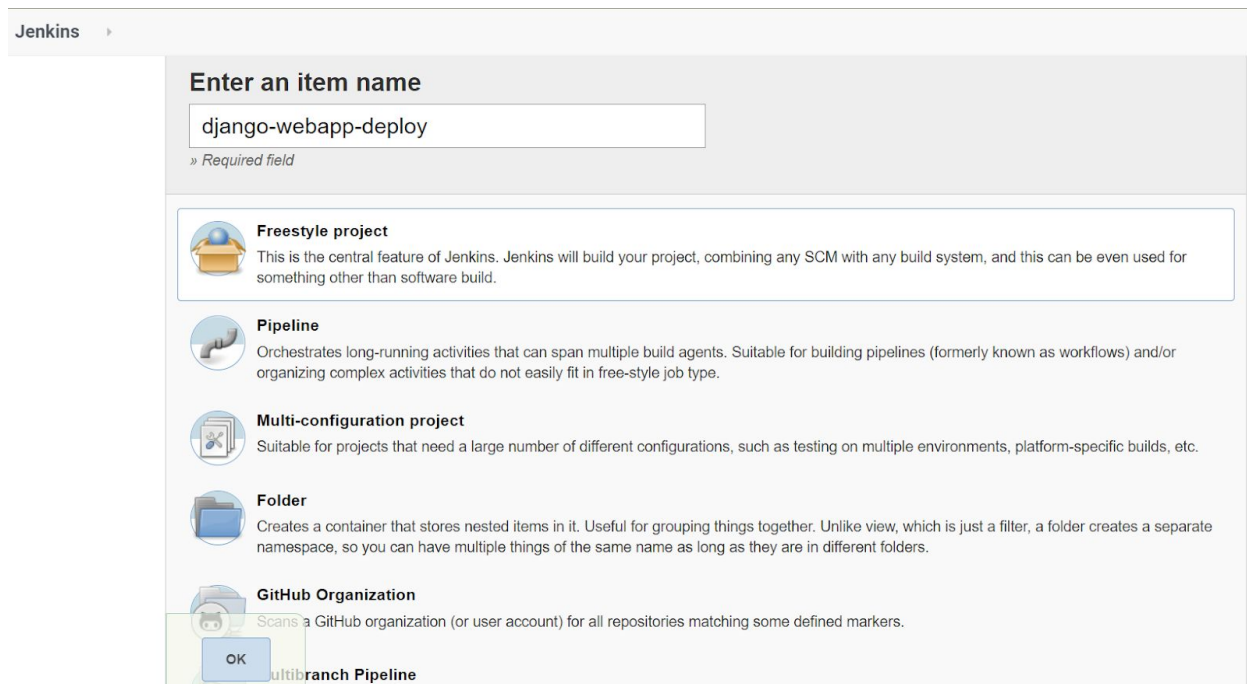
1 Idle  
2 Idle



There is no job in this Jenkins server.

**Step 3:** Click on the “create new jobs” link. A job creation wizard will run. Enter the name of the job on the first page. You are free to select any name you like. We are using “django-webapp-deploy” because this project doesn’t really need to be built but only to checked/tested that it is ready for deployment.

Then select the “Freestyle project” option and click the “OK” button.



Jenkins

### Enter an item name

» 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**  
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

OK Multibranch Pipeline

On the next page, enter some description of the job.

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

Description Job to test Django Webapp

[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

[Advanced...](#)

[Save](#) [Apply](#)

**Step 4:** Open the Django project repository page on GitLab and copy the “Clone with HTTP” path.

**django-todolist** Project ID: 2

☆ Star 0

27 Commits 1 Branch 0 Tags 287 KB Files

master django-todolist

History Find file Clone

Update stop-containers.sh  
Administrator authored 1 week ago

README MIT License

Name Last commit Last update

Clone with SSH  
git@k1vv6pc7kfww1jrpi6uww6q77.c

Clone with HTTP  
http://k1vv6pc7kfww1jrpi6uww6q7

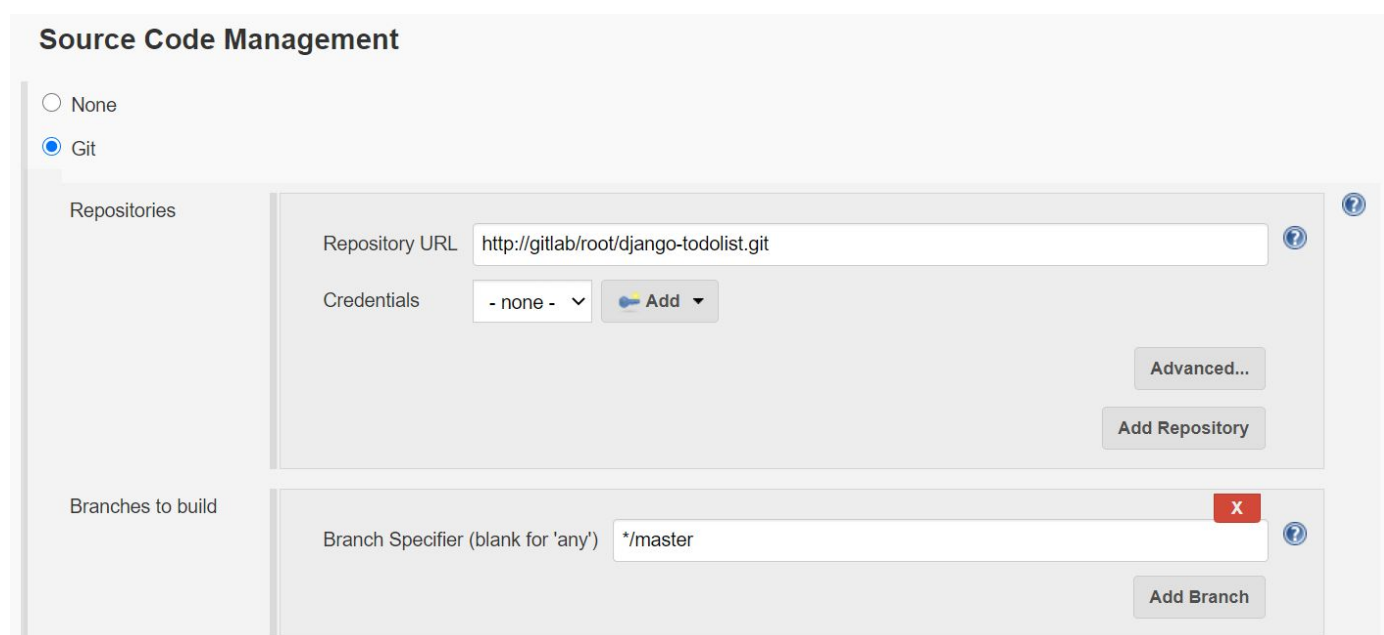
**Step 5:** Select “Git” in the Source Code Management section and paste this path in the “Repository URL” field.



The copied URL will look something like this:  
`http://<random_server_URL>/root/django-todolist.git`

This URL is to make sure that the GitLab files can be accessed through web UI. However, to clone it to the Jenkins server, change it to:

`http://gitlab/root/django-todolist.git`



The screenshot shows the 'Source Code Management' configuration page in Jenkins. On the left, there are two sections: 'Repositories' and 'Branches to build'. In the 'Repositories' section, the 'Git' radio button is selected. The 'Repository URL' field contains 'http://gitlab/root/django-todolist.git'. The 'Credentials' dropdown is set to '- none -' with an 'Add' button next to it. There are 'Advanced...' and 'Add Repository' buttons. In the 'Branches to build' section, the 'Branch Specifier (blank for \'any\')' field contains '\*/master'. There is a red 'X' icon and an 'Add Branch' button.

**Step 6:** In the build section, select “Execute shell”. Run migrate and test on the webapp. This is to test if the webapp is ready to be deployed. Paste the following commands in the build step:

### Commands

```
python3 manage.py migrate  
python3 manage.py test
```

## Build

### Execute shell

Command `python3 manage.py migrate`  
`python3 manage.py test`

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

Advanced...

Add build step ▾

### Post-build Actions

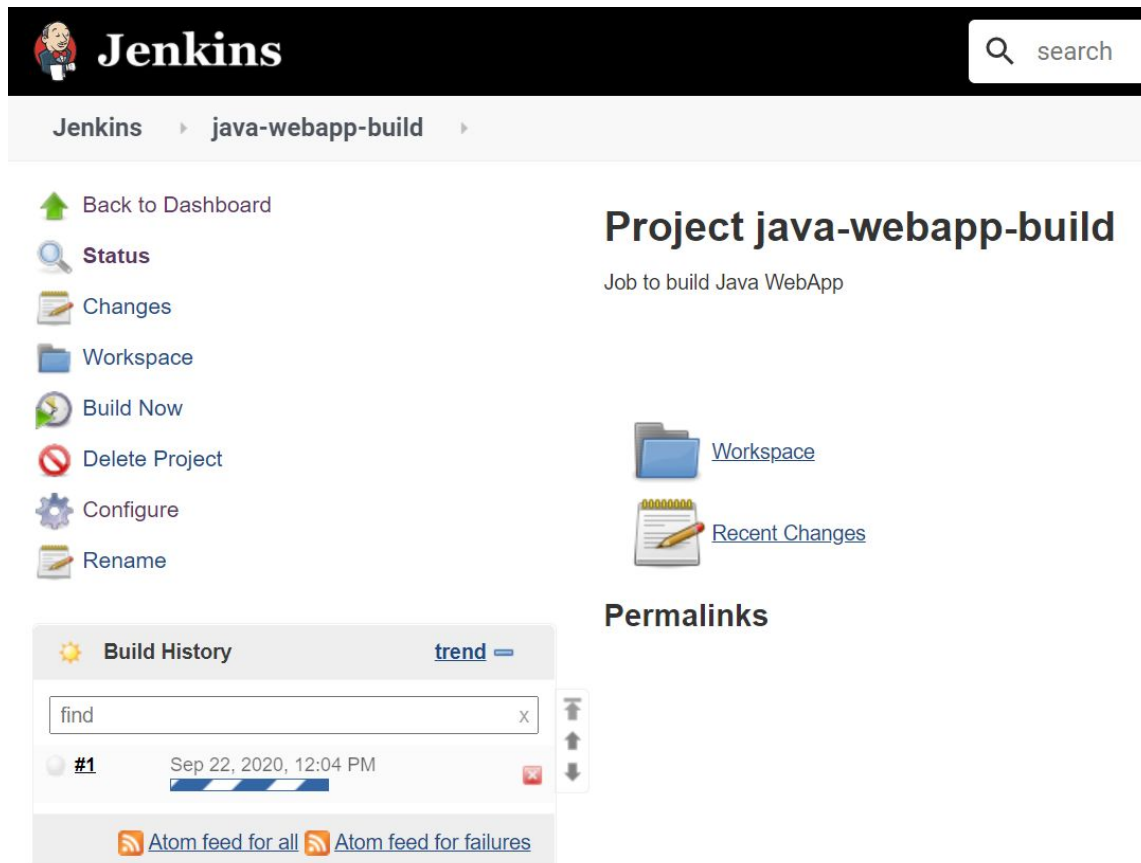
Save

Apply

On clicking the “Save” button, the job will be saved and the page will redirect to the Job page.

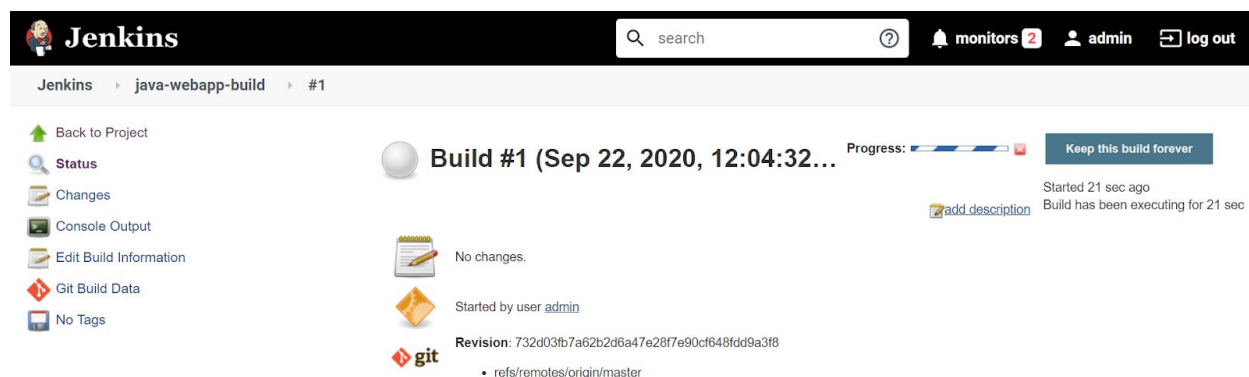
The screenshot shows the Jenkins web interface. At the top is a black header with the Jenkins logo, the word "Jenkins", a search bar, and a help icon. Below the header is a breadcrumb trail: "Jenkins" > "django-webapp-deploy". The main content area is divided into two columns. The left column contains a sidebar menu with icons and labels for: "Back to Dashboard", "Status", "Changes", "Workspace", "Build Now", "Delete Project", "Configure", and "Rename". The right column displays the job title "Project django-webapp-deploy" with the subtitle "Job to deploy Django Webapp". Below this are two links: "Workspace" (with a folder icon) and "Recent Changes" (with a notepad icon). Further down is a "Permalinks" section. At the bottom left, there is a "Build History" section with a search bar containing the text "find" and two RSS feed links: "Atom feed for all" and "Atom feed for failures".

**Step 8:** Click “Build now” from the left-hand menu to fire the job.



The screenshot shows the Jenkins web interface for a project named 'java-webapp-build'. The top navigation bar includes the Jenkins logo, a search bar, and a breadcrumb trail: 'Jenkins > java-webapp-build'. On the left, a sidebar contains links: 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build Now', 'Delete Project', 'Configure', and 'Rename'. The main content area is titled 'Project java-webapp-build' with the subtitle 'Job to build Java WebApp'. It features links for 'Workspace' and 'Recent Changes'. Below this is a 'Permalinks' section. A 'Build History' widget shows a table with one entry: '#1' with a status of 'In Progress' (indicated by a blue and white striped bar) and a timestamp of 'Sep 22, 2020, 12:04 PM'. Below the table are links for 'Atom feed for all' and 'Atom feed for failures'.

An in-progress will appear under the “Build History”. Click on the number #1 to visit the build page.



The screenshot shows the Jenkins build detail page for 'Build #1 (Sep 22, 2020, 12:04:32...)'. The top navigation bar includes the Jenkins logo, a search bar, and links for 'monitors 2', 'admin', and 'log out'. The breadcrumb trail is 'Jenkins > java-webapp-build > #1'. On the left, a sidebar contains links: 'Back to Project', 'Status', 'Changes', 'Console Output', 'Edit Build Information', 'Git Build Data', and 'No Tags'. The main content area shows the build progress bar (blue and white striped) and a 'Keep this build forever' button. Below the progress bar, there is a 'No changes' message, a 'Started by user admin' message, and a 'Revision: 732d03fb7a62b2d6a47e28f7e90cf648fdd9a3f8' message. The revision is linked to 'refs/remotes/origin/master'. There is also a 'add description' button.

**Step 9:** Click on “Console Output” to view the build logs.



Back to Project

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#1'

Git Build Data

No Tags



## Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/java-webapp-build
No credentials specified
Cloning the remote Git repository
Cloning repository http://gitlab/root/java-mvn-hello-world-web-app.git
> git init /var/lib/jenkins/workspace/java-webapp-build # timeout=10
Fetching upstream changes from http://gitlab/root/java-mvn-hello-world-web-app.git
> git --version # timeout=10
> git --version # 'git version 2.25.1'
> git fetch --tags --force --progress -- http://gitlab/root/java-mvn-hello-world-web-app.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
```

```
[django-webapp-deploy] $ /bin/sh -xe /tmp/jenkins4620753108026315603.sh
```

```
+ python3 manage.py migrate
```

```
Operations to perform:
```

```
Apply all migrations: admin, auth, contenttypes, lists, sessions
```

```
Running migrations:
```

```
Applying contenttypes.0001_initial... OK
Applying auth.0001_initial... OK
Applying admin.0001_initial... OK
Applying admin.0002_logentry_remove_auto_add... OK
Applying admin.0003_logentry_add_action_flag_choices... OK
Applying contenttypes.0002_remove_content_type_name... OK
Applying auth.0002_alter_permission_name_max_length... OK
Applying auth.0003_alter_user_email_max_length... OK
Applying auth.0004_alter_user_username_opts... OK
Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_add_error_messages... OK
Applying auth.0008_alter_user_username_max_length... OK
Applying auth.0009_alter_user_last_name_max_length... OK
Applying auth.0010_alter_group_name_max_length... OK
Applying auth.0011_update_proxy_permissions... OK
```



```
+ python3 manage.py test
Creating test database for alias 'default'...
.....
-----
Ran 54 tests in 6.098s

OK
Destroying test database for alias 'default'...
System check identified no issues (0 silenced).
Finished: SUCCESS
```

The job completed successfully after the successful execution of migrate and test commands.

## Learning

Testing the Django web app using Jenkins.