# ATTACK DEFENSE

by PentesterAcademy

|  |  |
|---|---|
| **Name** | Fix the App: Django WebApp II |
| **URL** | https://www.attackdefense.com/challengedetails?cid=2273 |
| **Type** | Pipeline Basics: Web Applications |

**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

DevOps practices are to combine software development (Dev) and IT operations (Ops) in order to improve the delivery process. DevOps pipelines are chained tasks and components that run in a sequence to cover different phases of software compilation, packaging, automated testing, and test deployment.

In this lab, we have a DevSecOps pipeline for a Django web application. The pipeline consists of the following components (and tasks):

- VS Code Server (For pulling, modifying, and pushing the code)
- GitLab server (For hosting code)
- Jenkins server (For integrating). Different phases and components used:
  - Build: Django
  - Code testing: Django
  - Test Deployment: Ansible
  - Dynamic Testing: Selenium
- Test server (For test deployment)
- Archery Sec server (For Vulnerability management)

It is suggested to play the DevOps focused lab before playing this lab.

DevSecOps refer to introducing security in different stages of the DevOps process. This is done to catch the vulnerabilities/insecurities as soon as possible in the pipeline. In this lab, the pipeline consists of the following components (and tasks):

| | |
|---|---|
| Automated Code Review: | DevSkim |
| Sensitive Information Scan phase: | Truffle Hog |
| Software Component Analysis: | Safety |
| Static Code Analysis: | Bandit |
| Dynamic Application Security Testing: | OWASP ZAP |
| Compliance as Code: | Inspec |

**Objective:** Fix the Issues in the stages of the pipeline and Find the flags!

**Instructions:**

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

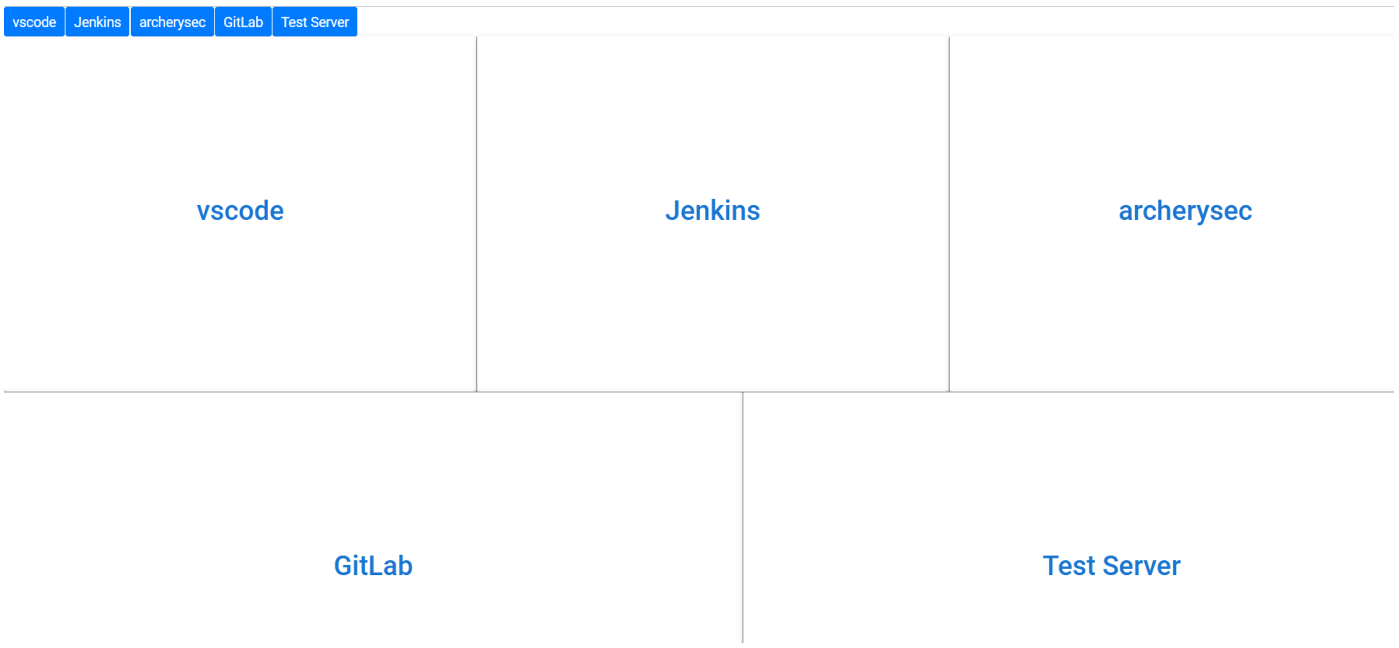
| Username | Password |
|---|---|
| root | welcome123 |

- The Archery server is reachable by the name "archerysec"
- ArcherySec credentials:

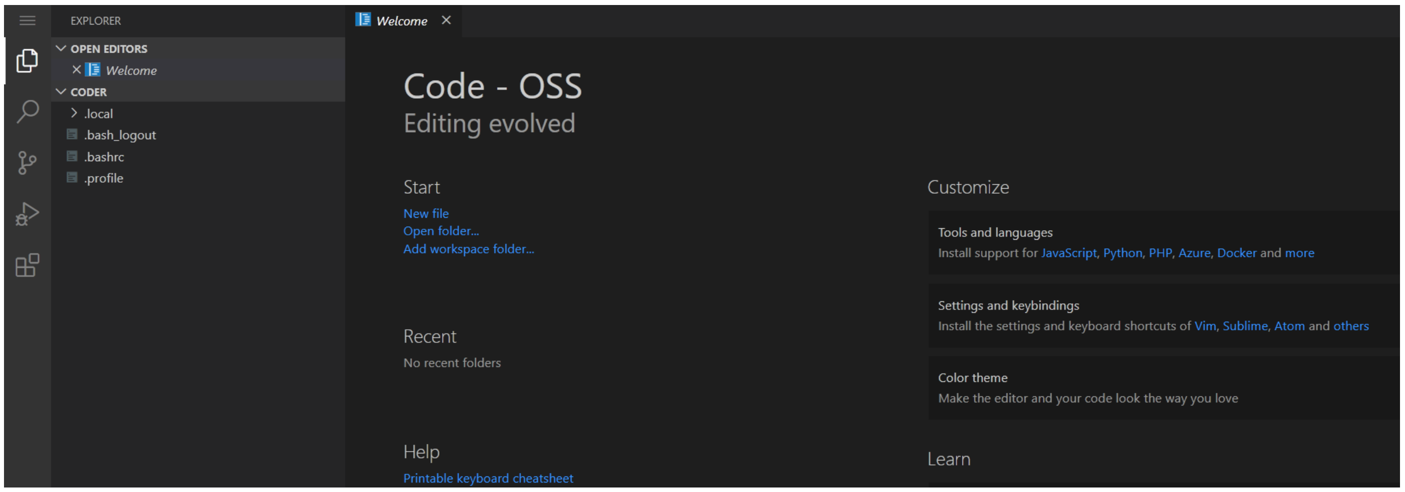
| Username | Password |
|---|---|
| admin | admin |

## Lab Setup

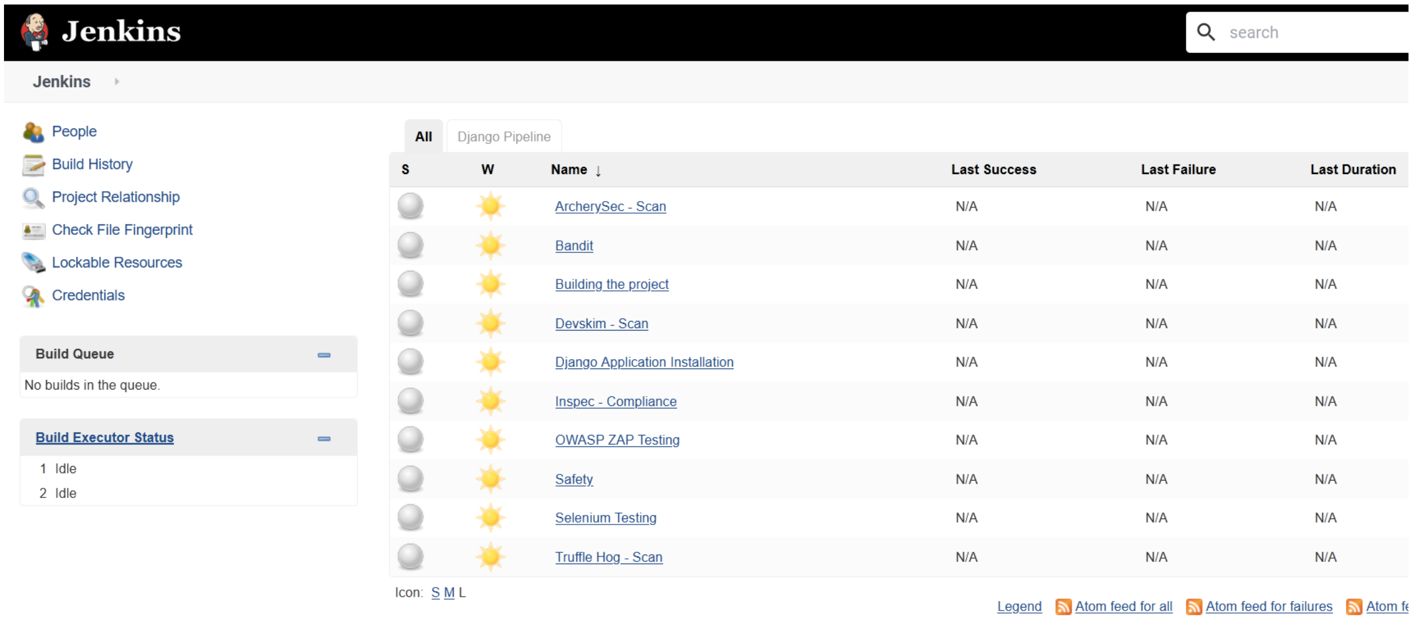
On starting the lab, the following interface will be accessible to the user.



On choosing (clicking the text in the center) top left panel, **vscode** will open in a new tab



Similarly on selecting the top middle panel, a web UI of **Jenkins** will open in a new tab.


Jenkins

search

Jenkins

- People
- Build History
- Project Relationship
- Check File Fingerprint
- Lockable Resources
- Credentials

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

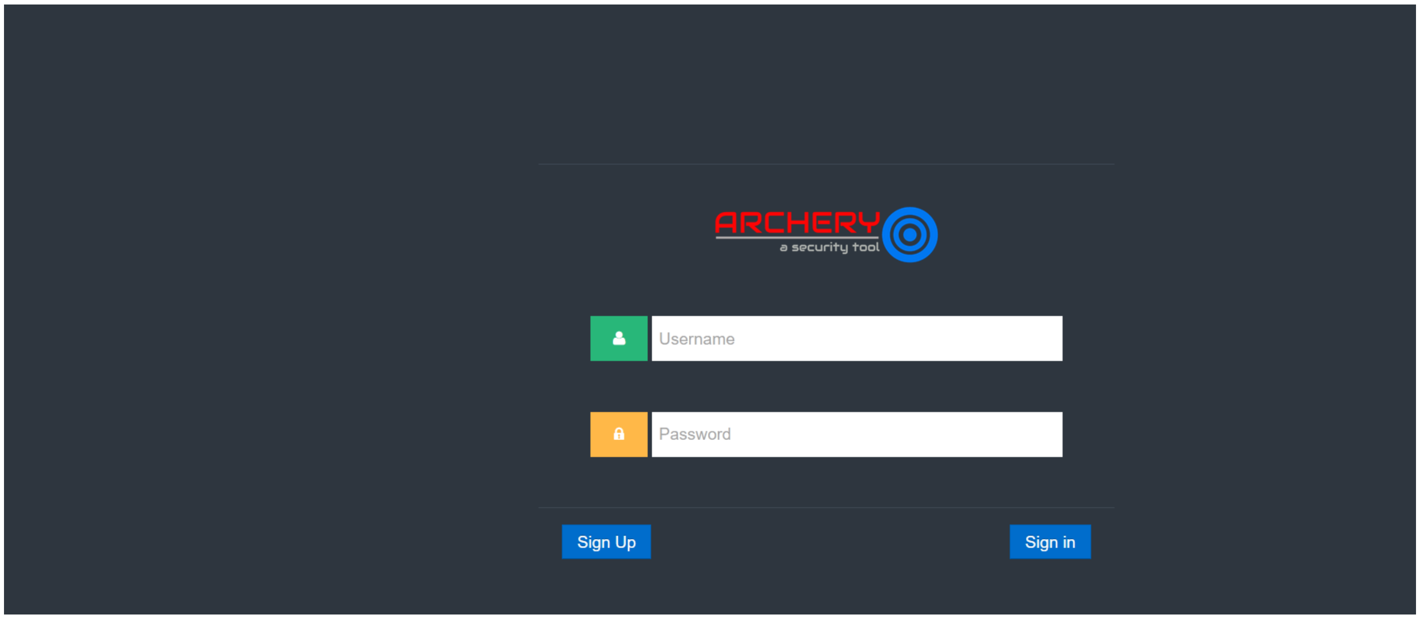
All | Django Pipeline

| S | W | Name ↓ | Last Success | Last Failure | Last Duration |
|---|---|---|---|---|---|
|  |  | ArcherySec - Scan | N/A | N/A | N/A |
|  |  | Bandit | N/A | N/A | N/A |
|  |  | Building the project | N/A | N/A | N/A |
|  |  | Devskim - Scan | N/A | N/A | N/A |
|  |  | Django Application Installation | N/A | N/A | N/A |
|  |  | Inspec - Compliance | N/A | N/A | N/A |
|  |  | OWASP ZAP Testing | N/A | N/A | N/A |
|  |  | Safety | N/A | N/A | N/A |
|  |  | Selenium Testing | N/A | N/A | N/A |
|  |  | Truffle Hog - Scan | N/A | N/A | N/A |

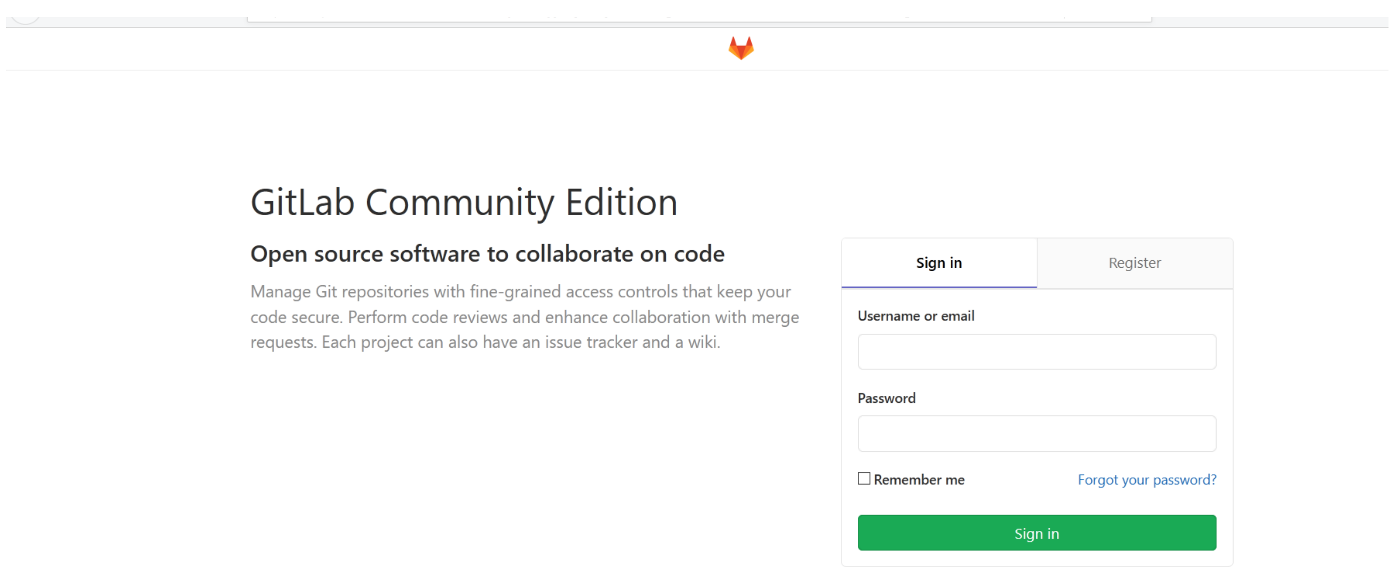
Icon: S M L

Legend Atom feed for all Atom feed for failures Atom fe

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



On selecting the bottom left 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.

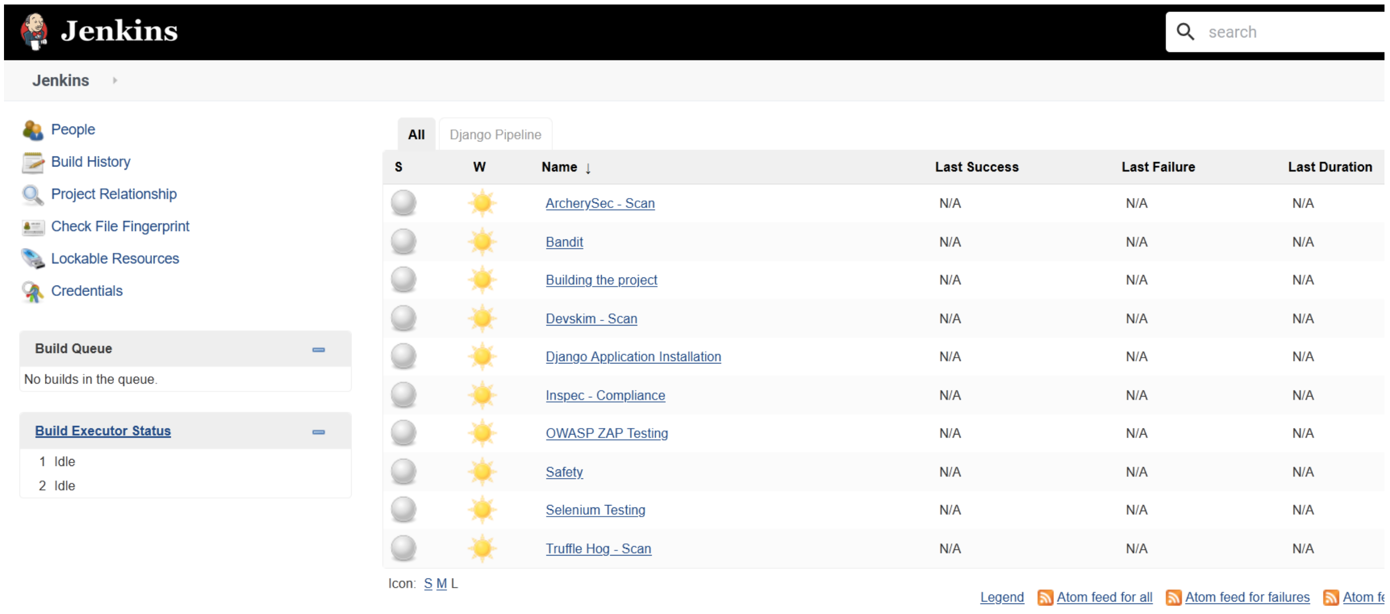
And on selecting the bottom right panel, a web UI of **Test Server** will open in a new tab.

```
Bad Gateway
```

The page will reload until the test-server has started running the web service at port 8000

## Solution

**Step 1:** Open the Jenkins page



Build Queue

No builds in the queue.

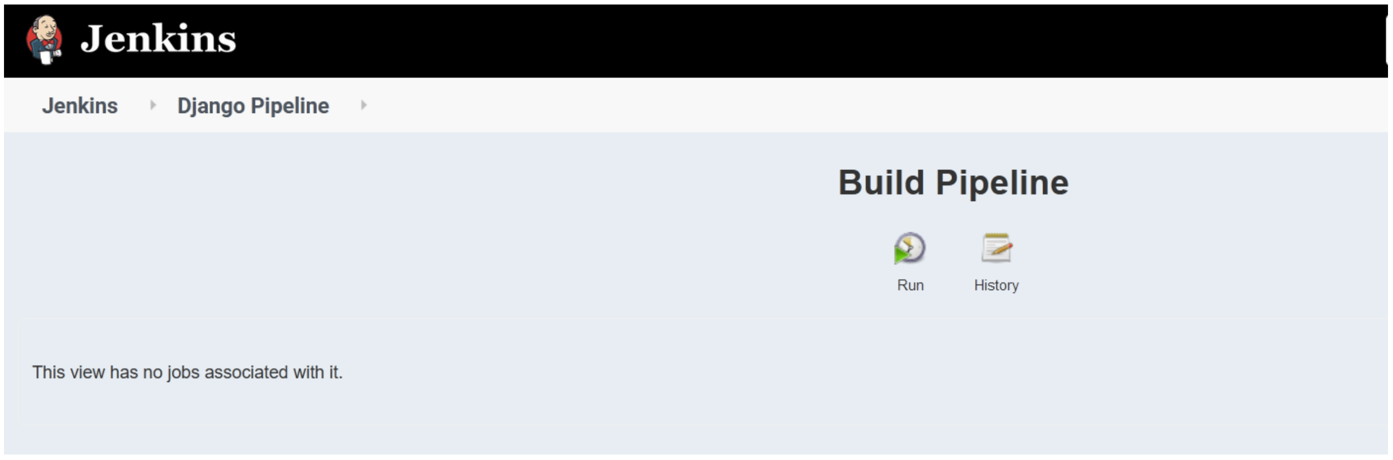
Build Executor Status

1 Idle

2 Idle

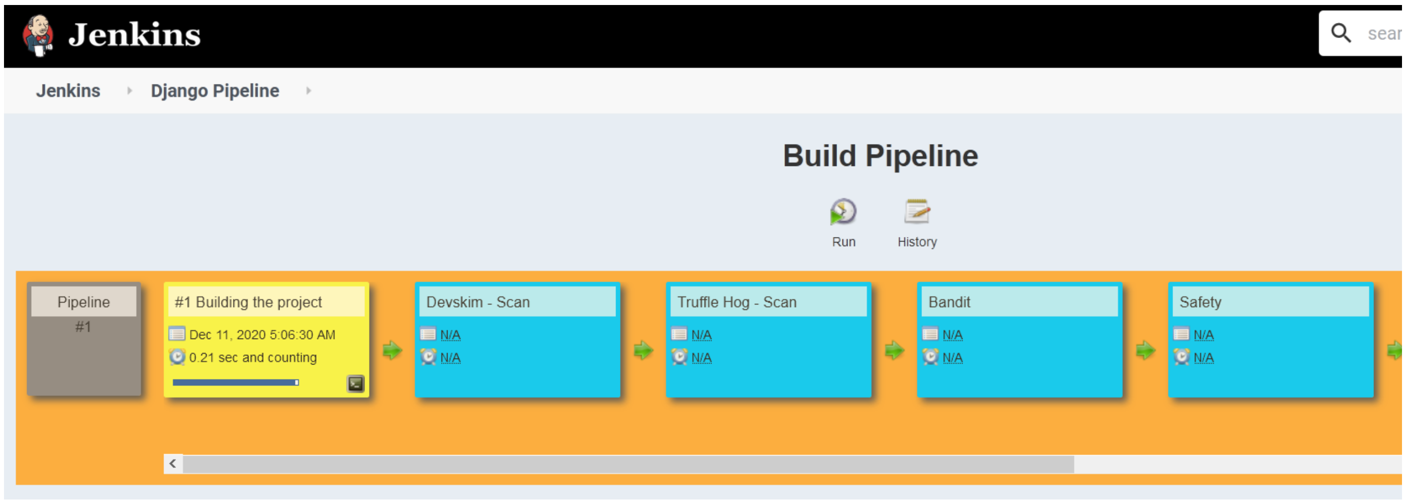
| S | W | Name ↓ | Last Success | Last Failure | Last Duration |
|---|---|---|---|---|---|
| | | ArcherySec - Scan | N/A | N/A | N/A |
| | | Bandit | N/A | N/A | N/A |
| | | Building the project | N/A | N/A | N/A |
| | | Devskim - Scan | N/A | N/A | N/A |
| | | Django Application Installation | N/A | N/A | N/A |
| | | Inspec - Compliance | N/A | N/A | N/A |
| | | OWASP ZAP Testing | N/A | N/A | N/A |
| | | Safety | N/A | N/A | N/A |
| | | Selenium Testing | N/A | N/A | N/A |
| | | Truffle Hog - Scan | N/A | N/A | N/A |

There are 10 jobs present in the Jenkins interface. Navigate to the Django Pipeline view section.

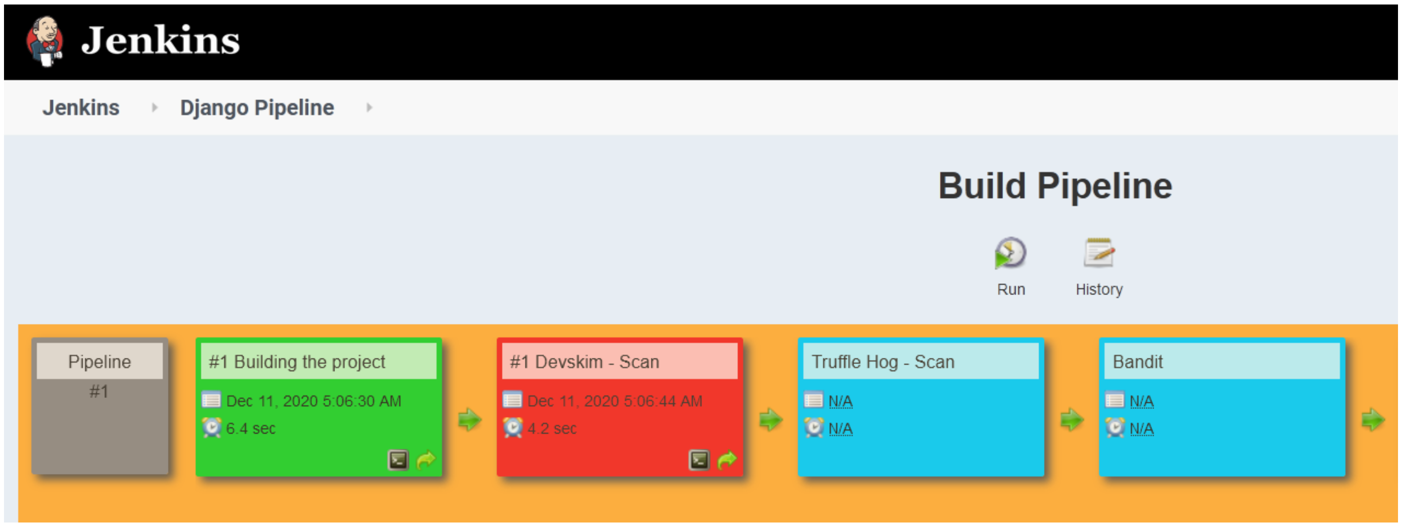


This view has no jobs associated with it.

Click on the Run button to start building the pipeline.



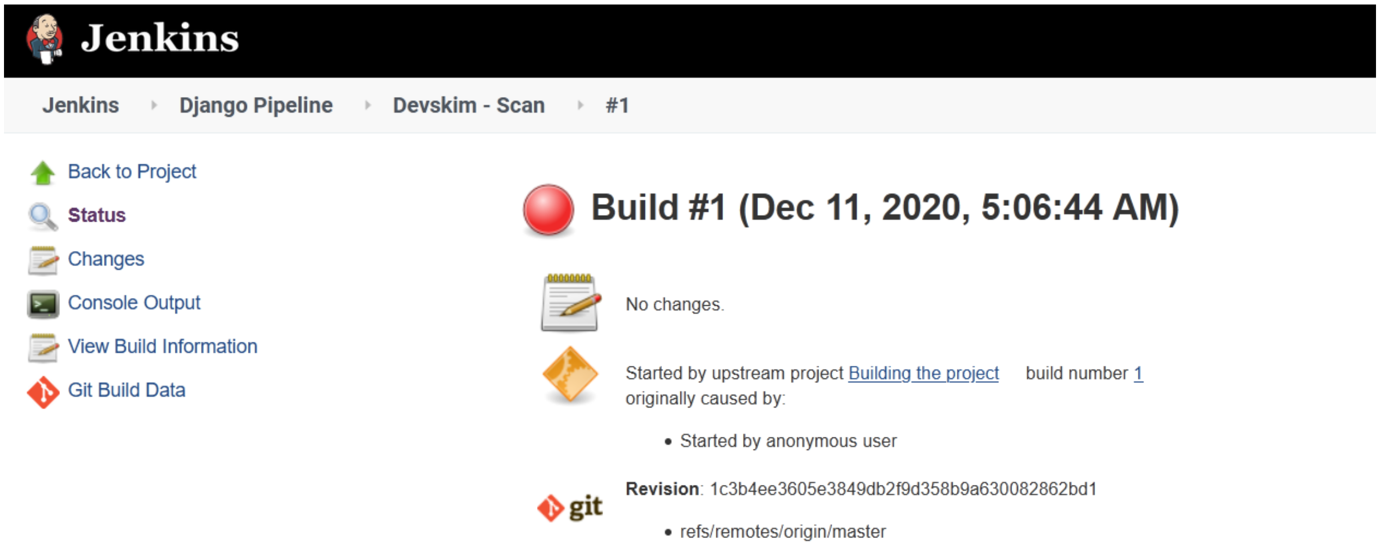
The projects will start building one by one. Keep reloading the page in intervals to check the changes on the page.



The build failed.

**Devskim Issue**

**Step 1:** Click on the ‘Devskim - Scan’ to check the job build page.



Jenkins › Django Pipeline › Devskim - Scan › #1

Back to Project
Status
Changes
Console Output
View Build Information
Git Build Data

**Build #1 (Dec 11, 2020, 5:06:44 AM)**

No changes.

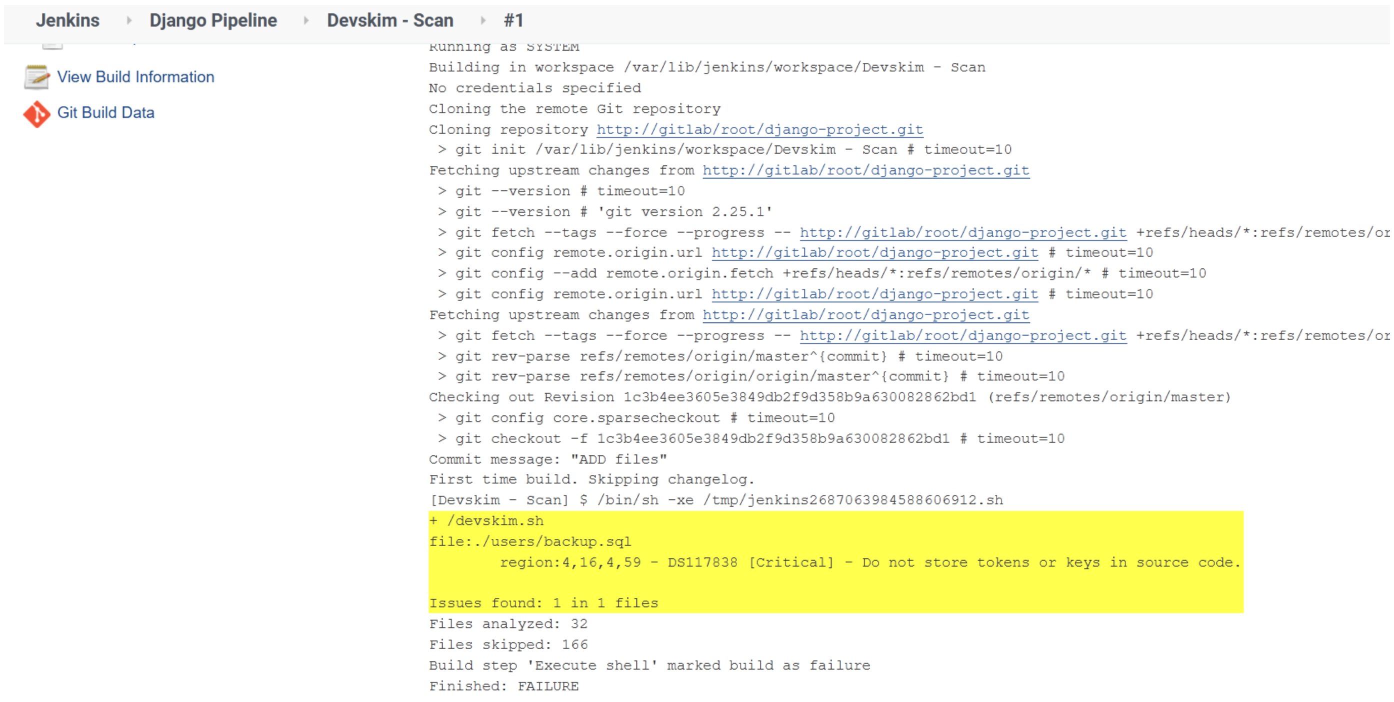
Started by upstream project Building the project build number 1
originally caused by:

- Started by anonymous user

**Revision**: 1c3b4ee3605e3849db2f9d358b9a630082862bd1

- refs/remotes/origin/master

**Step 2:** Click on the "Console Output" to check the issues found by devskim tool.



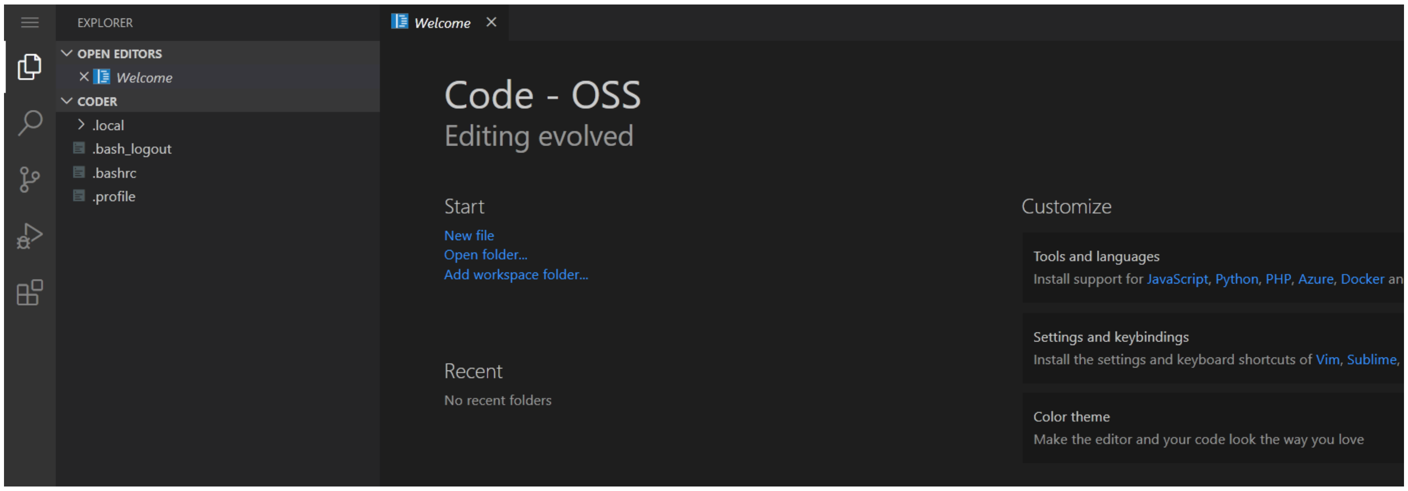
Jenkins › Django Pipeline › Devskim - Scan › #1

View Build Information
Git Build Data

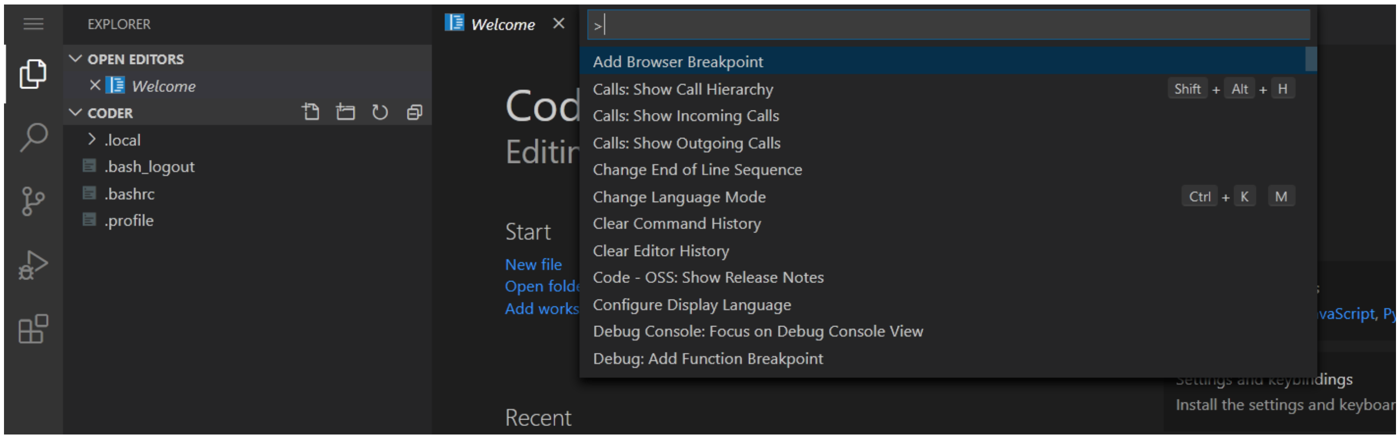
```
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Devskim - Scan
No credentials specified
Cloning the remote Git repository
Cloning repository http://gitlab/root/django-project.git
 > git init /var/lib/jenkins/workspace/Devskim - Scan # timeout=10
Fetching upstream changes from http://gitlab/root/django-project.git
 > git --version # timeout=10
 > git --version # 'git version 2.25.1'
 > git fetch --tags --force --progress -- http://gitlab/root/django-project.git +refs/heads/*:refs/remotes/or
 > git config remote.origin.url http://gitlab/root/django-project.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url http://gitlab/root/django-project.git # timeout=10
Fetching upstream changes from http://gitlab/root/django-project.git
 > git fetch --tags --force --progress -- http://gitlab/root/django-project.git +refs/heads/*:refs/remotes/or
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
 > git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 1c3b4ee3605e3849db2f9d358b9a630082862bd1 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 1c3b4ee3605e3849db2f9d358b9a630082862bd1 # timeout=10
Commit message: "ADD files"
First time build. Skipping changelog.
[Devskim - Scan] $ /bin/sh -xe /tmp/jenkins2687063984588606912.sh
+ /devskim.sh
file:./users/backup.sql
        region:4,16,4,59 - DS117838 [Critical] - Do not store tokens or keys in source code.

Issues found: 1 in 1 files
Files analyzed: 32
Files skipped: 166
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```

The devskim identified hardcoded keys in the users directory.

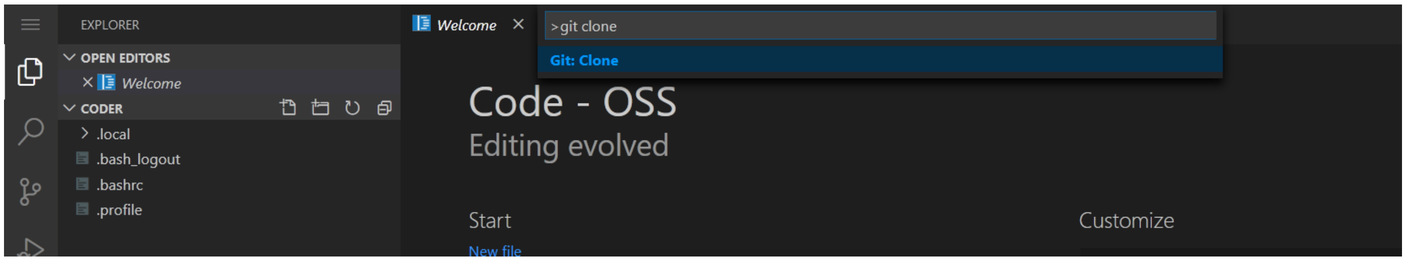
**Step 3:** Open the vscode server page.



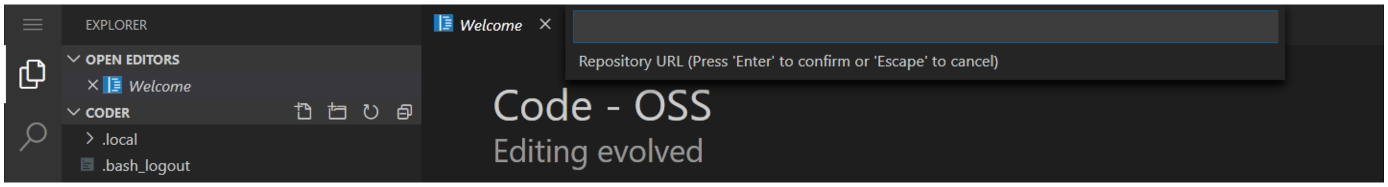
Press CTRL + SHIFT + P to open the command palette or click on the settings bar available in the bottom left and choose the “Command Palette” option.


EXPLORER
OPEN EDITORS
Welcome
CODER
.local
.bash_logout
.bashrc
.profile
Welcome
>
Add Browser Breakpoint
Calls: Show Call Hierarchy Shift + Alt + H
Calls: Show Incoming Calls
Calls: Show Outgoing Calls
Change End of Line Sequence
Change Language Mode Ctrl + K M
Clear Command History
Clear Editor History
Code - OSS: Show Release Notes
Configure Display Language
Debug Console: Focus on Debug Console View
Debug: Add Function Breakpoint
Start
New file
Recent
Install the settings and keyboar

**Step 4:** Enter the command “git clone” in the command palette in order to clone the repository and make changes.



Press enter to choose the Git Clone option.

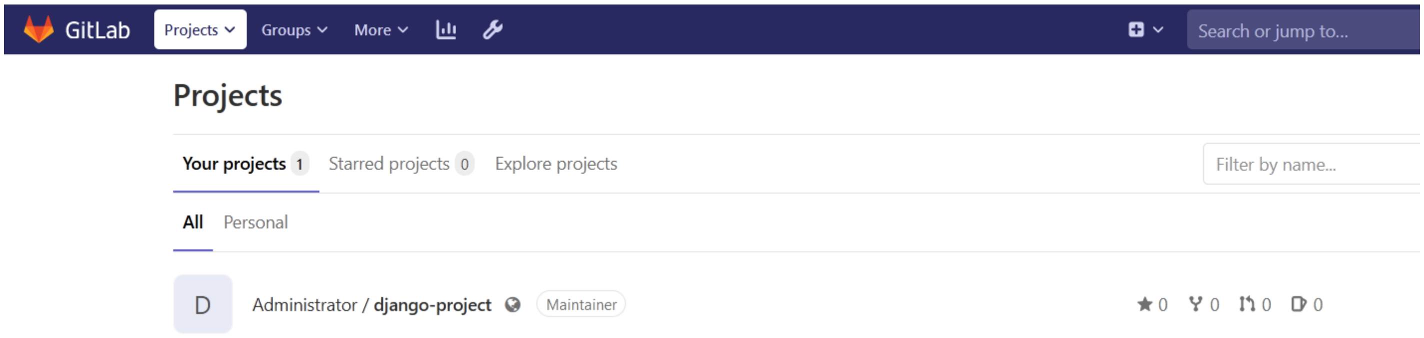


Repository URL (Press 'Enter' to confirm or 'Escape' to cancel)

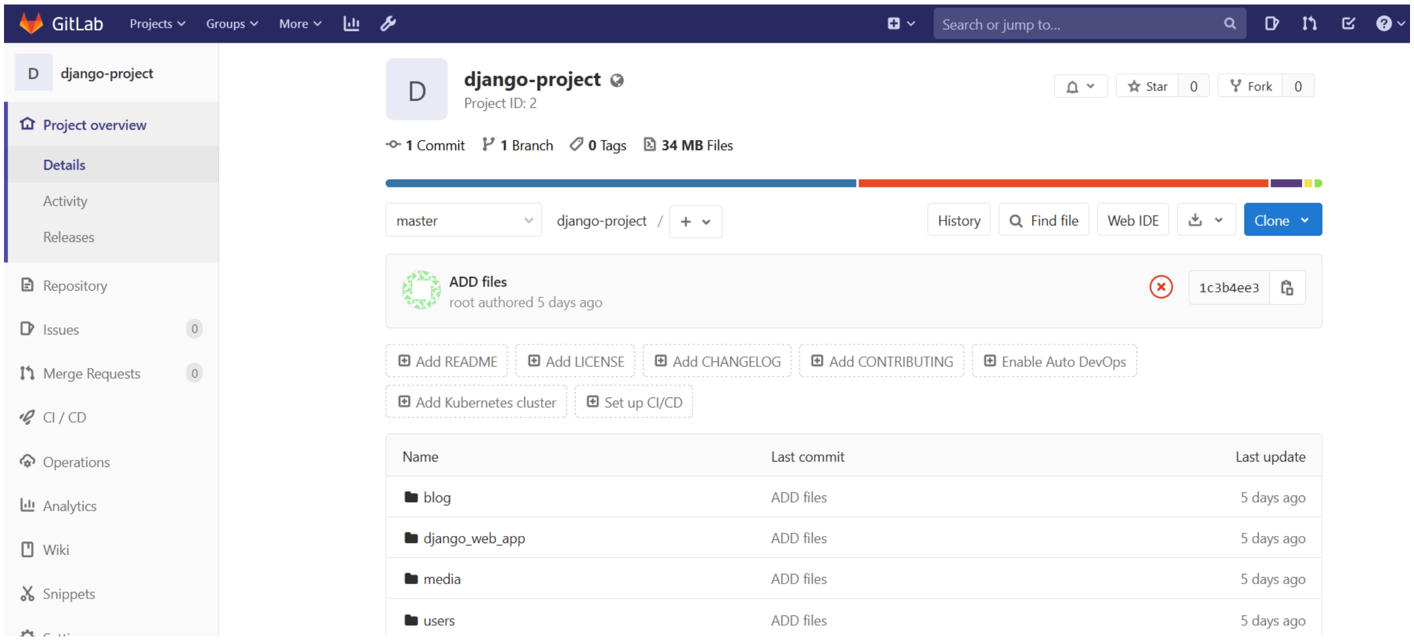
**Step 5:** Open the gitlab page and log in using the credentials provided in the challenge description.

**Credentials:**

- **Username:** root
- **Password:** welcome123



Click on the django-project link to open the repository page.



GitLab Projects Groups More
Search or jump to...

django-project
Project ID: 2
1 Commit · 1 Branch · 0 Tags · 34 MB Files

master · django-project / + · History · Find file · Web IDE · Clone

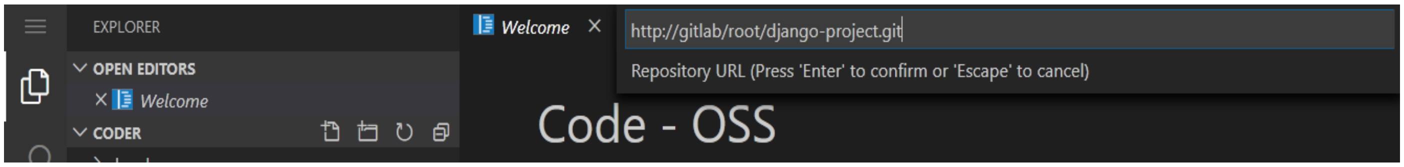
ADD files
root authored 5 days ago · 1c3b4ee3

Add README · Add LICENSE · Add CHANGELOG · Add CONTRIBUTING · Enable Auto DevOps · Add Kubernetes cluster · Set up CI/CD

| Name | Last commit | Last update |
|---|---|---|
| blog | ADD files | 5 days ago |
| django_web_app | ADD files | 5 days ago |
| media | ADD files | 5 days ago |
| users | ADD files | 5 days ago |

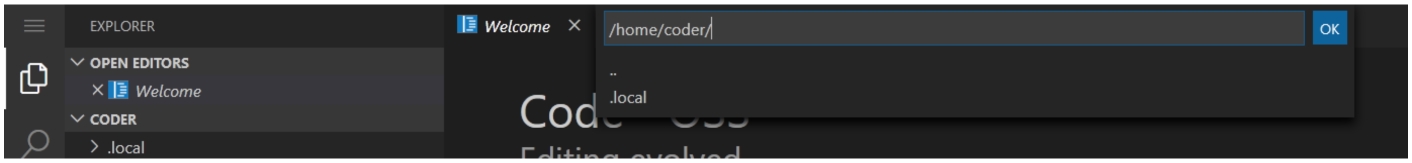
Copy the git link to the GitLab repository.

**Step 6:** Enter the link into the clone function at vscode server.



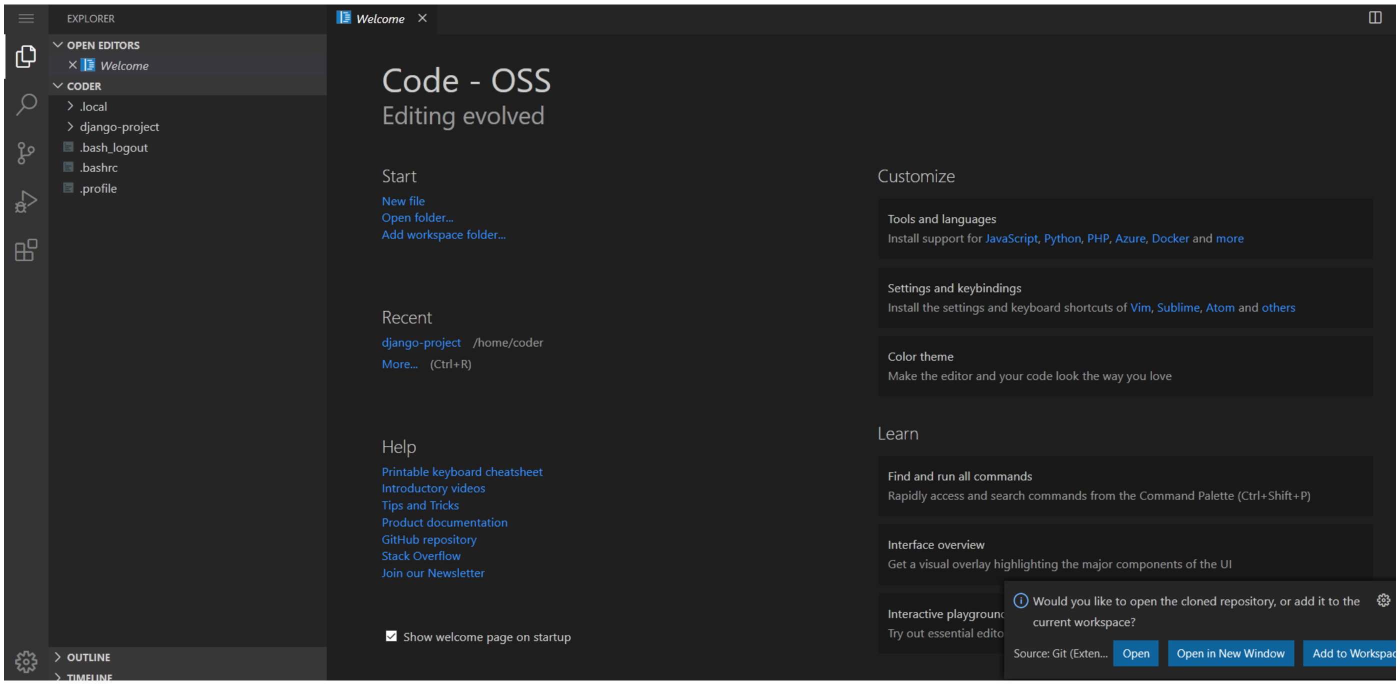
```
http://gitlab/root/django-project.git
Repository URL (Press 'Enter' to confirm or 'Escape' to cancel)
```

Press Enter.



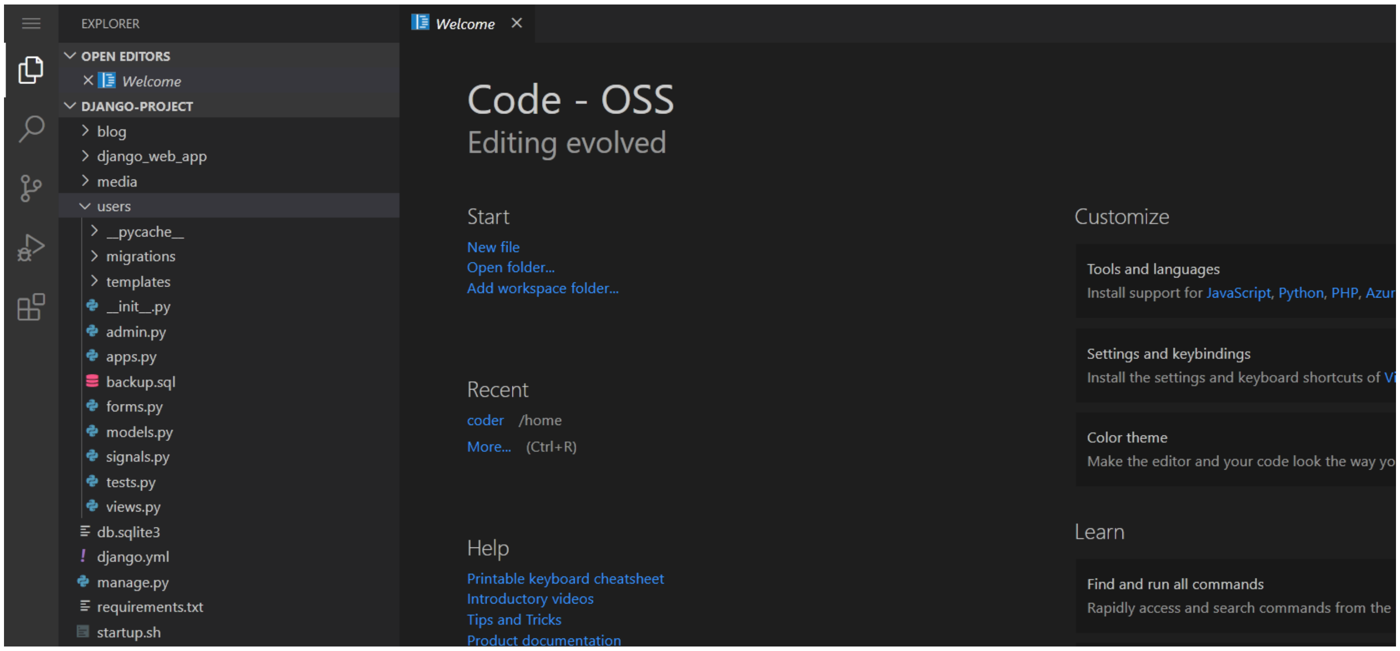
```
/home/coder/
..
.local
```

Choose the path to clone to the repository.

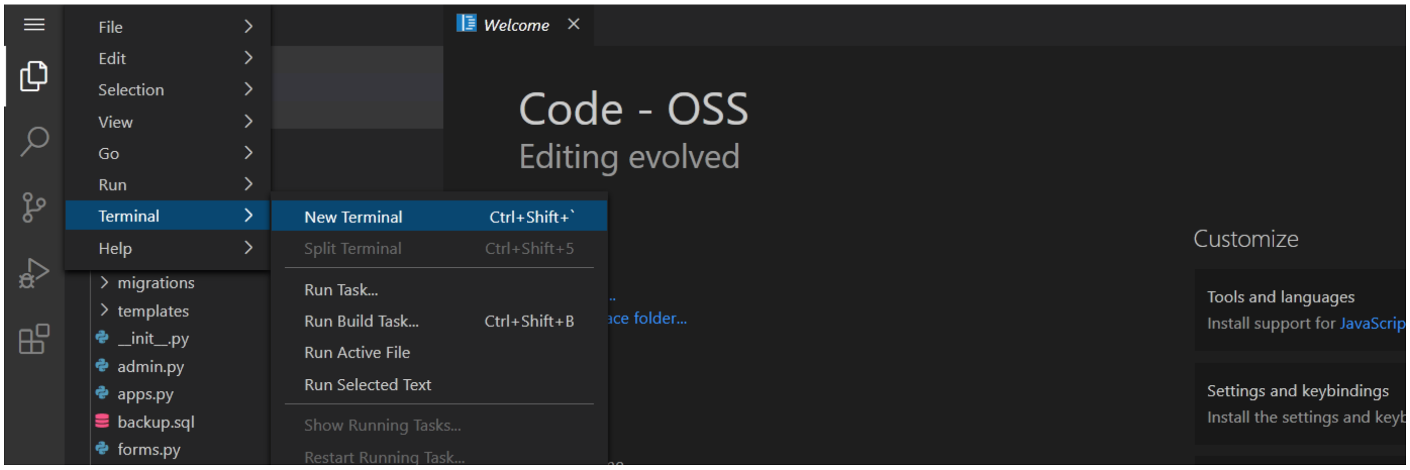


Would you like to open the cloned repository, or add it to the current workspace?

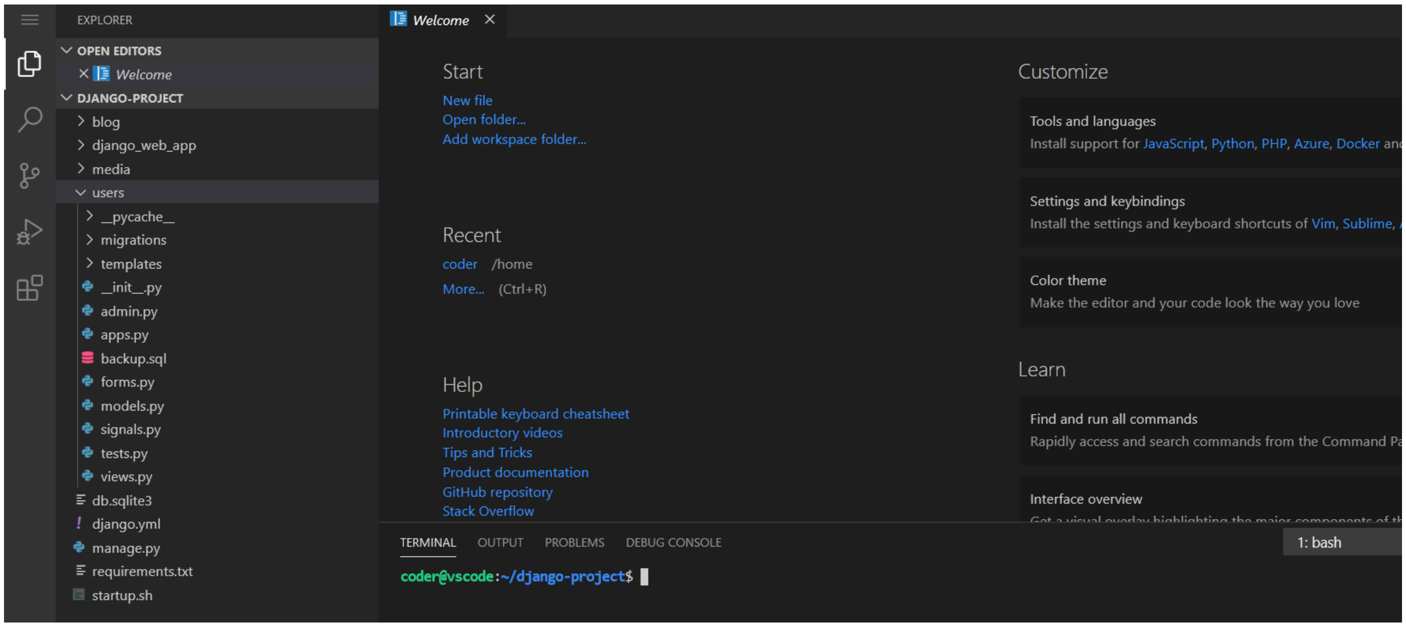
Click on the “Open” button to open the source code directory of the django-project.



**Step 7:** Open the Terminal in the directory.



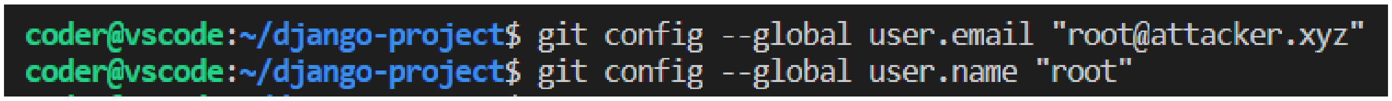
Select the “New Terminal” option.



```
coder@vscode:~/django-project$
```

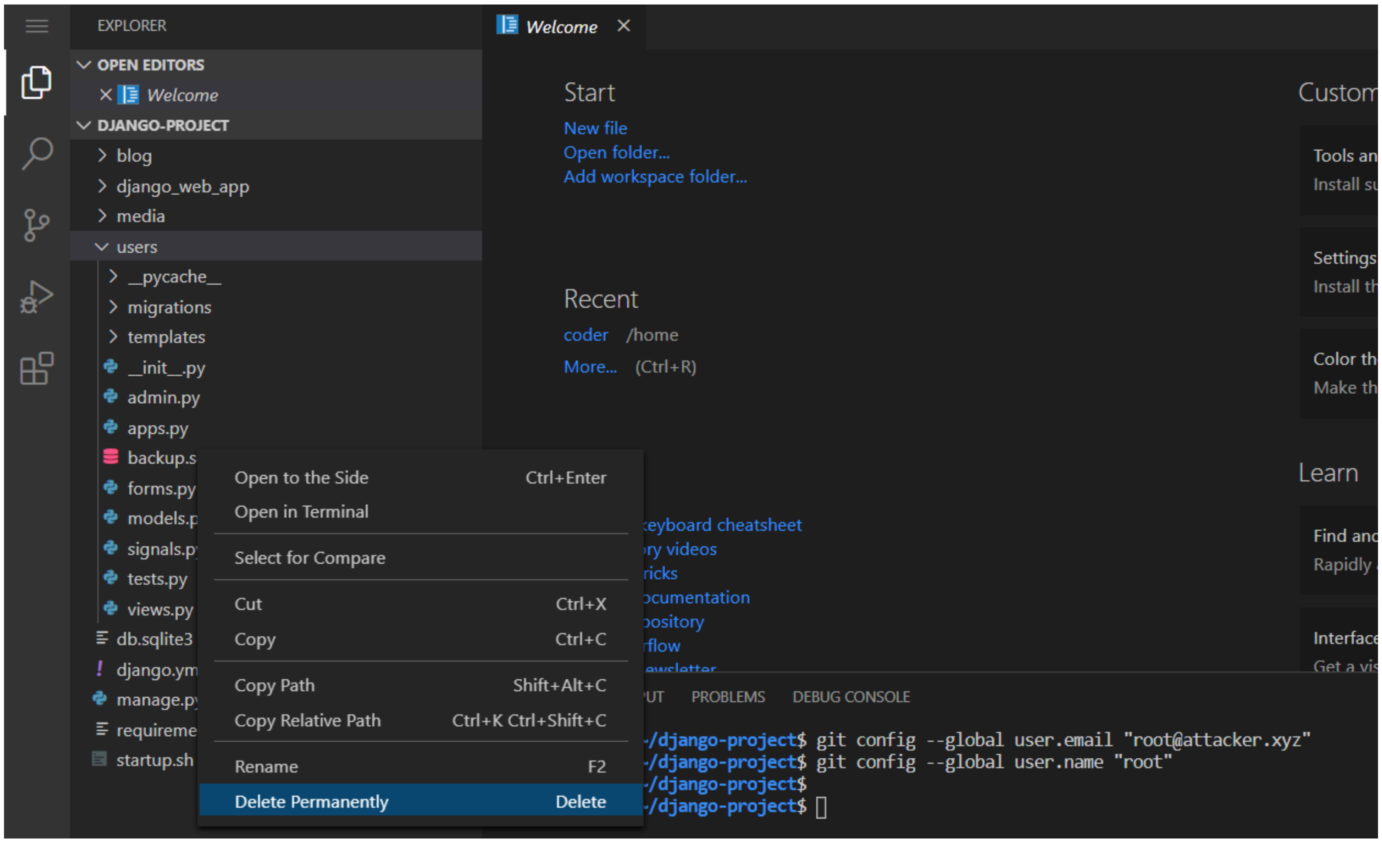
**Step 8:** Enter the git credentials into the terminal to save the identity of the user.

**Commands:**
git config --global user.email "root@attacker.xyz"
git config --global user.name "root"



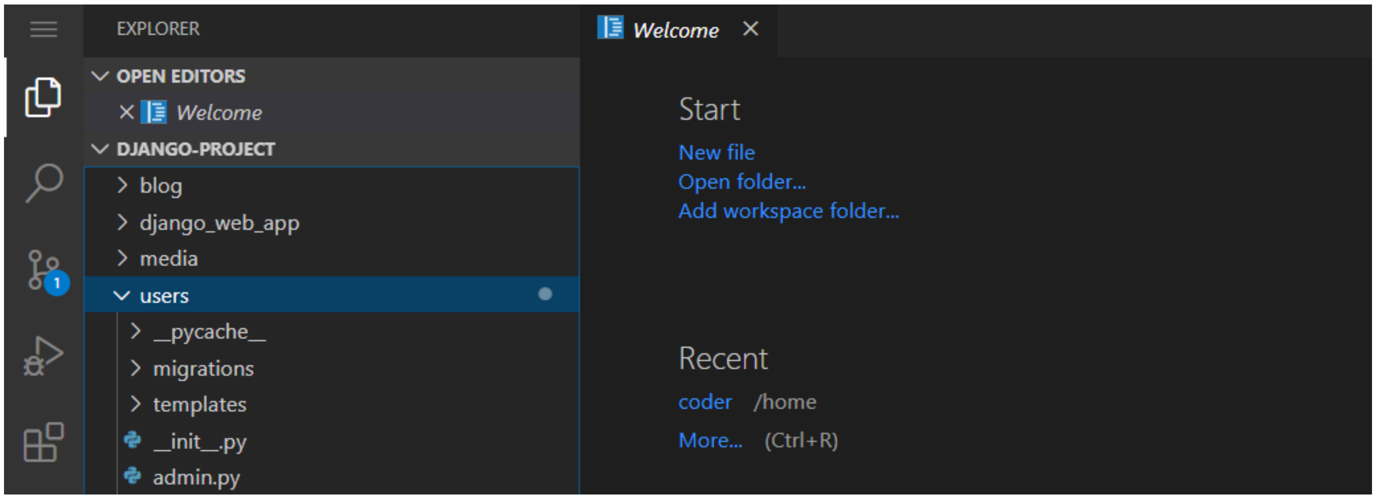
```
coder@vscode:~/django-project$ git config --global user.email "root@attacker.xyz"
coder@vscode:~/django-project$ git config --global user.name "root"
```

**Step 9:** Navigate inside the users directory and right-click on the backup.sql file.

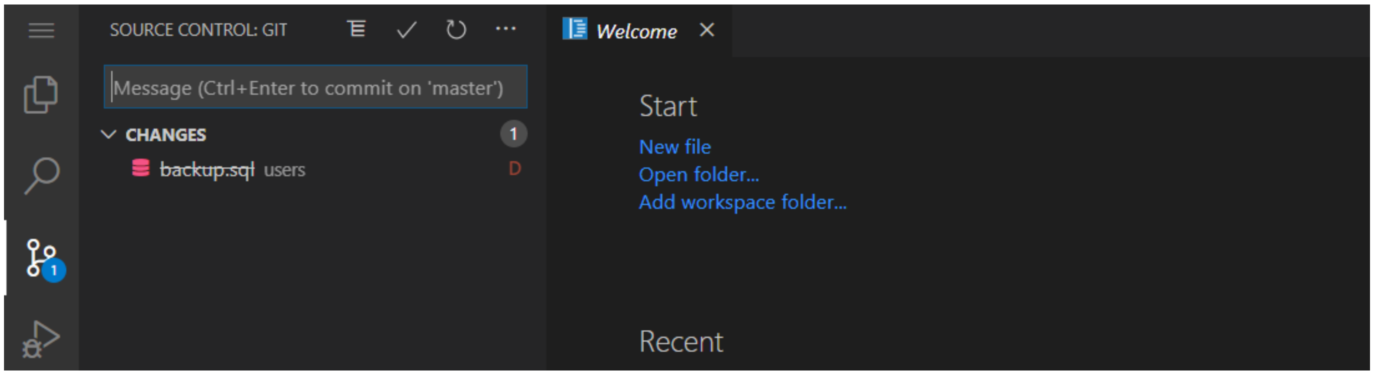


```
/django-project$ git config --global user.email "root@attacker.xyz"
/django-project$ git config --global user.name "root"
/django-project$
/django-project$
```

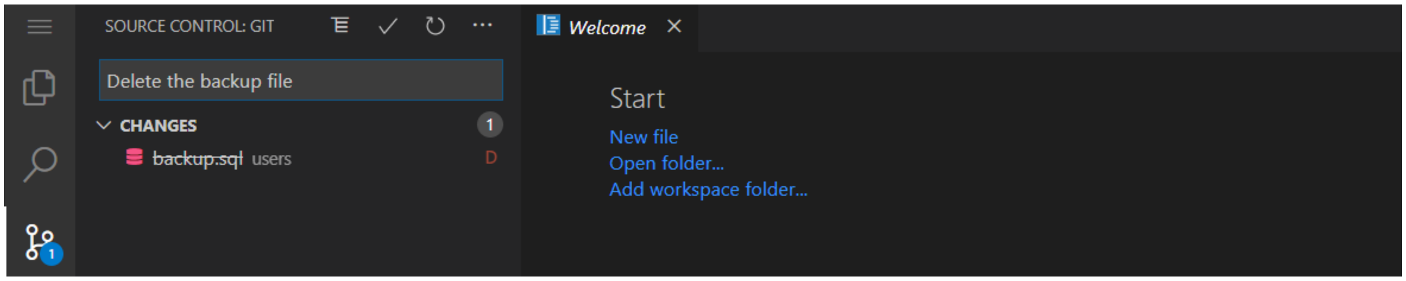
Select “Delete Permanently” to delete the backup.sql file from the project.
(Select ‘Delete’ when prompted)



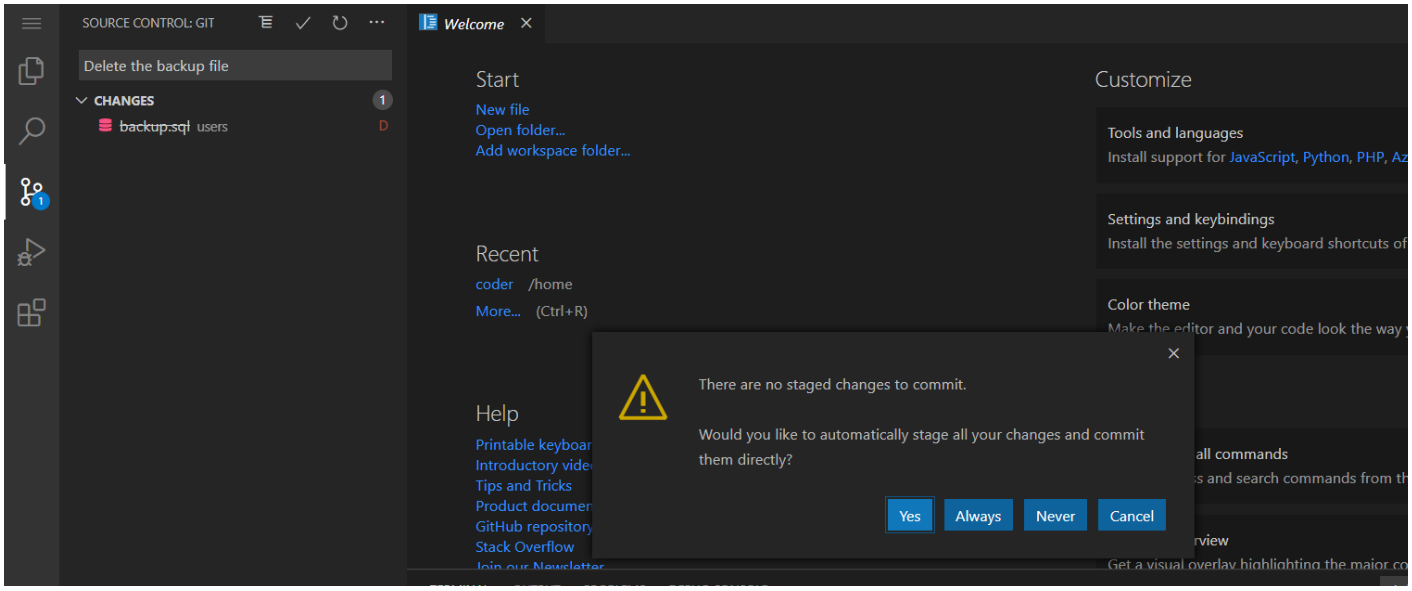
**Step 10:** Navigate to the Source Control section.



Enter a message in the message field to set a comment on the commit.



Click on the Commit button (Tick icon).

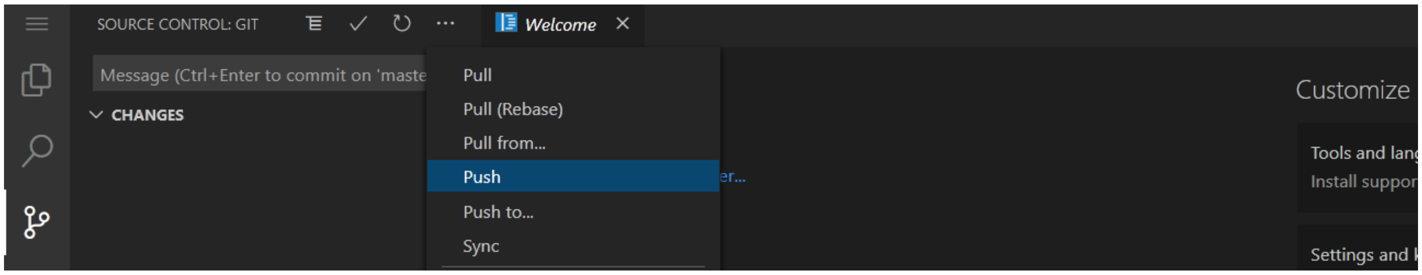


There are no staged changes to commit.

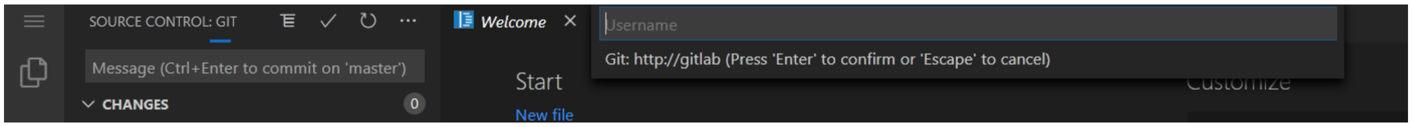
Would you like to automatically stage all your changes and commit them directly?

Choose the option “Always” to stage the commits always after the changes are made to the files.

**Step 11:** Click on the “More Actions” button (3 dots)



And select the Push button to push the changes to the remote repository.

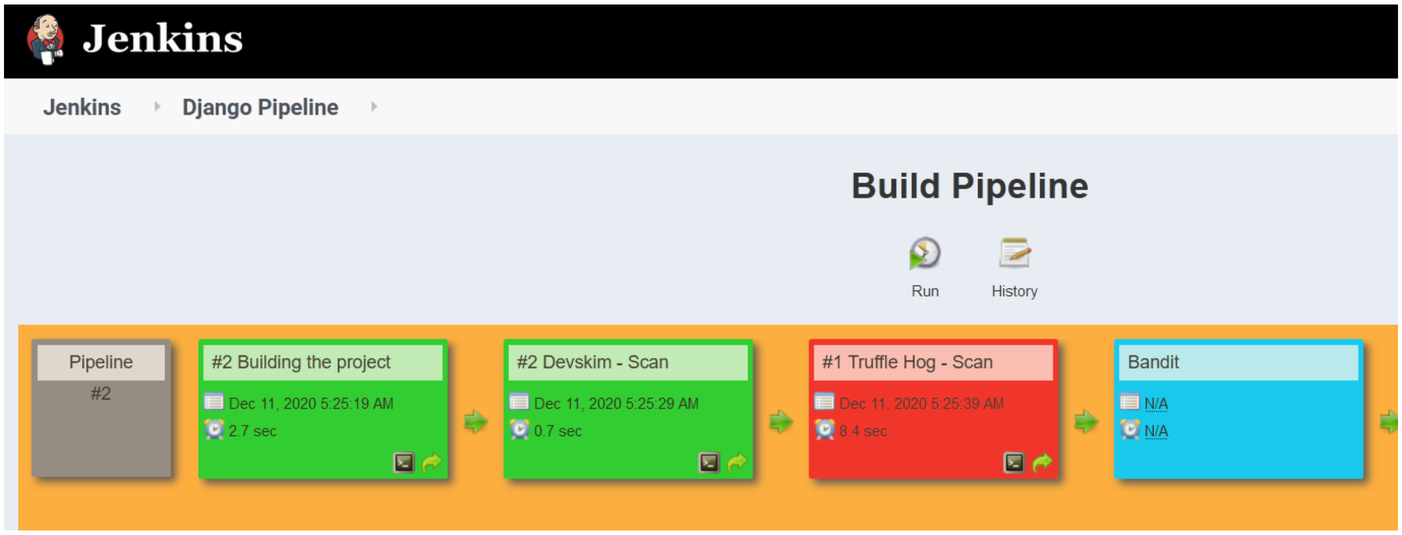


Git: http://gitlab (Press 'Enter' to confirm or 'Escape' to cancel)

Enter the git username and password when asked in the fields. The credentials are the same from gitlab.

**Note:** As soon as the code is pushed to the remote repository, the pipeline will start building automatically.

**Step 12:** Navigate back to the Pipeline view to check if Devskim stage passes or not.



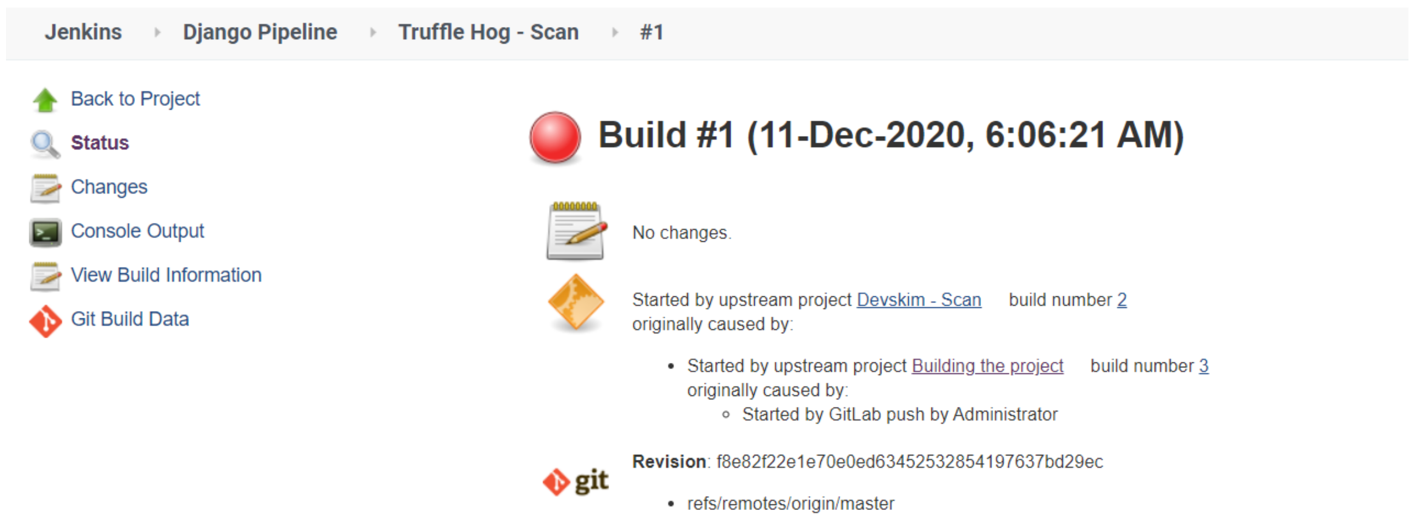
The Devskim stage has passed successfully. The logs can be checked from clicking on the "Devskim - Scan" and opening the console output.

```
+ /devskim.sh
Issues found: 0 in 0 files
Files analyzed: 31
Files skipped: 169
FLAG 1: 5bbf3d90fb303699b48e378715526b50
Triggering a new build of Truffle Hog - Scan
Finished: SUCCESS
```

**FLAG 1:** 5bbf3d90fb303699b48e378715526b50

**TruffleHog Issue**

**Step 1:** Click on the 'TruffleHog- Scan' to check the job build page.


Jenkins › Django Pipeline › Truffle Hog - Scan › #1

- Back to Project
- Status
- Changes
- Console Output
- View Build Information
- Git Build Data

**Build #1 (11-Dec-2020, 6:06:21 AM)**

No changes.

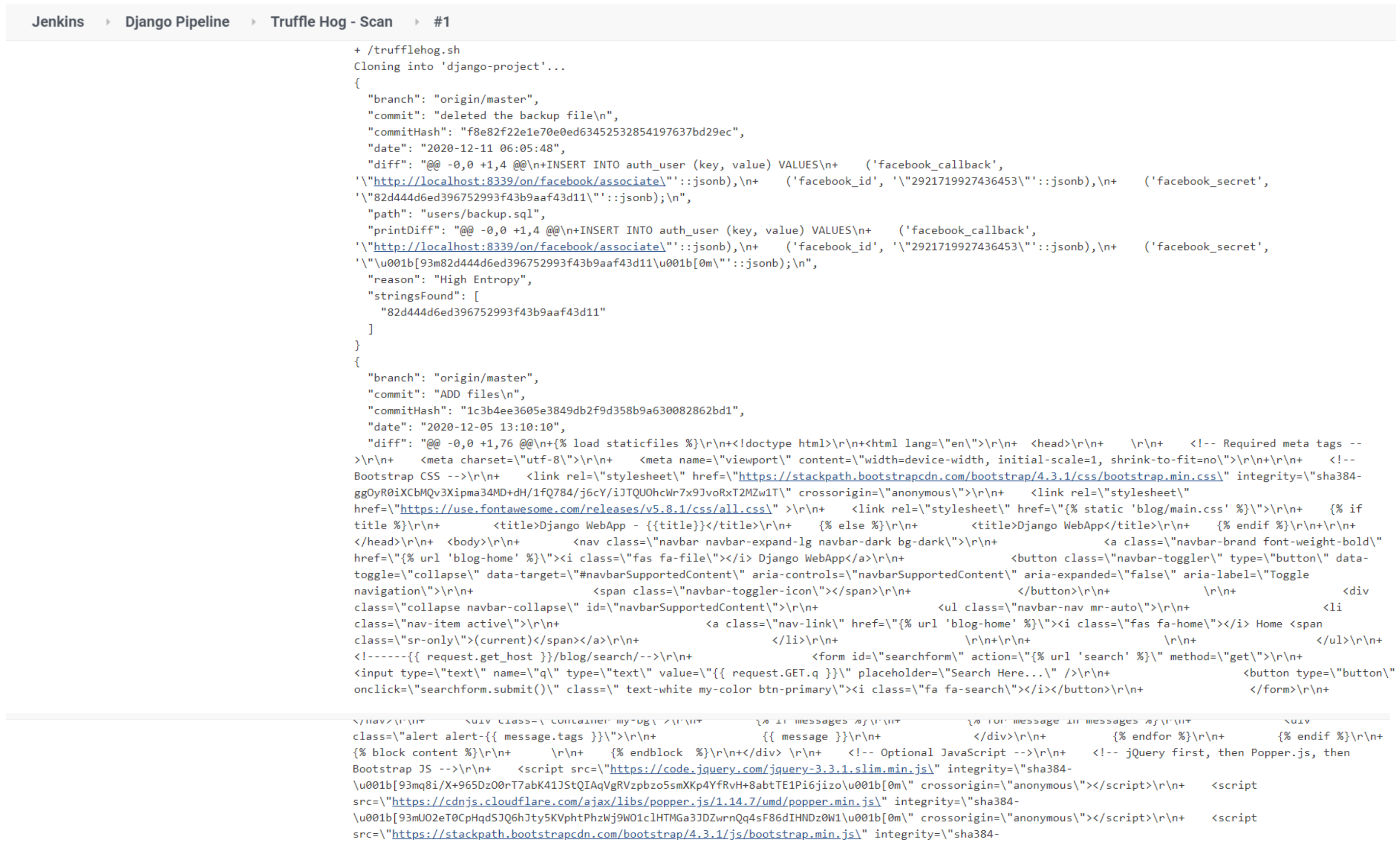
Started by upstream project Devskim - Scan build number 2
originally caused by:

- Started by upstream project Building the project build number 3
  originally caused by:
  - Started by GitLab push by Administrator

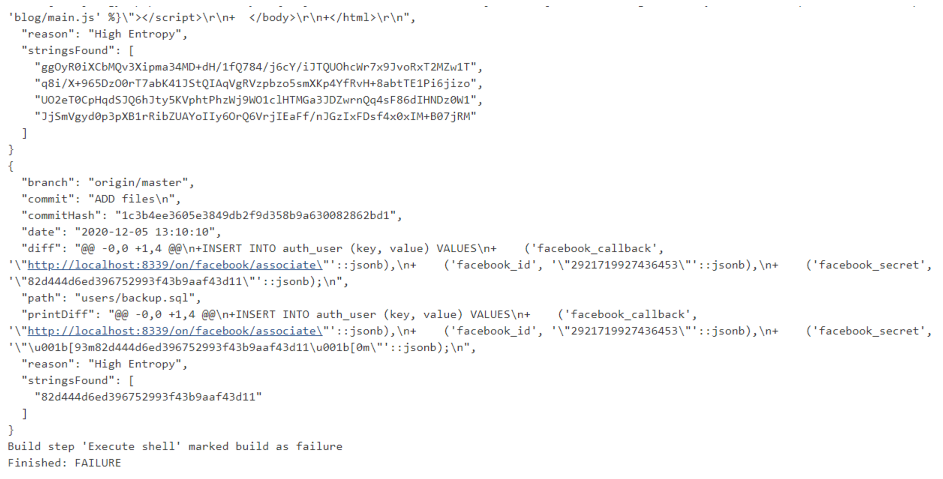
**Revision**: f8e82f22e1e70e0ed63452532854197637bd29ec

- refs/remotes/origin/master

**Step 2:** Click on the “Console Output” to check the issues found by TruffleHog tool.


Jenkins › Django Pipeline › Truffle Hog - Scan › #1

```
+ /trufflehog.sh
Cloning into 'django-project'...
{
  "branch": "origin/master",
  "commit": "deleted the backup file\n",
  "commitHash": "f8e82f22e1e70e0ed63452532854197637bd29ec",
  "date": "2020-12-11 06:05:48",
  "diff": "@@ -0,0 +1,4 @@\n+INSERT INTO auth_user (key, value) VALUES\n+    ('facebook_callback',
'\"http://localhost:8339/on/facebook/associate\"'::jsonb),\n+    ('facebook_id', '\"2921719927436453\"'::jsonb),\n+    ('facebook_secret',
'\"82d444d6ed396752993f43b9aaf43d11\"'::jsonb);\n",
  "path": "users/backup.sql",
  "printDiff": "@@ -0,0 +1,4 @@\n+INSERT INTO auth_user (key, value) VALUES\n+    ('facebook_callback',
'\"http://localhost:8339/on/facebook/associate\"'::jsonb),\n+    ('facebook_id', '\"2921719927436453\"'::jsonb),\n+    ('facebook_secret',
'\"\u001b[93m82d444d6ed396752993f43b9aaf43d11\u001b[0m\"'::jsonb);\n",
  "reason": "High Entropy",
  "stringsFound": [
    "82d444d6ed396752993f43b9aaf43d11"
  ]
}
{
  "branch": "origin/master",
  "commit": "ADD files\n",
  "commitHash": "1c3b4ee3605e3849db2f9d358b9a630082862bd1",
  "date": "2020-12-05 13:10:10",
  "diff": "@@ -0,0 +1,76 @@\n+{% load staticfiles %}\r\n+<!doctype html>\r\n+<html lang=\"en\">\r\n+  <head>\r\n+    \r\n+    <!-- Required meta tags --
>\r\n+    <meta charset=\"utf-8\">\r\n+    <meta name=\"viewport\" content=\"width=device-width, initial-scale=1, shrink-to-fit=no\">\r\n+\r\n+    <!--
Bootstrap CSS -->\r\n+    <link rel=\"stylesheet\" href=\"https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css\" integrity=\"sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T\" crossorigin=\"anonymous\">\r\n+    <link rel=\"stylesheet\"
href=\"https://use.fontawesome.com/releases/v5.8.1/css/all.css\" >\r\n+    <link rel=\"stylesheet\" href=\"{% static 'blog/main.css' %}\">\r\n+    {% if
title %}\r\n+        <title>Django WebApp - {{title}}</title>\r\n+    {% else %}\r\n+        <title>Django WebApp</title>\r\n+    {% endif %}\r\n+\r\n+
</head>\r\n+  <body>\r\n+        <nav class=\"navbar navbar-expand-lg navbar-dark bg-dark\">\r\n+                <a class=\"navbar-brand font-weight-bold\"
href=\"{% url 'blog-home' %}\"><i class=\"fas fa-file\"></i> Django WebApp</a>\r\n+                <button class=\"navbar-toggler\" type=\"button\" data-
toggle=\"collapse\" data-target=\"#navbarSupportedContent\" aria-controls=\"navbarSupportedContent\" aria-expanded=\"false\" aria-label=\"Toggle
navigation\">\r\n+                  <span class=\"navbar-toggler-icon\"></span>\r\n+                </button>\r\n+              \r\n+              <div
class=\"collapse navbar-collapse\" id=\"navbarSupportedContent\">\r\n+                  <ul class=\"navbar-nav mr-auto\">\r\n+                  <li
class=\"nav-item active\">\r\n+                      <a class=\"nav-link\" href=\"{% url 'blog-home' %}\"><i class=\"fas fa-home\"></i> Home <span
class=\"sr-only\">(current)</span></a>\r\n+                    </li>\r\n+                  \r\n+\r\n+                 \r\n+                  </ul>\r\n+
<!------{{ request.get_host }}/blog/search/-->\r\n+                  <form id=\"searchform\" action=\"{% url 'search' %}\" method=\"get\">\r\n+
<input type=\"text\" name=\"q\" type=\"text\" value=\"{{ request.GET.q }}\" placeholder=\"Search Here...\" />\r\n+                  <button type=\"button\"
onclick=\"searchform.submit()\" class=\" text-white my-color btn-primary\"><i class=\"fa fa-search\"></i></button>\r\n+              </form>\r\n+

</nav>\r\n+      <div class=\"container my-bg\" >\r\n+          {% if messages %}\r\n+              {% for message in messages %}\r\n+                  <div
class=\"alert alert-{{ message.tags }}\">\r\n+                      {{ message }}\r\n+                  </div>\r\n+              {% endfor %}\r\n+          {% endif %}\r\n+
{% block content %}\r\n+        \r\n+    {% endblock  %}\r\n+</div> \r\n+    <!-- Optional JavaScript -->\r\n+    <!-- jQuery first, then Popper.js, then
Bootstrap JS -->\r\n+    <script src=\"https://code.jquery.com/jquery-3.3.1.slim.min.js\" integrity=\"sha384-
\u001b[93mq8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo\u001b[0m\" crossorigin=\"anonymous\"></script>\r\n+    <script
src=\"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js\" integrity=\"sha384-
\u001b[93mUO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1\u001b[0m\" crossorigin=\"anonymous\"></script>\r\n+    <script
src=\"https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js\" integrity=\"sha384-
```



```
'blog/main.js' %}\"></script>\r\n+  </body>\r\n+</html>\r\n",
  "reason": "High Entropy",
  "stringsFound": [
    "ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T",
    "q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo",
    "UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1",
    "JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
  ]
}
{
  "branch": "origin/master",
  "commit": "ADD files\n",
  "commitHash": "1c3b4ee3605e3849db2f9d358b9a630082862bd1",
  "date": "2020-12-05 13:10:10",
  "diff": "@@ -0,0 +1,4 @@\n+INSERT INTO auth_user (key, value) VALUES\n+    ('facebook_callback',
'\"http://localhost:8339/on/facebook/associate\"'::jsonb),\n+    ('facebook_id', '\"2921719927436453\"'::jsonb),\n+    ('facebook_secret',
'\"82d444d6ed396752993f43b9aaf43d11\"'::jsonb);\n",
  "path": "users/backup.sql",
  "printDiff": "@@ -0,0 +1,4 @@\n+INSERT INTO auth_user (key, value) VALUES\n+    ('facebook_callback',
'\"http://localhost:8339/on/facebook/associate\"'::jsonb),\n+    ('facebook_id', '\"2921719927436453\"'::jsonb),\n+    ('facebook_secret',
'\"\u001b[93m82d444d6ed396752993f43b9aaf43d11\u001b[0m\"'::jsonb);\n",
  "reason": "High Entropy",
  "stringsFound": [
    "82d444d6ed396752993f43b9aaf43d11"
  ]
}
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```

The trufflehog identified several strings in the files and commit history which is flagged as sensitive information.

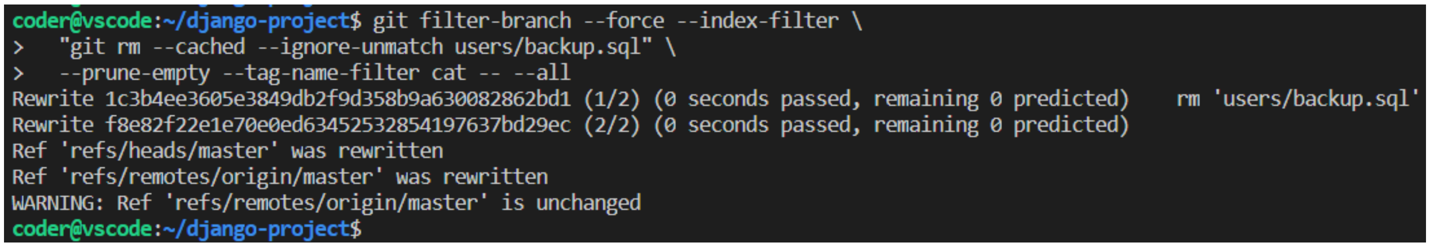
**Files:**

- users/backup.sql (commit history since current commit does not have the file present)
- blog/templates/blog/base.html

**Step 3:** Remove the backup SQL credentials file and base.html file from the commit history. Make a backup of the base.html which will be placed right back after removing the flagged strings.
(Execute the command on the VS Code server)

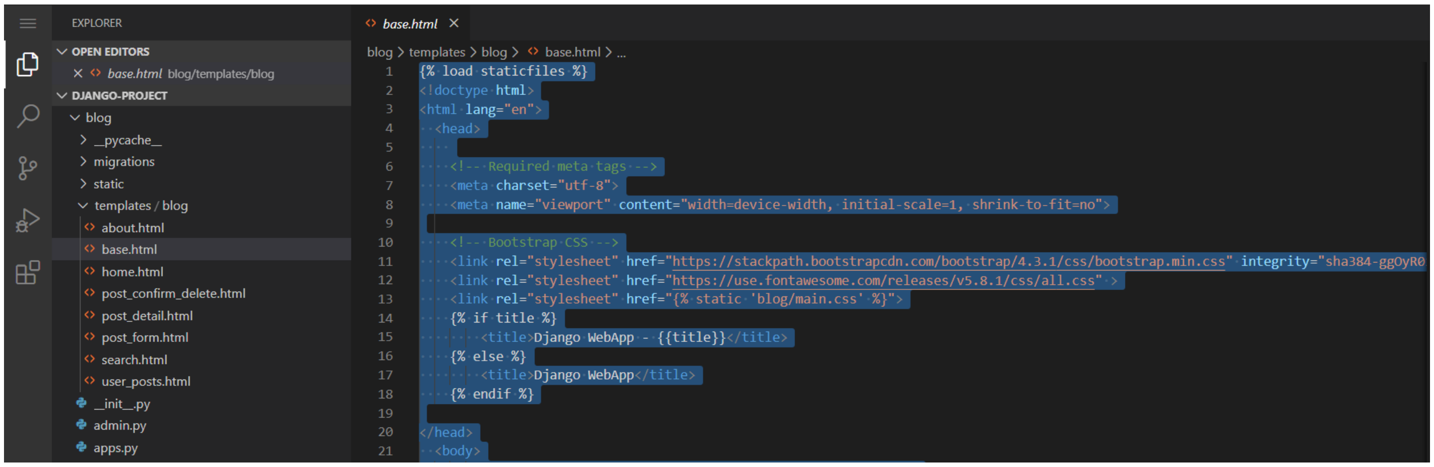
**Command:** git filter-branch --force --index-filter \
 "git rm --cached --ignore-unmatch users/backup.sql" \
 --prune-empty --tag-name-filter cat -- --all

The command will remove all the commits having the file credentials.txt and filter-branch will create commits in order to fix up the commits history.



```
coder@vscode:~/django-project$ git filter-branch --force --index-filter \
>   "git rm --cached --ignore-unmatch users/backup.sql" \
>   --prune-empty --tag-name-filter cat -- --all
Rewrite 1c3b4ee3605e3849db2f9d358b9a630082862bd1 (1/2) (0 seconds passed, remaining 0 predicted)    rm 'users/backup.sql'
Rewrite f8e82f22e1e70e0ed63452532854197637bd29ec (2/2) (0 seconds passed, remaining 0 predicted)
Ref 'refs/heads/master' was rewritten
Ref 'refs/remotes/origin/master' was rewritten
WARNING: Ref 'refs/remotes/origin/master' is unchanged
coder@vscode:~/django-project$
```

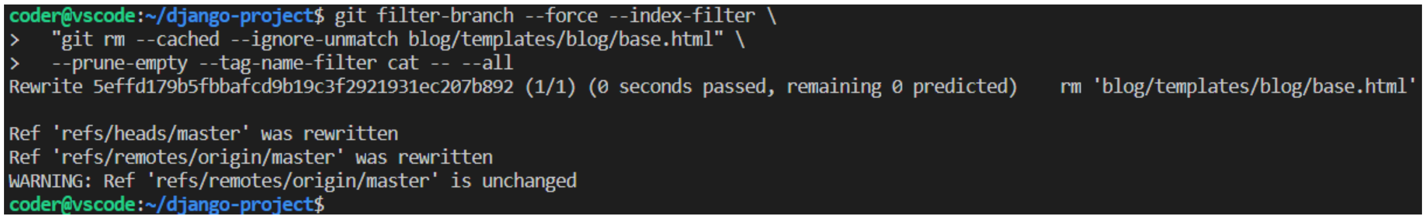
Copy the source code of base.html (blog/templates/base.html) and create a backup somewhere.



```
{% load staticfiles %}
<!doctype html>
<html lang="en">
  <head>

    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.1/css/all.css" >
    <link rel="stylesheet" href="{% static 'blog/main.css' %}">
    {% if title %}
        <title>Django WebApp - {{title}}</title>
    {% else %}
        <title>Django WebApp</title>
    {% endif %}

</head>
  <body>
```

**Step 4:** Enter the command provided below to delete the base.html from current and previous commits.

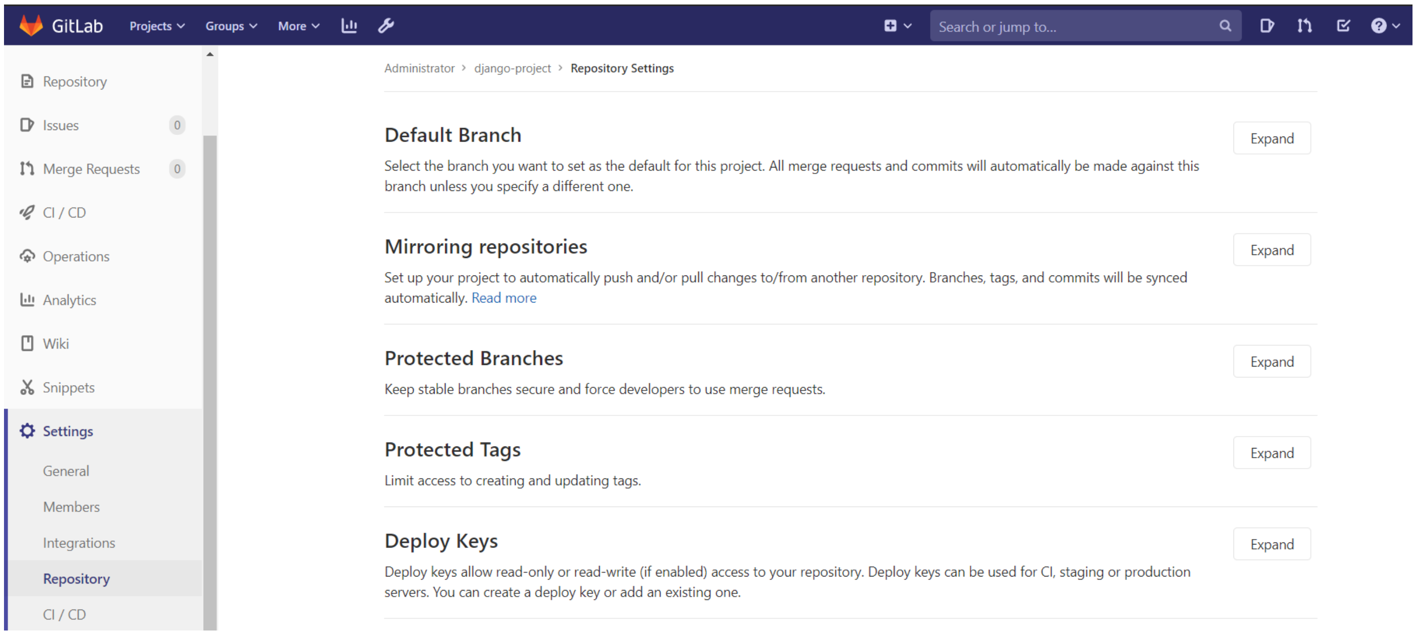
**Command:** git filter-branch --force --index-filter \
  "git rm --cached --ignore-unmatch blog/templates/blog/base.html" \
  --prune-empty --tag-name-filter cat -- --all



```
coder@vscode:~/django-project$ git filter-branch --force --index-filter \
>   "git rm --cached --ignore-unmatch blog/templates/blog/base.html" \
>   --prune-empty --tag-name-filter cat -- --all
Rewrite 5effd179b5fbbafcd9b19c3f2921931ec207b892 (1/1) (0 seconds passed, remaining 0 predicted)    rm 'blog/templates/blog/base.html'

Ref 'refs/heads/master' was rewritten
Ref 'refs/remotes/origin/master' was rewritten
WARNING: Ref 'refs/remotes/origin/master' is unchanged
coder@vscode:~/django-project$
```

The base.html has been removed from every commit happened in the repository.

**Step 5:** Navigate to the gitlab website and open the repository settings (Repository Settings)



Administrator › django-project › Repository Settings

Default Branch

Select the branch you want to set as the default for this project. All merge requests and commits will automatically be made against this branch unless you specify a different one.

Mirroring repositories

Set up your project to automatically push and/or pull changes to/from another repository. Branches, tags, and commits will be synced automatically. Read more

Protected Branches

Keep stable branches secure and force developers to use merge requests.

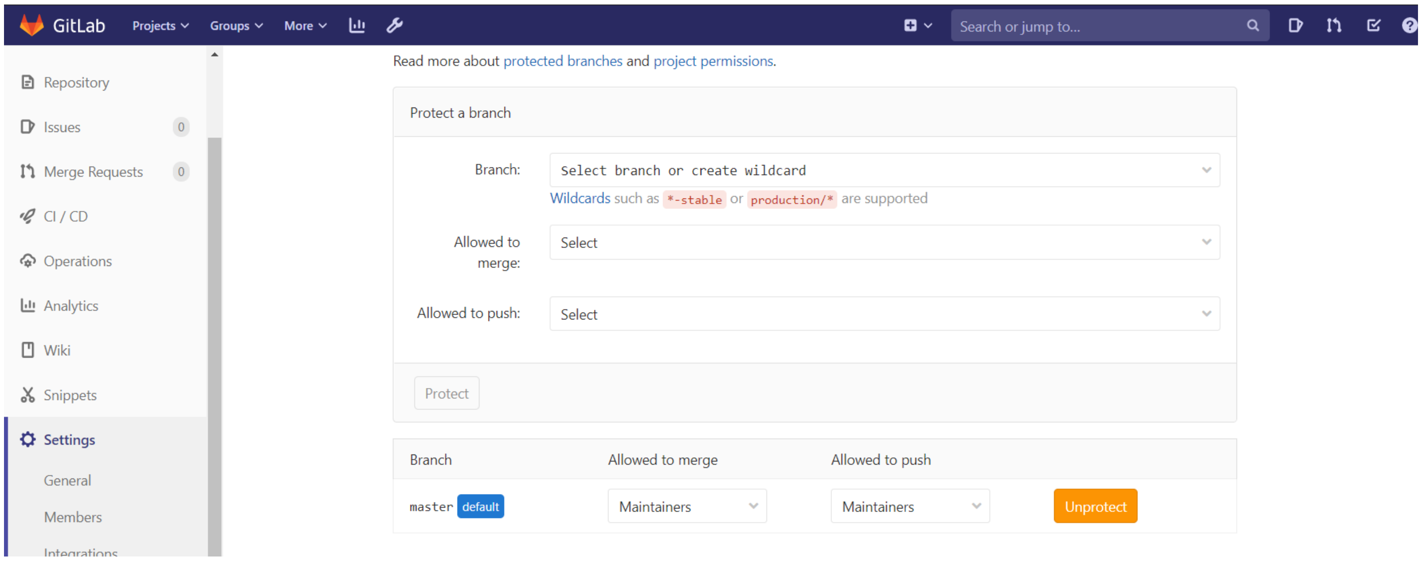
Protected Tags

Limit access to creating and updating tags.

Deploy Keys

Deploy keys allow read-only or read-write (if enabled) access to your repository. Deploy keys can be used for CI, staging or production servers. You can create a deploy key or add an existing one.

Expand the Protected Branches section.



Read more about protected branches and project permissions.

Protect a branch

Branch: Select branch or create wildcard

Wildcards such as `*-stable` or `production/*` are supported

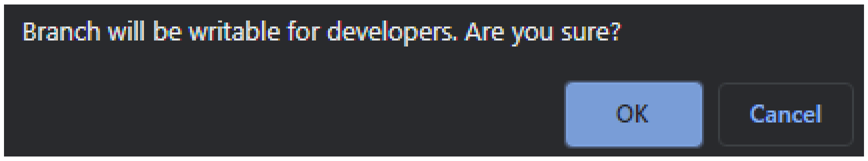
Allowed to merge: Select

Allowed to push: Select

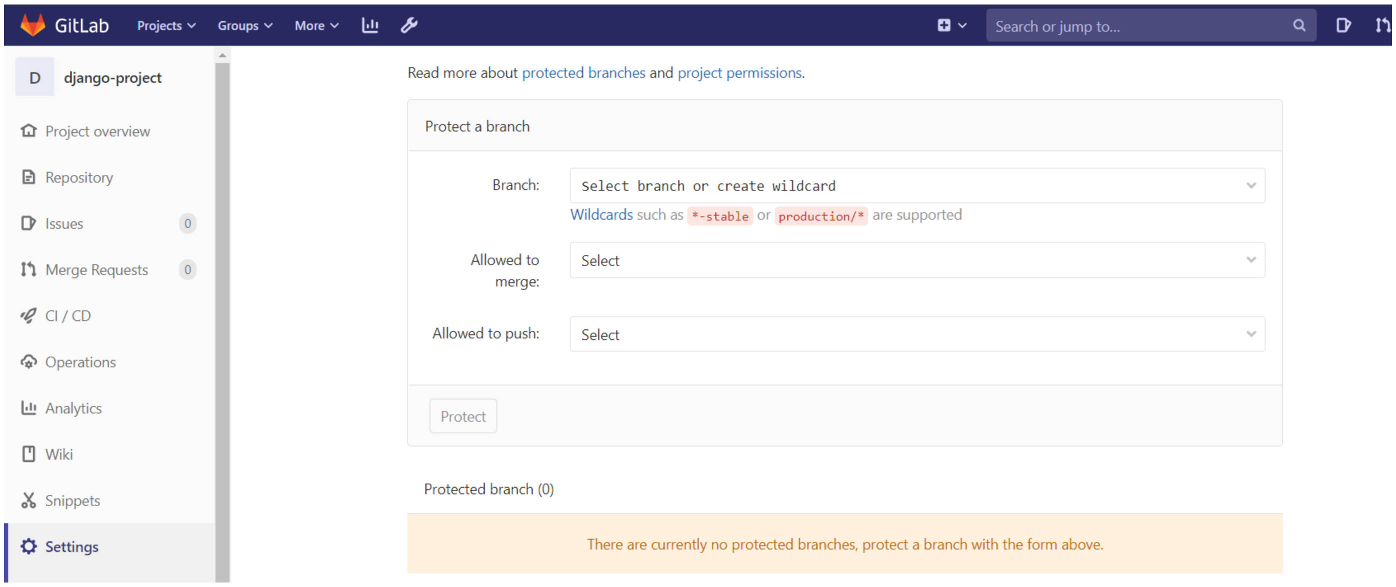
| Branch | Allowed to merge | Allowed to push | |
|---|---|---|---|
| master default | Maintainers | Maintainers | Unprotect |

In order to push the changes which include altering the base commits (previous), We need to pass a force push to the remote repository and to do that the main branch needs to be set to "Unprotect" mode.

**Step 6:** Click on the Unprotect button.


Branch will be writable for developers. Are you sure?

Click on ‘OK’ when prompted.



Read more about protected branches and project permissions.

Protect a branch

Branch: Select branch or create wildcard

Wildcards such as `*-stable` or `production/*` are supported

Allowed to merge: Select

Allowed to push: Select

Protect

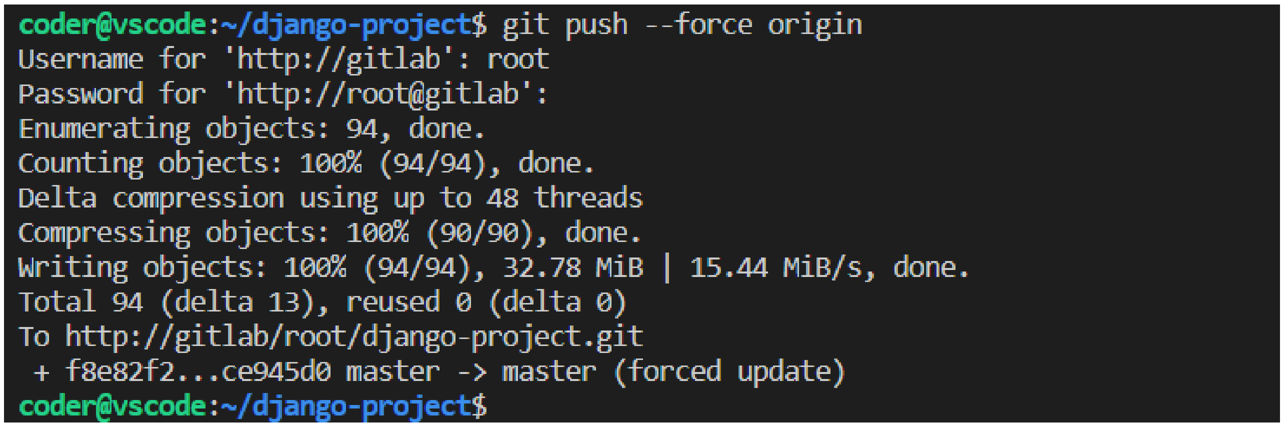
Protected branch (0)

There are currently no protected branches, protect a branch with the form above.

**Step 7:** Pass a force push to the remote repository. (Execute the command on VS Code server)

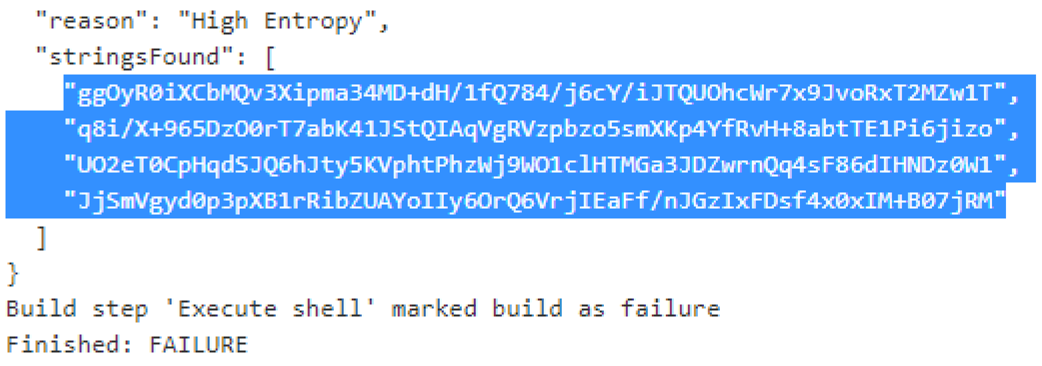
**Command:** git push --force origin

Enter GitLab credentials when prompted.

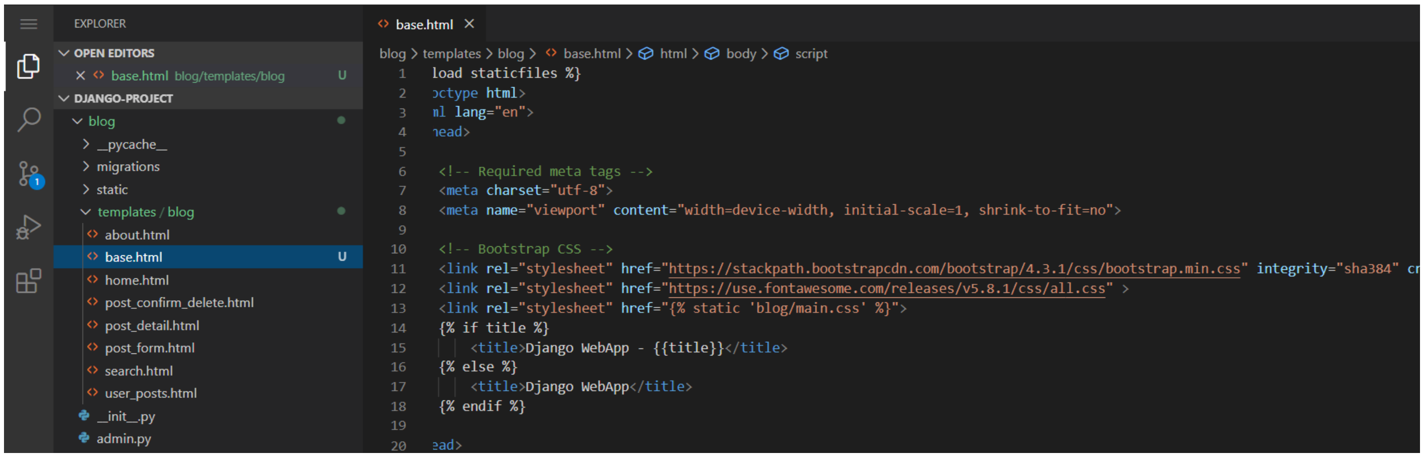


```
coder@vscode:~/django-project$ git push --force origin
Username for 'http://gitlab': root
Password for 'http://root@gitlab':
Enumerating objects: 94, done.
Counting objects: 100% (94/94), done.
Delta compression using up to 48 threads
Compressing objects: 100% (90/90), done.
Writing objects: 100% (94/94), 32.78 MiB | 15.44 MiB/s, done.
Total 94 (delta 13), reused 0 (delta 0)
To http://gitlab/root/django-project.git
 + f8e82f2...ce945d0 master -> master (forced update)
coder@vscode:~/django-project$
```

**Step 8:** Create the base.html file and place the content of the base.html (backed up earlier) after removing the flagged strings. (Flagged strings are mentioned at the end of the Truffle Hog's Console Output)

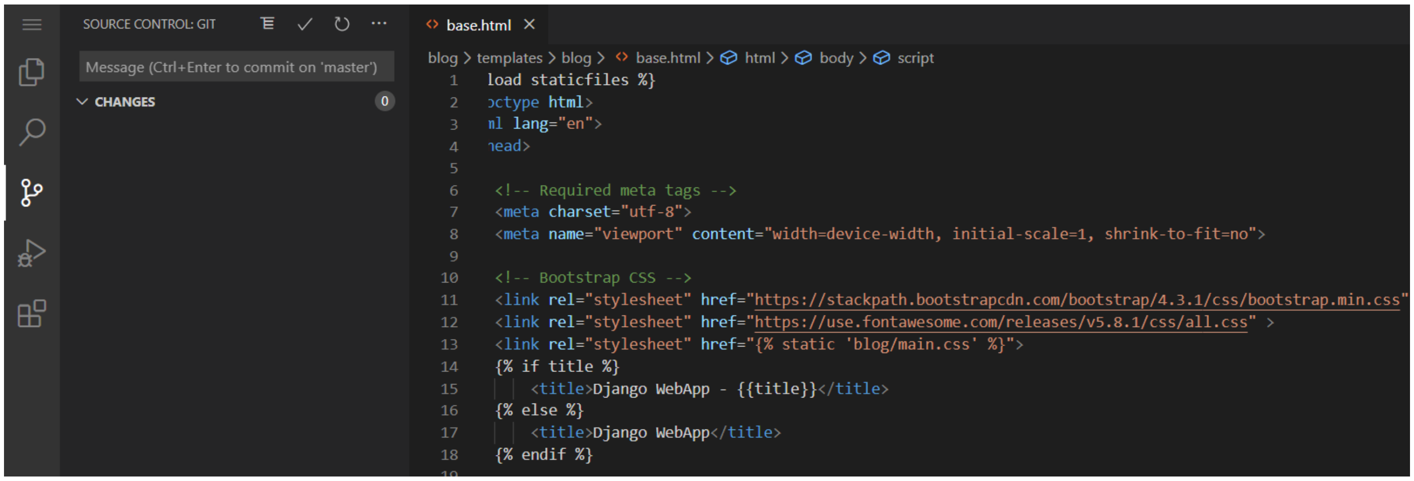


```
    "reason": "High Entropy",
    "stringsFound": [
      "ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T",
      "q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo",
      "UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1",
      "JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
    ]
  }
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```



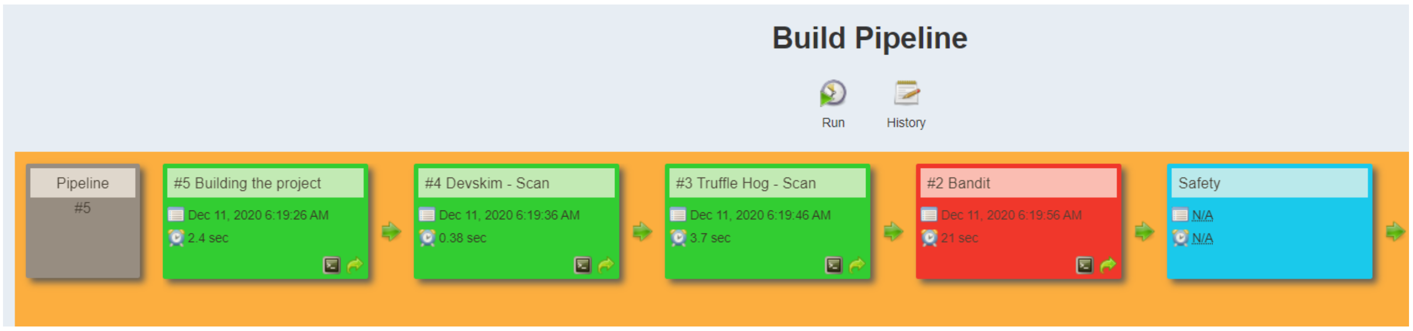
```
load staticfiles %}
ctype html>
l lang="en">
ead>

 <!-- Required meta tags -->
 <meta charset="utf-8">
 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

 <!-- Bootstrap CSS -->
 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384" cr
 <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.1/css/all.css" >
 <link rel="stylesheet" href="{% static 'blog/main.css' %}">
 {% if title %}
     <title>Django WebApp - {{title}}</title>
 {% else %}
     <title>Django WebApp</title>
 {% endif %}

ead>
```

**Step 9:** Commit the changes and push them to the remote repository.
(Same as Step 10 Onwards of the DevSkim section)


SOURCE CONTROL: GIT
Message (Ctrl+Enter to commit on 'master')
CHANGES 0
base.html
blog > templates > blog > base.html > html > body > script

```
load staticfiles %}
octype html>
ml lang="en">
head>

  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.1/css/all.css" >
  <link rel="stylesheet" href="{% static 'blog/main.css' %}">
  {% if title %}
      <title>Django WebApp - {{title}}</title>
  {% else %}
      <title>Django WebApp</title>
  {% endif %}
```

**Step 10:** Check the pipeline for the changes.



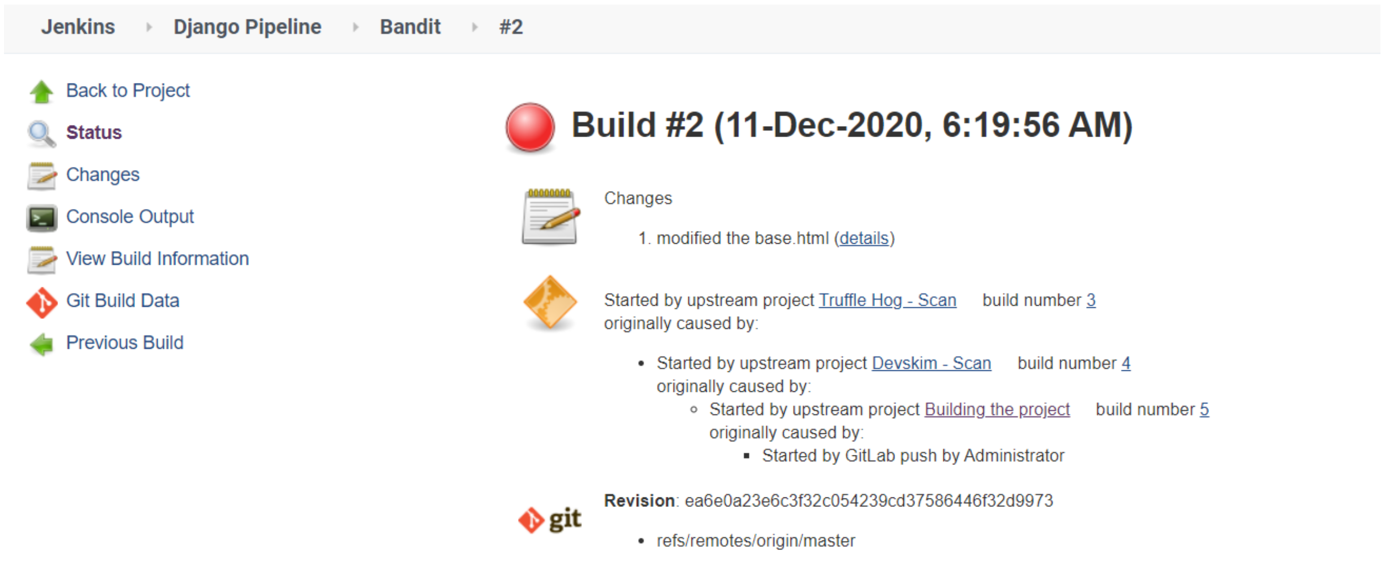
The TruffleHog stage has passed successfully. The logs can be checked from clicking on the "Truffle Hog - Scan" and opening the console output.

```
+ /trufflehog.sh
Cloning into 'django-project'...
FLAG 2: be88d953d8650af500f77e017d985b63
Triggering a new build of Bandit
Finished: SUCCESS
```

**FlAG 2:** be88d953d8650af500f77e017d985b63

## Bandit Issue

**Step 1:** Click on the ‘Bandit’ to check the job build page.



Build #2 (11-Dec-2020, 6:19:56 AM)

Changes

1. modified the base.html (details)

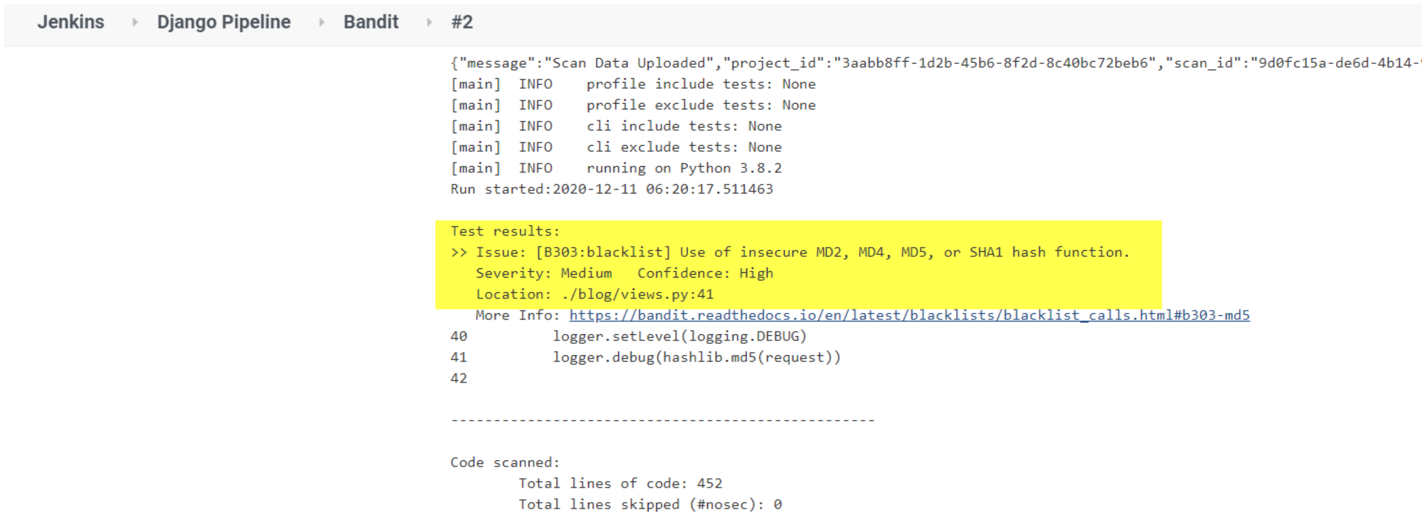
Started by upstream project Truffle Hog - Scan build number 3
originally caused by:

- Started by upstream project Devskim - Scan build number 4
  originally caused by:
  - Started by upstream project Building the project build number 5
    originally caused by:
    - Started by GitLab push by Administrator

**Revision**: ea6e0a23e6c3f32c054239cd37586446f32d9973

- refs/remotes/origin/master

**Step 2:** Click on the “Console Output” to check the issues found by the Bandit tool.



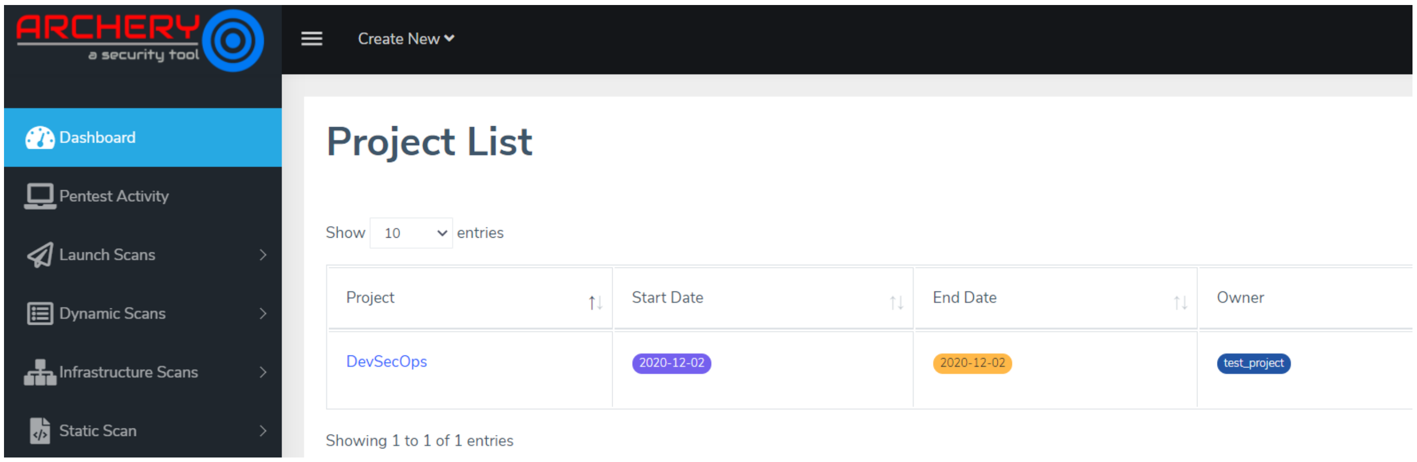
```
{"message":"Scan Data Uploaded","project_id":"3aabb8ff-1d2b-45b6-8f2d-8c40bc72beb6","scan_id":"9d0fc15a-de6d-4b14-
[main]  INFO    profile include tests: None
[main]  INFO    profile exclude tests: None
[main]  INFO    cli include tests: None
[main]  INFO    cli exclude tests: None
[main]  INFO    running on Python 3.8.2
Run started:2020-12-11 06:20:17.511463

Test results:
>> Issue: [B303:blacklist] Use of insecure MD2, MD4, MD5, or SHA1 hash function.
   Severity: Medium   Confidence: High
   Location: ./blog/views.py:41
   More Info: https://bandit.readthedocs.io/en/latest/blacklists/blacklist_calls.html#b303-md5
40          logger.setLevel(logging.DEBUG)
41          logger.debug(hashlib.md5(request))
42

--------------------------------------------------

Code scanned:
        Total lines of code: 452
        Total lines skipped (#nosec): 0
```

The results of the scan can be checked on archery sec portal.

**Step 3:** Open the ArcherySec page and log in using the credentials provided.

**Credentials:**

- **Username:** admin
- **Password:** admin



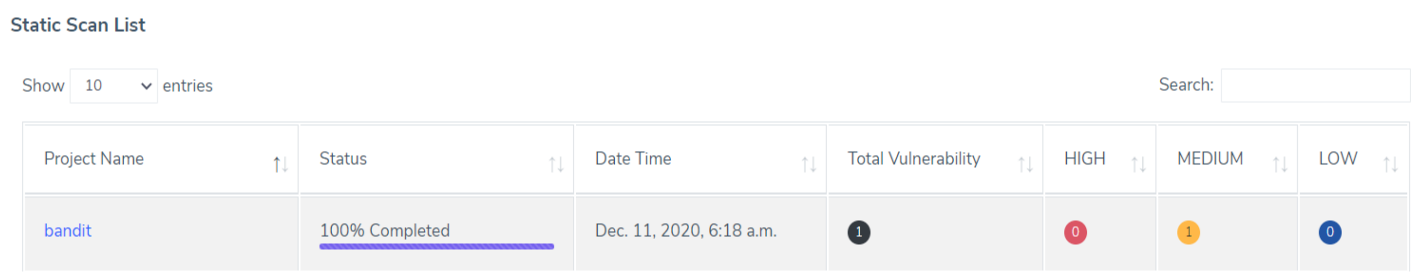
## Project List

Show 10 entries

| Project | Start Date | End Date | Owner |
|---|---|---|---|
| DevSecOps | 2020-12-02 | 2020-12-02 | test_project |

Showing 1 to 1 of 1 entries

Click on the project name “DevSecOps” and see results from the bandit tool.



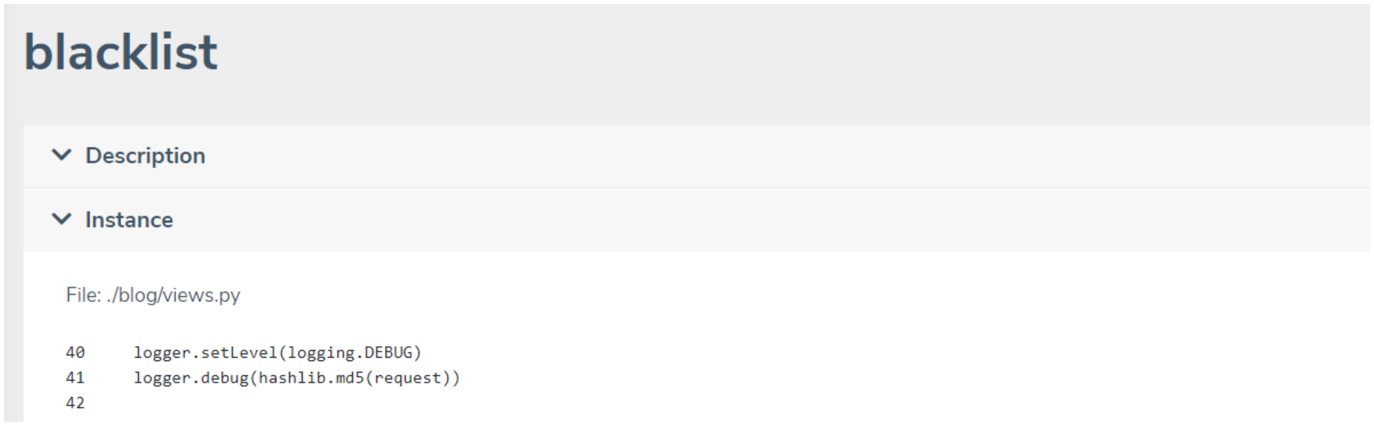
Static Scan List

Show 10 entries

Search:

| Project Name | Status | Date Time | Total Vulnerability | HIGH | MEDIUM | LOW |
|---|---|---|---|---|---|---|
| bandit | 100% Completed | Dec. 11, 2020, 6:18 a.m. | 1 | 0 | 1 | 0 |

Click on the Bandit and check the issue found.



blacklist

Description

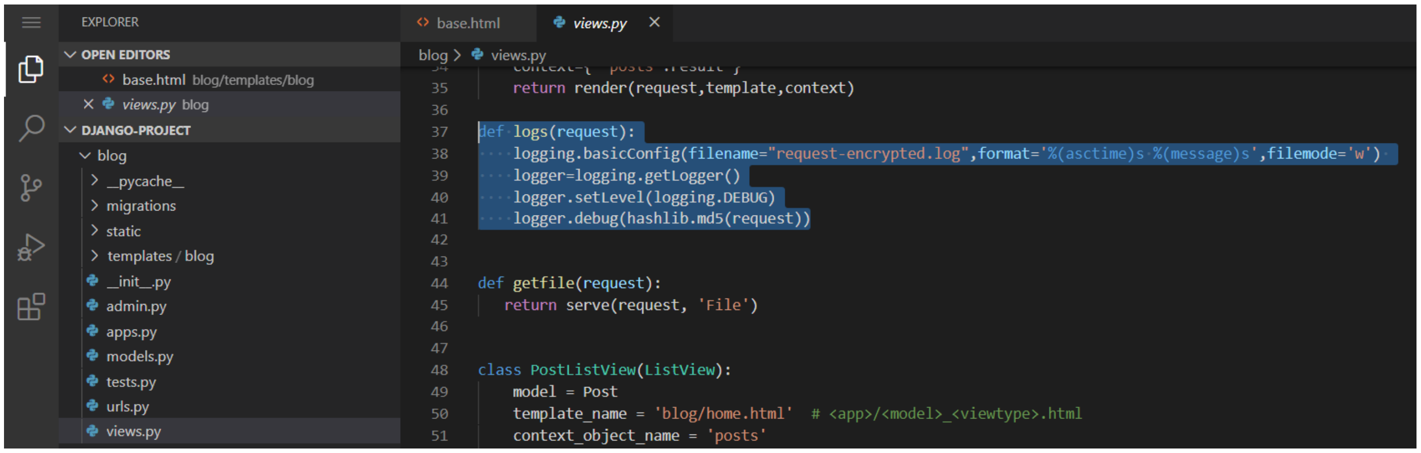
Instance

File: ./blog/views.py

```
40      logger.setLevel(logging.DEBUG)
41      logger.debug(hashlib.md5(request))
42
```

The file blog/views.py has a logging function which is using MD5 algorithm to hash the request. The MD5 algorithm is insecure which could lead to the compromise of sensitive data by brute-forcing the values.

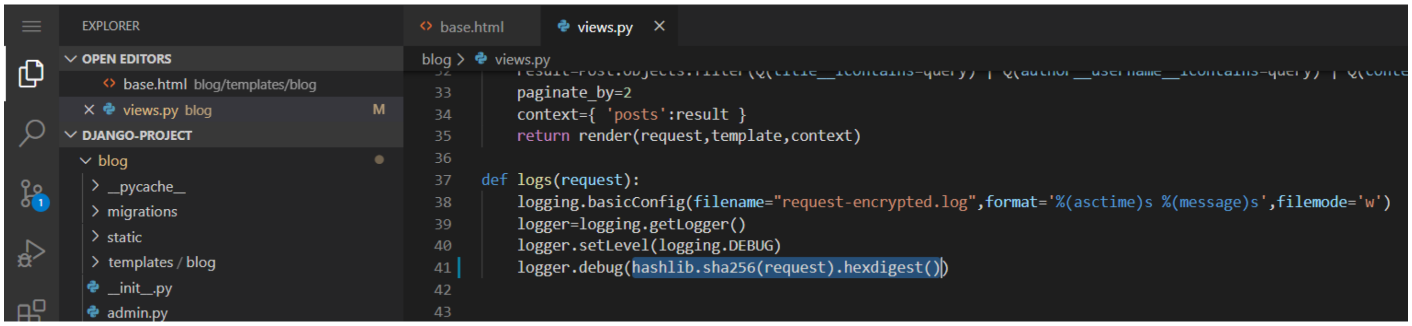
**Step 4:** Open the file in vs code server.



```
35      return render(request,template,context)
36
37  def logs(request):
38      logging.basicConfig(filename="request-encrypted.log",format='%(asctime)s %(message)s',filemode='w')
39      logger=logging.getLogger()
40      logger.setLevel(logging.DEBUG)
41      logger.debug(hashlib.md5(request))
42
43
44  def getfile(request):
45      return serve(request, 'File')
46
47
48  class PostListView(ListView):
49      model = Post
50      template_name = 'blog/home.html'  # <app>/<model>_<viewtype>.html
51      context_object_name = 'posts'
```

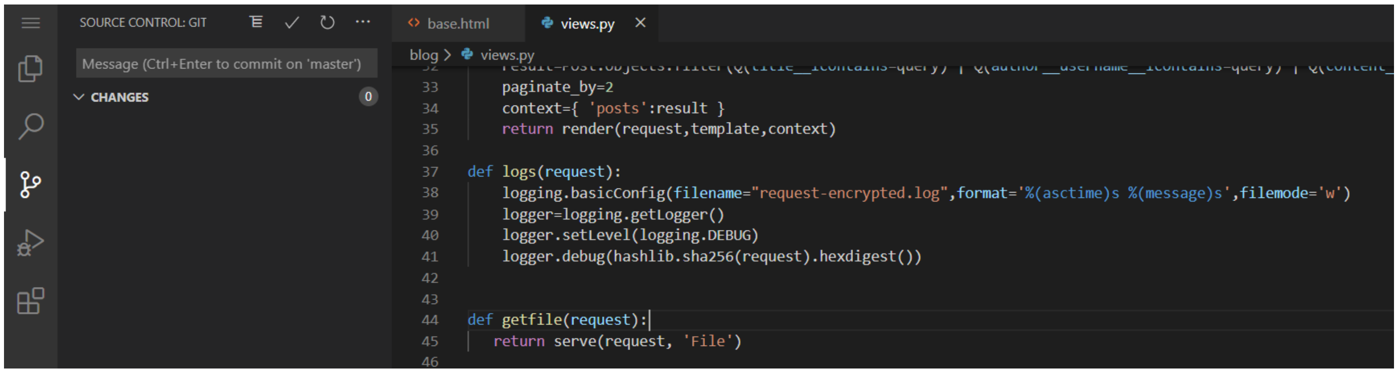
Modify the algorithm to secure the logging implementation.

Line 41
From: logger.debug(hashlib.md5(request))
To: logger.debug(hashlib.sha256(request).hexdigest())



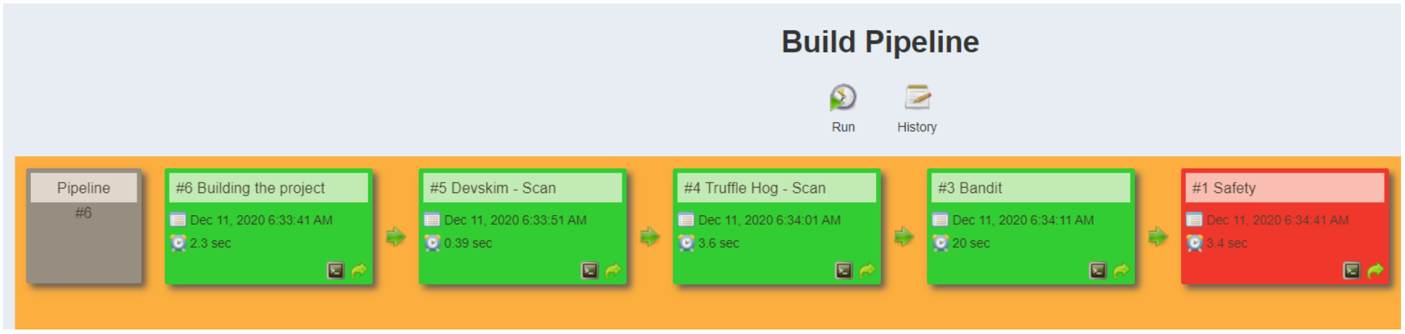
```
    paginate_by=2
    context={ 'posts':result }
    return render(request,template,context)

def logs(request):
    logging.basicConfig(filename="request-encrypted.log",format='%(asctime)s %(message)s',filemode='w')
    logger=logging.getLogger()
    logger.setLevel(logging.DEBUG)
    logger.debug(hashlib.sha256(request).hexdigest())
```

Commit and push the changes to the remote repository.
(Same as Step 10 Onwards of the DevSkim section)

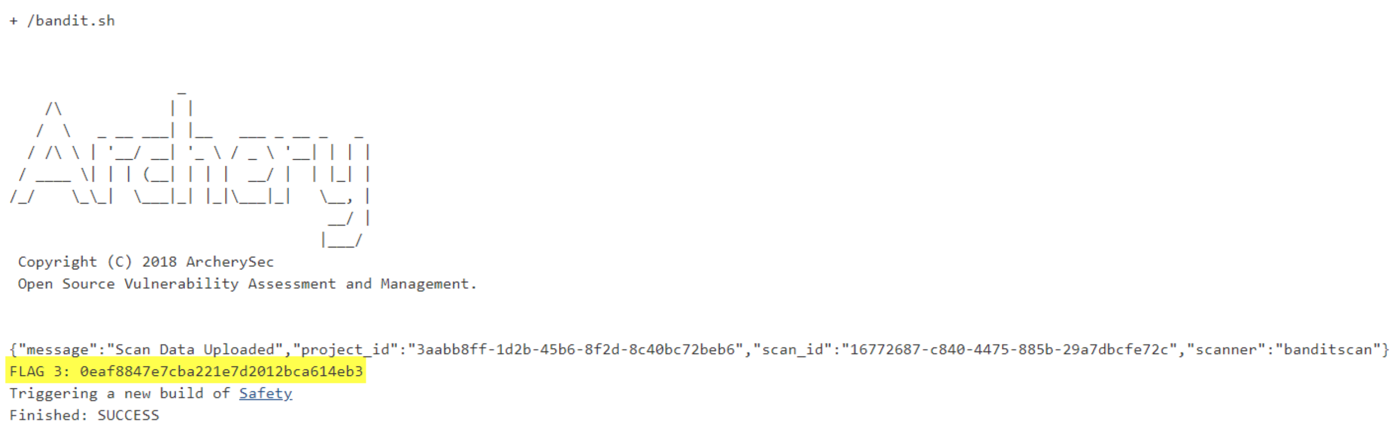


```
    paginate_by=2
    context={ 'posts':result }
    return render(request,template,context)

def logs(request):
    logging.basicConfig(filename="request-encrypted.log",format='%(asctime)s %(message)s',filemode='w')
    logger=logging.getLogger()
    logger.setLevel(logging.DEBUG)
    logger.debug(hashlib.sha256(request).hexdigest())


def getfile(request):
    return serve(request, 'File')
```

**Step 5:** Check the pipeline to see any changes.



The Bandit stage has passed successfully. The logs can be checked from clicking on the "Bandit" and opening the console output.

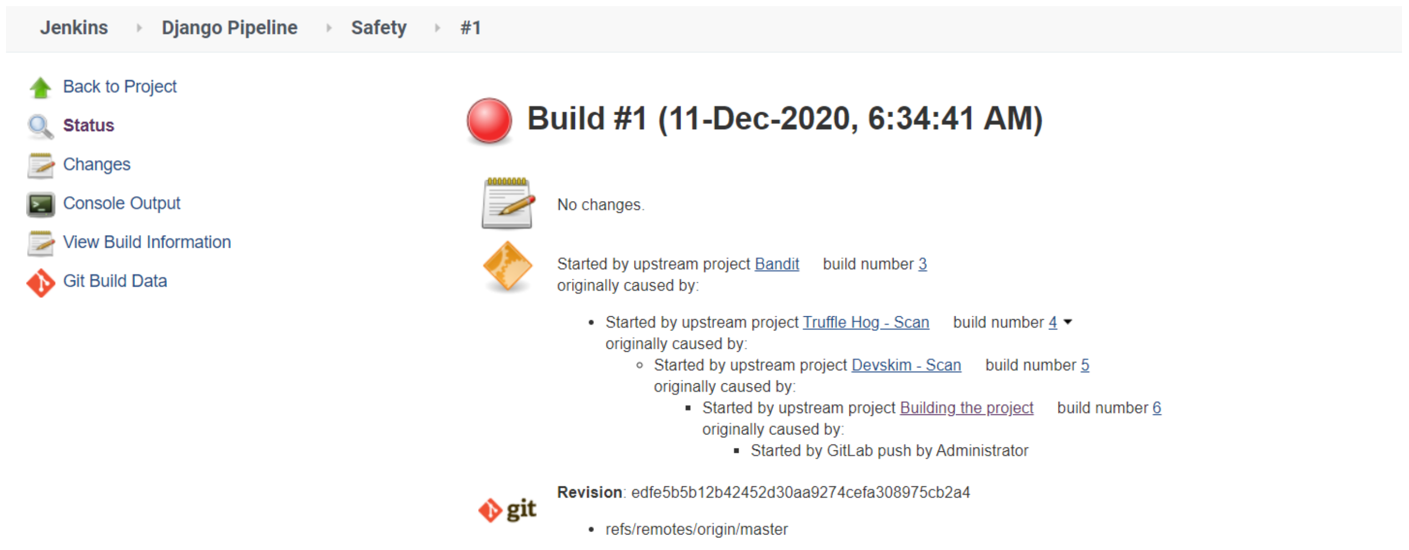


```
+ /bandit.sh


                   _
    /\            | |
   /  \   _ __ ___| |__   ___ _ __ _   _
  / /\ \ | '__/ __| '_ \ / _ \ '__| | | |
 / ____ \| | | (__| | | |  __/ |  | |_| |
/_/    \_\_|  \___|_| |_|\___|_|   \__, |
                                    __/ |
                                   |___/
 Copyright (C) 2018 ArcherySec
 Open Source Vulnerability Assessment and Management.


{"message":"Scan Data Uploaded","project_id":"3aabb8ff-1d2b-45b6-8f2d-8c40bc72beb6","scan_id":"16772687-c840-4475-885b-29a7dbcfe72c","scanner":"banditscan"}
FLAG 3: 0eaf8847e7cba221e7d2012bca614eb3
Triggering a new build of Safety
Finished: SUCCESS
```

**FLAG 3:** 0eaf8847e7cba221e7d2012bca614eb3

**Safety Issue**

**Step 1:** Click on the ‘Safety’ to check the job build page.



Build #1 (11-Dec-2020, 6:34:41 AM)

No changes.

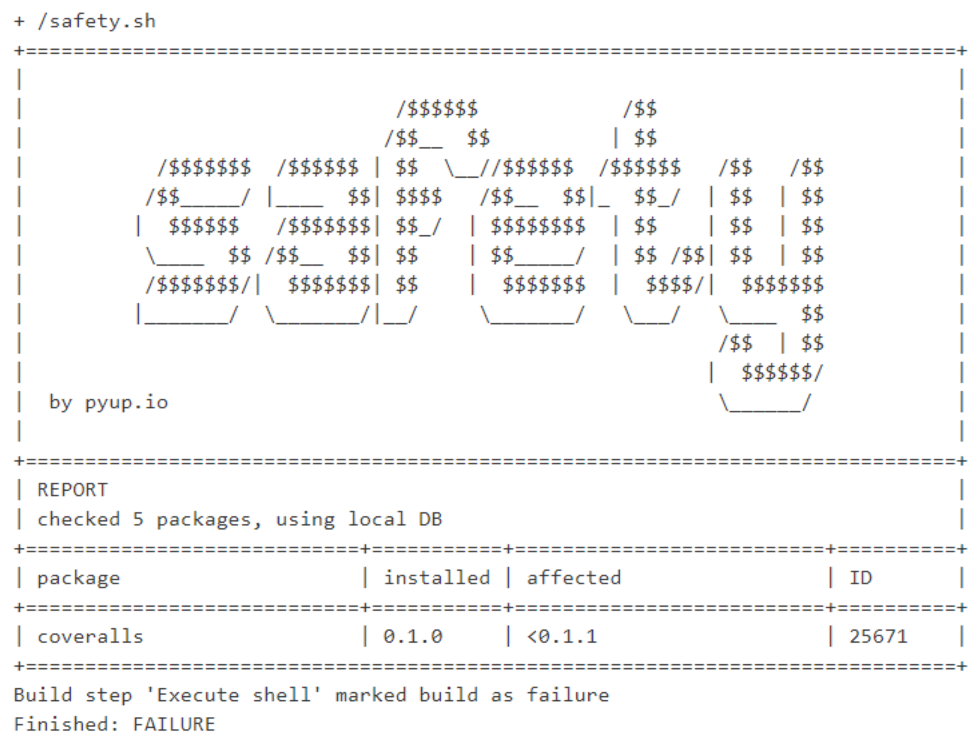
Started by upstream project Bandit build number 3
originally caused by:

- Started by upstream project Truffle Hog - Scan build number 4
  originally caused by:
  - Started by upstream project Devskim - Scan build number 5
    originally caused by:
    - Started by upstream project Building the project build number 6
      originally caused by:
      - Started by GitLab push by Administrator

**Revision**: edfe5b5b12b42452d30aa9274cefa308975cb2a4

- refs/remotes/origin/master

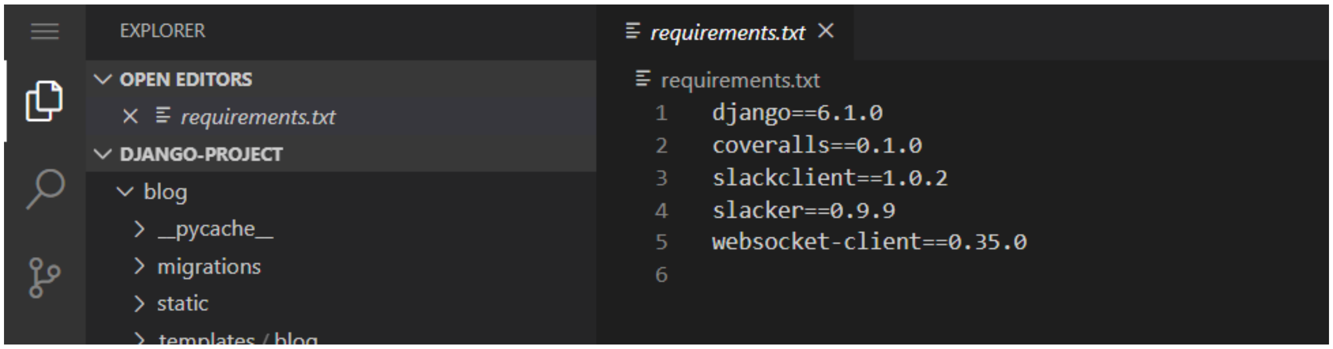
**Step 2:** Click on the “Console Output” to check the issues found by Safety tool.



```
+ /safety.sh
+==============================================================================+
|                                                                              |
|                               /$$$$$$            /$$                         |
|                              /$$__  $$          | $$                         |
|           /$$$$$$$  /$$$$$$ | $$  \__//$$$$$$  /$$$$$$   /$$   /$$           |
|          /$$_____/ |____  $$| $$$$   /$$__  $$|_  $$_/  | $$  | $$           |
|         |  $$$$$$   /$$$$$$$| $$_/  | $$$$$$$$  | $$    | $$  | $$           |
|          \____  $$ /$$__  $$| $$    | $$_____/  | $$ /$$| $$  | $$           |
|          /$$$$$$$/|  $$$$$$$| $$    |  $$$$$$$  |  $$$$/|  $$$$$$$           |
|         |_______/  \_______/|__/     \_______/   \___/   \____  $$           |
|                                                          /$$  | $$           |
|                                                         |  $$$$$$/           |
|  by pyup.io                                             \______/            |
|                                                                              |
+==============================================================================+
| REPORT                                                                       |
| checked 5 packages, using local DB                                           |
+============================+===========+==========================+==========+
| package                    | installed | affected                 | ID       |
+============================+===========+==========================+==========+
| coveralls                  | 0.1.0     | <0.1.1                   | 25671    |
+==============================================================================+
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```

There is only one issue found with the version installed of coveralls pip module.

**Step 3:** Open the requirements.txt file for the django-project directory. (On VS Code server)

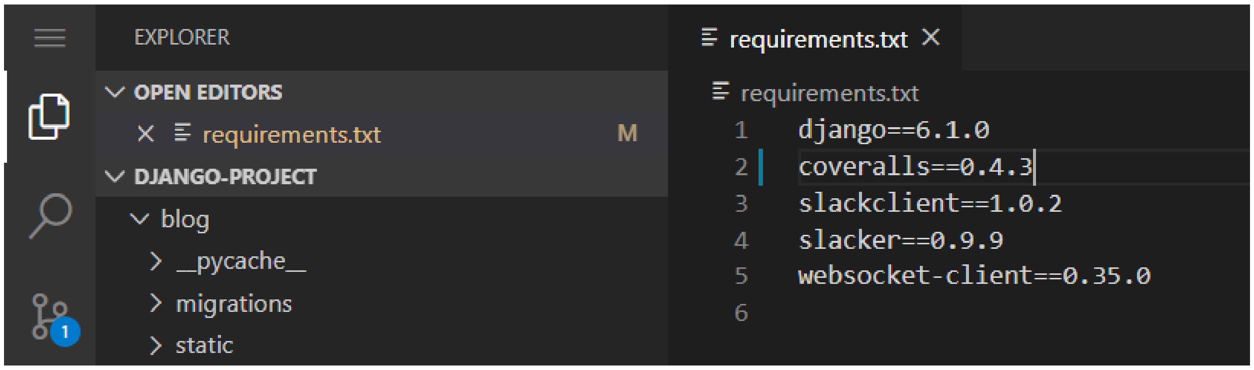


```
requirements.txt
django==6.1.0
coveralls==0.1.0
slackclient==1.0.2
slacker==0.9.9
websocket-client==0.35.0
```

In order to fix the issue, Modify the version numbers from the requirements.txt to the latest in order to resolve the issue.

From: coveralls==0.1.0

To: coveralls==0.4.3



EXPLORER

OPEN EDITORS

requirements.txt M

DJANGO-PROJECT

blog

__pycache__

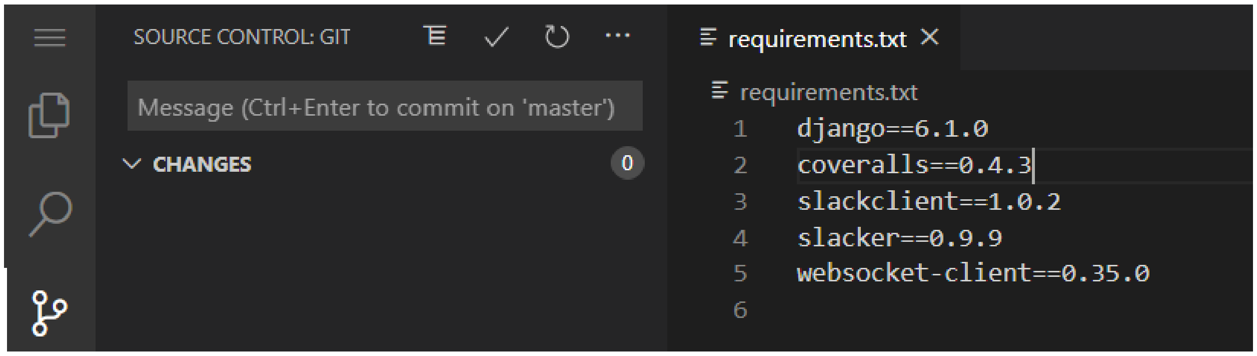
migrations

static

requirements.txt

```
django==6.1.0
coveralls==0.4.3
slackclient==1.0.2
slacker==0.9.9
websocket-client==0.35.0

```

Commit the changes and push the files to the remote repository.
(Same as Step 10 Onwards of the DevSkim section)



SOURCE CONTROL: GIT

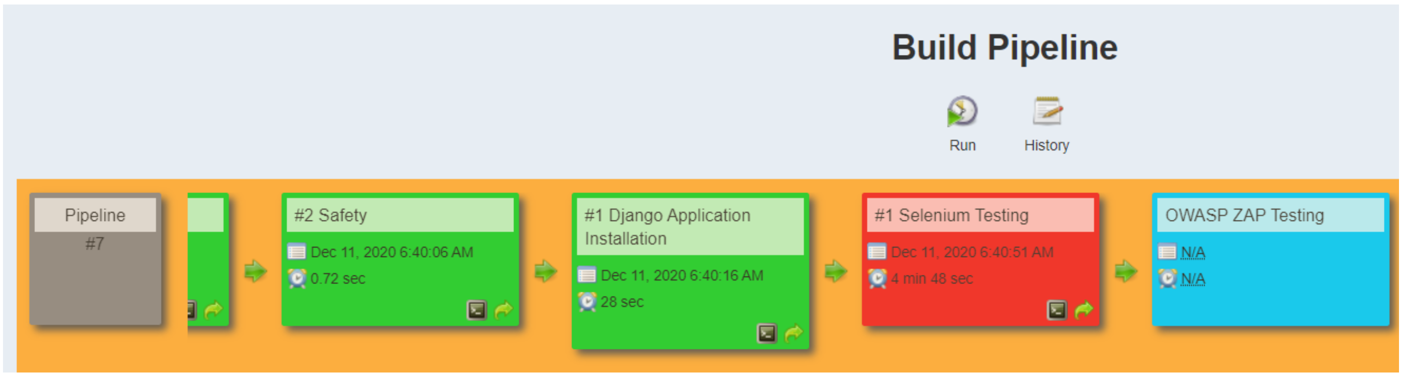
Message (Ctrl+Enter to commit on 'master')

CHANGES 0

requirements.txt

```
django==6.1.0
coveralls==0.4.3
slackclient==1.0.2
slacker==0.9.9
websocket-client==0.35.0

```

**Step 5:** Check the pipeline to see any changes.

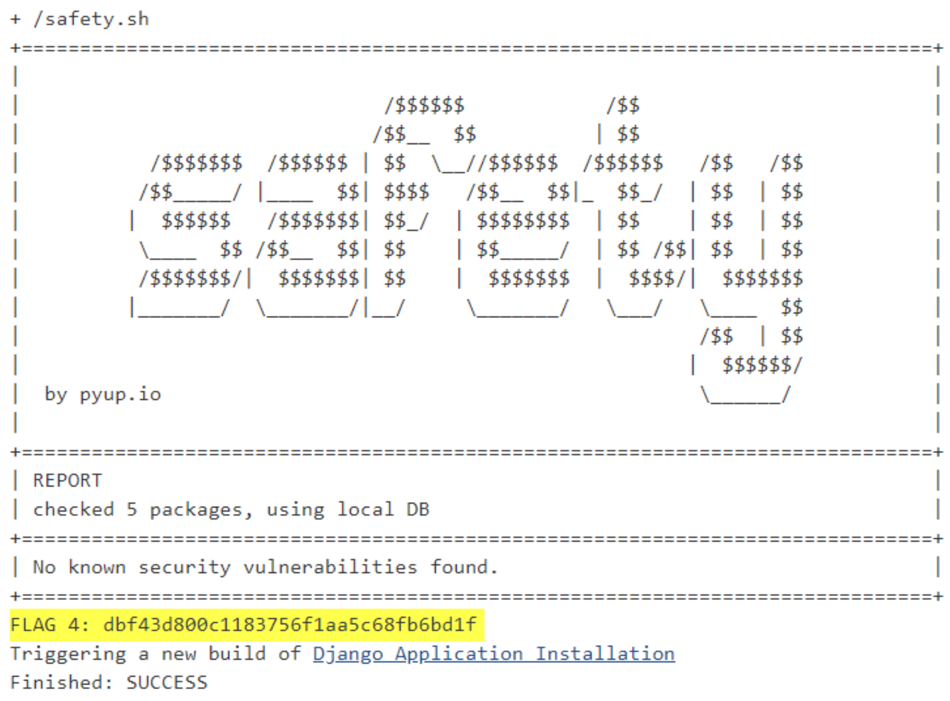


Build Pipeline

Run History

| Pipeline #7 | #2 Safety | #1 Django Application Installation | #1 Selenium Testing | OWASP ZAP Testing |
|---|---|---|---|---|
| | Dec 11, 2020 6:40:06 AM | Dec 11, 2020 6:40:16 AM | Dec 11, 2020 6:40:51 AM | N/A |
| | 0.72 sec | 28 sec | 4 min 48 sec | N/A |

The Safety stage has passed successfully. The logs can be checked from clicking on the “Safety” and opening the console output.

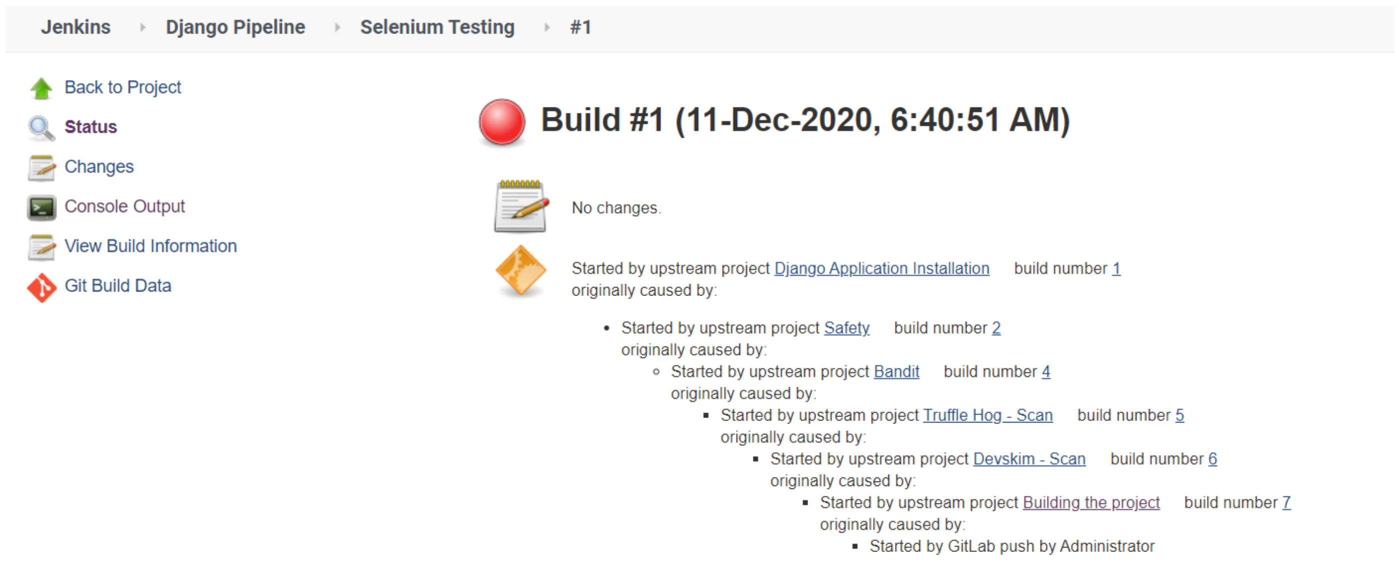


```
+ /safety.sh
+==============================================================================+
|                                                                              |
|                               /$$$$$$            /$$                         |
|                              /$$__  $$          | $$                         |
|           /$$$$$$$  /$$$$$$ | $$  \__//$$$$$$  /$$$$$$   /$$   /$$           |
|          /$$_____/ |____  $$| $$$$   /$$__  $$|_  $$_/  | $$  | $$           |
|         |  $$$$$$   /$$$$$$$| $$_/  | $$$$$$$$  | $$    | $$  | $$           |
|          \____  $$ /$$__  $$| $$    | $$_____/  | $$ /$$| $$  | $$           |
|          /$$$$$$$/|  $$$$$$$| $$    |  $$$$$$$  |  $$$$/|  $$$$$$$           |
|         |_______/  \_______/|__/     \_______/   \___/   \____  $$           |
|                                                          /$$  | $$           |
|                                                         |  $$$$$$/           |
|  by pyup.io                                             \______/            |
|                                                                              |
+==============================================================================+
| REPORT                                                                       |
| checked 5 packages, using local DB                                           |
+==============================================================================+
| No known security vulnerabilities found.                                     |
+==============================================================================+
FLAG 4: dbf43d800c1183756f1aa5c68fb6bd1f
Triggering a new build of Django Application Installation
Finished: SUCCESS
```

**FLAG 4:** dbf43d800c1183756f1aa5c68fb6bd1f

**Selenium Testing**

**Step 1:** Click on the ‘Selenium Testing’ to check the job build page.



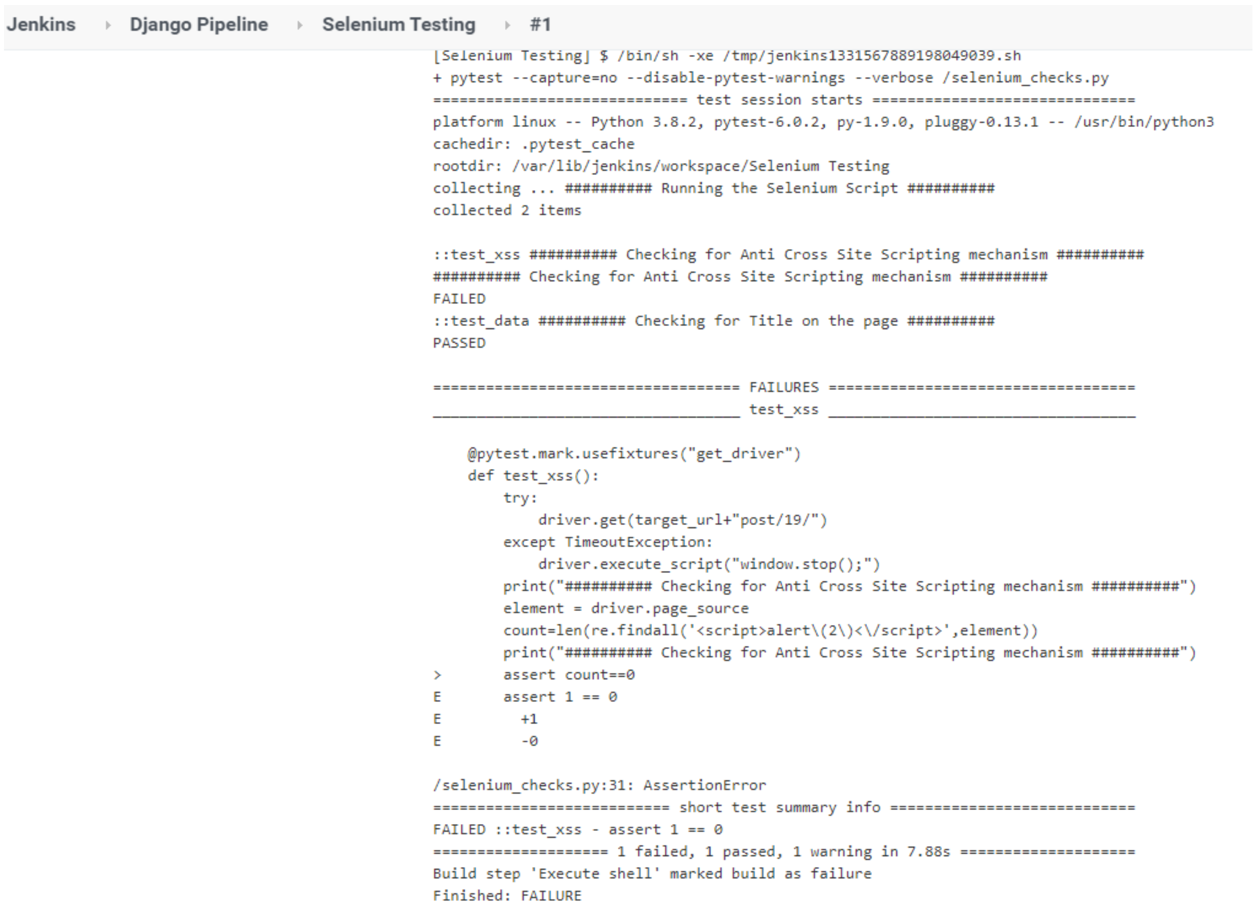
Build #1 (11-Dec-2020, 6:40:51 AM)

No changes.

Started by upstream project Django Application Installation build number 1
originally caused by:

- Started by upstream project Safety build number 2
  originally caused by:
  - Started by upstream project Bandit build number 4
    originally caused by:
    - Started by upstream project Truffle Hog - Scan build number 5
      originally caused by:
      - Started by upstream project Devskim - Scan build number 6
        originally caused by:
        - Started by upstream project Building the project build number 7
          originally caused by:
          - Started by GitLab push by Administrator

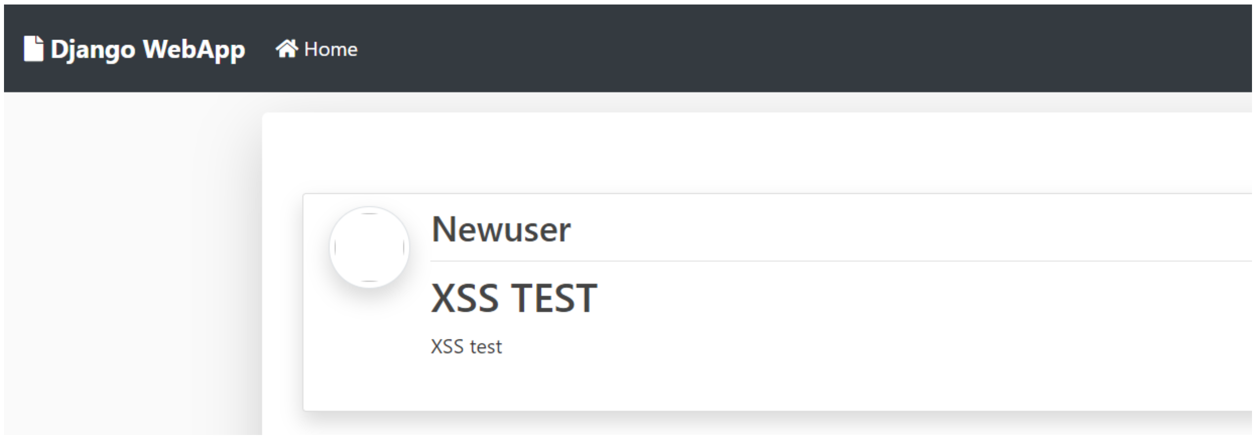
**Step 2:** Click on the “Console Output” to check the issues found in Selenium Testing.



```
[Selenium Testing] $ /bin/sh -xe /tmp/jenkins1331567889198049039.sh
+ pytest --capture=no --disable-pytest-warnings --verbose /selenium_checks.py
============================ test session starts =============================
platform linux -- Python 3.8.2, pytest-6.0.2, py-1.9.0, pluggy-0.13.1 -- /usr/bin/python3
cachedir: .pytest_cache
rootdir: /var/lib/jenkins/workspace/Selenium Testing
collecting ... ########## Running the Selenium Script ##########
collected 2 items

::test_xss ########## Checking for Anti Cross Site Scripting mechanism ##########
########## Checking for Anti Cross Site Scripting mechanism ##########
FAILED
::test_data ########## Checking for Title on the page ##########
PASSED

=================================== FAILURES ===================================
___________________________________ test_xss ___________________________________

    @pytest.mark.usefixtures("get_driver")
    def test_xss():
        try:
            driver.get(target_url+"post/19/")
        except TimeoutException:
            driver.execute_script("window.stop();")
        print("########## Checking for Anti Cross Site Scripting mechanism ##########")
        element = driver.page_source
        count=len(re.findall('<script>alert\(2\)<\/script>',element))
        print("########## Checking for Anti Cross Site Scripting mechanism ##########")
>       assert count==0
E       assert 1 == 0
E         +1
E         -0

/selenium_checks.py:31: AssertionError
========================== short test summary info ===========================
FAILED ::test_xss - assert 1 == 0
==================== 1 failed, 1 passed, 1 warning in 7.88s ====================
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```

The Selenium test checks if there are any alert boxes being prompted while opening the post or not. If it finds any popups it will fail the test.

**Step 3:** Open the post on the test-server and check where the vulnerable parameter is located.

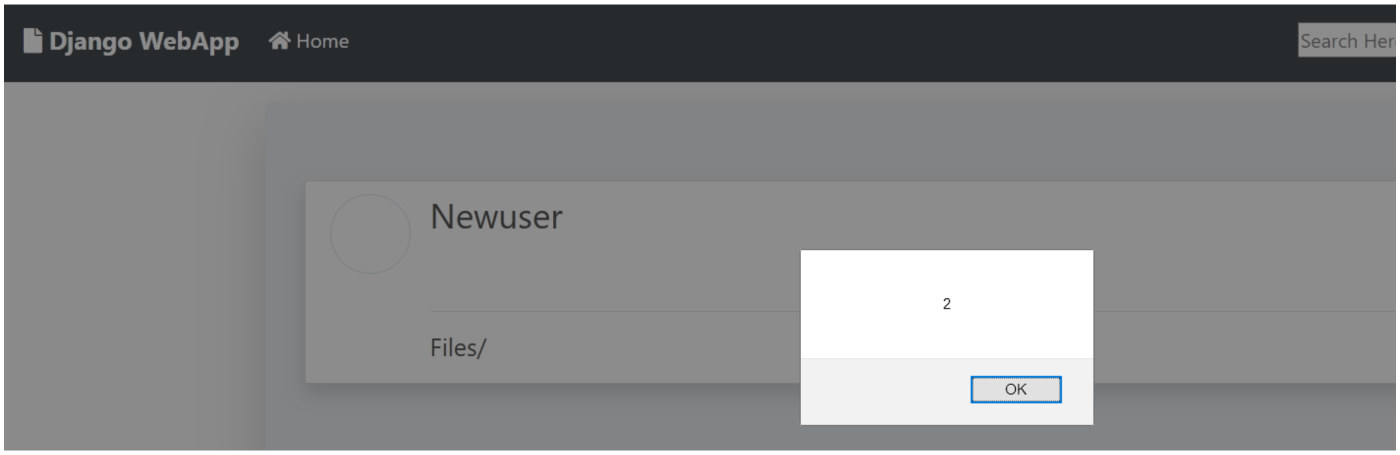


Newuser

XSS TEST

XSS test

Click on the “XSS TEST” post.



Newuser

Files/

2

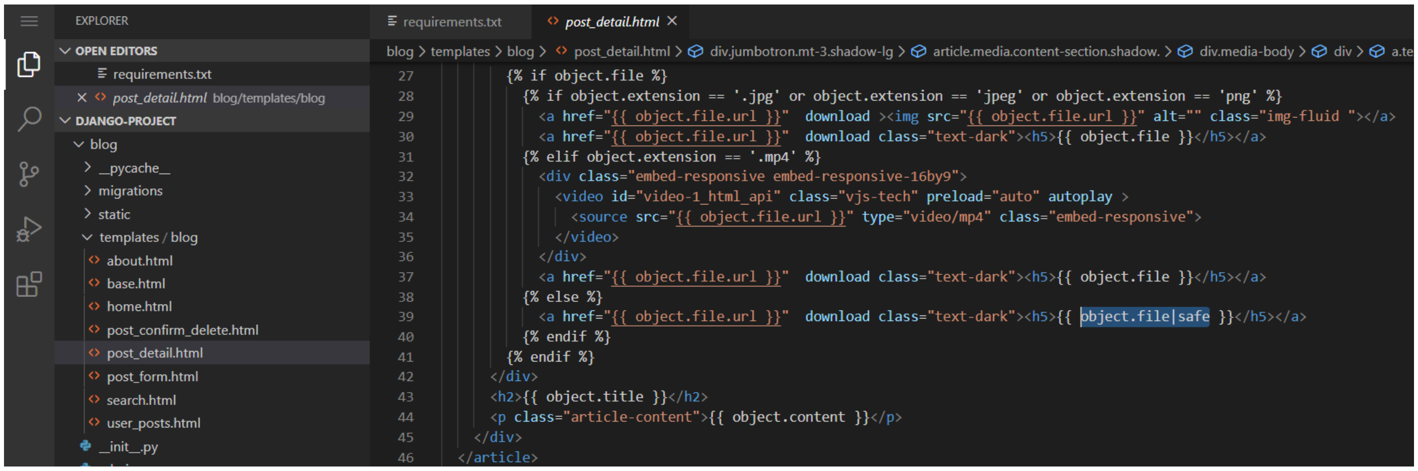
The XSS exists in the application.

**Step 4:** Right-click and check the source code to find ‘2’ in the alert box.

```
        </div>

      <hr>

    </div>
    <div>


        <a href="/media/Files/%3Cscript%3Ealert(2)%3C/script%3E"  download class="text-dark"><h5>Files/<script>alert(2)</script></h5></a>


    </div>
    <h2>XSS TEST</h2>
    <p class="article-content">XSS test</p>
```

The file name can be seen which is actually parsing the data without any sanitization.

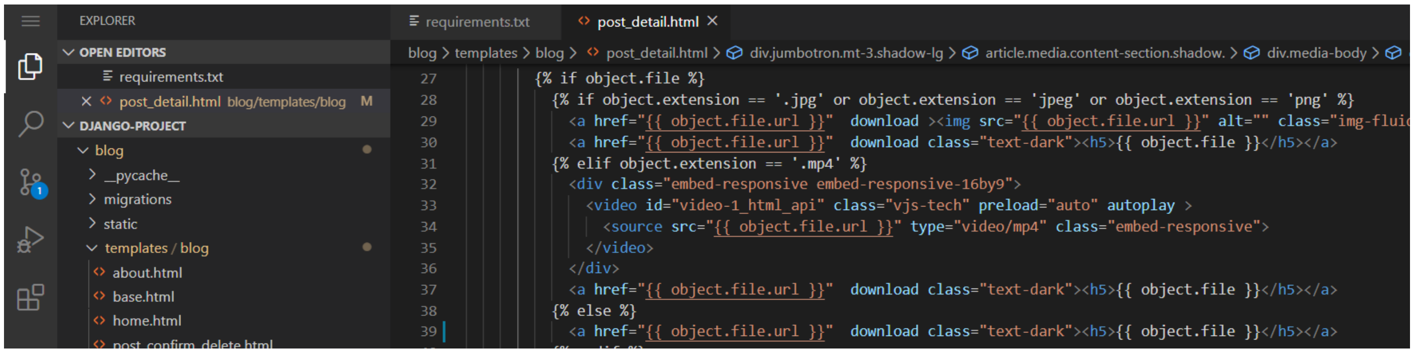
**Step 5:** Open the source code of the page which is responsible to print data about posts. (On VS Code server)



```
27              {% if object.file %}
28                {% if object.extension == '.jpg' or object.extension == 'jpeg' or object.extension == 'png' %}
29                  <a href="{{ object.file.url }}"  download ><img src="{{ object.file.url }}" alt="" class="img-fluid "></a>
30                  <a href="{{ object.file.url }}"  download class="text-dark"><h5>{{ object.file }}</h5></a>
31                {% elif object.extension == '.mp4' %}
32                  <div class="embed-responsive embed-responsive-16by9">
33                    <video id="video-1_html_api" class="vjs-tech" preload="auto" autoplay >
34                      <source src="{{ object.file.url }}" type="video/mp4" class="embed-responsive">
35                    </video>
36                  </div>
37                  <a href="{{ object.file.url }}"  download class="text-dark"><h5>{{ object.file }}</h5></a>
38                {% else %}
39                  <a href="{{ object.file.url }}"  download class="text-dark"><h5>{{ object.file|safe }}</h5></a>
40                {% endif %}
41              {% endif %}
42            </div>
43            <h2>{{ object.title }}</h2>
44            <p class="article-content">{{ object.content }}</p>
45          </div>
46        </article>
```

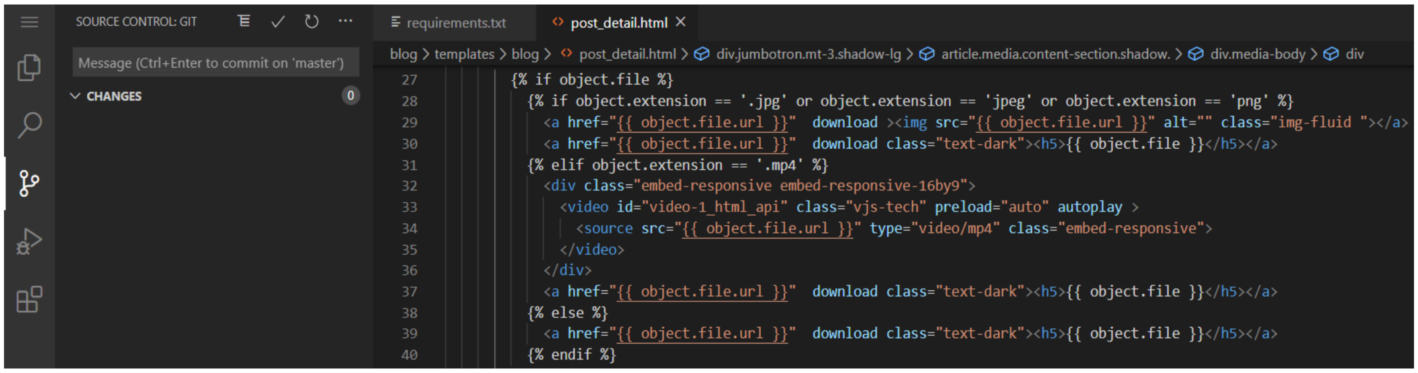
The object.file parameter is found in the post_detail.html of blog/templates/blog/ directory. The function ‘safe’ of Django will disable the HTML escaping on the field.

**Step 6:** Remove the ‘|safe’ from the object.file section of the code to prevent XSS attack. (Line 39)



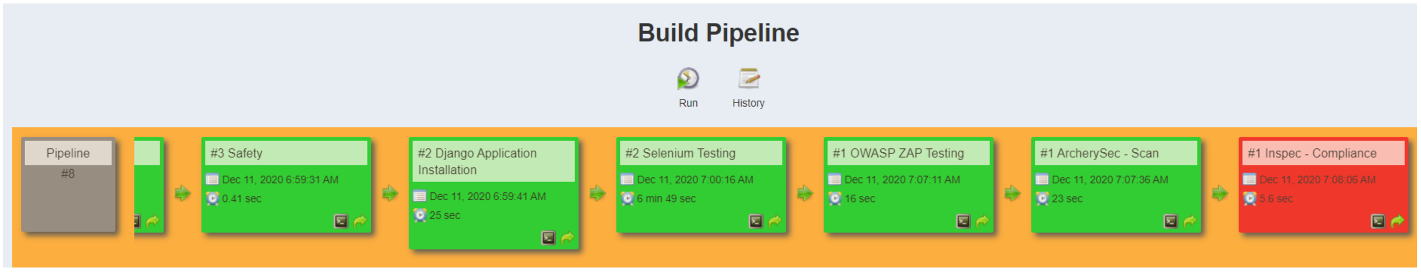
```
{% if object.file %}
  {% if object.extension == '.jpg' or object.extension == 'jpeg' or object.extension == 'png' %}
    <a href="{{ object.file.url }}"  download ><img src="{{ object.file.url }}" alt="" class="img-fluid
    <a href="{{ object.file.url }}"  download class="text-dark"><h5>{{ object.file }}</h5></a>
  {% elif object.extension == '.mp4' %}
    <div class="embed-responsive embed-responsive-16by9">
      <video id="video-1_html_api" class="vjs-tech" preload="auto" autoplay >
        <source src="{{ object.file.url }}" type="video/mp4" class="embed-responsive">
      </video>
    </div>
    <a href="{{ object.file.url }}"  download class="text-dark"><h5>{{ object.file }}</h5></a>
  {% else %}
    <a href="{{ object.file.url }}"  download class="text-dark"><h5>{{ object.file }}</h5></a>
```

Commit and Push the changes to the remote repository.
(Same as Step 10 Onwards of the DevSkim section)



```
{% if object.file %}
  {% if object.extension == '.jpg' or object.extension == 'jpeg' or object.extension == 'png' %}
    <a href="{{ object.file.url }}"  download ><img src="{{ object.file.url }}" alt="" class="img-fluid "></a>
    <a href="{{ object.file.url }}"  download class="text-dark"><h5>{{ object.file }}</h5></a>
  {% elif object.extension == '.mp4' %}
    <div class="embed-responsive embed-responsive-16by9">
      <video id="video-1_html_api" class="vjs-tech" preload="auto" autoplay >
        <source src="{{ object.file.url }}" type="video/mp4" class="embed-responsive">
      </video>
    </div>
    <a href="{{ object.file.url }}"  download class="text-dark"><h5>{{ object.file }}</h5></a>
  {% else %}
    <a href="{{ object.file.url }}"  download class="text-dark"><h5>{{ object.file }}</h5></a>
  {% endif %}
```

**Step 7:** Check the pipeline for the changes.



Build Pipeline

Pipeline #8

#3 Safety — Dec 11, 2020 6:59:31 AM — 0.41 sec

#2 Django Application Installation — Dec 11, 2020 6:59:41 AM — 25 sec

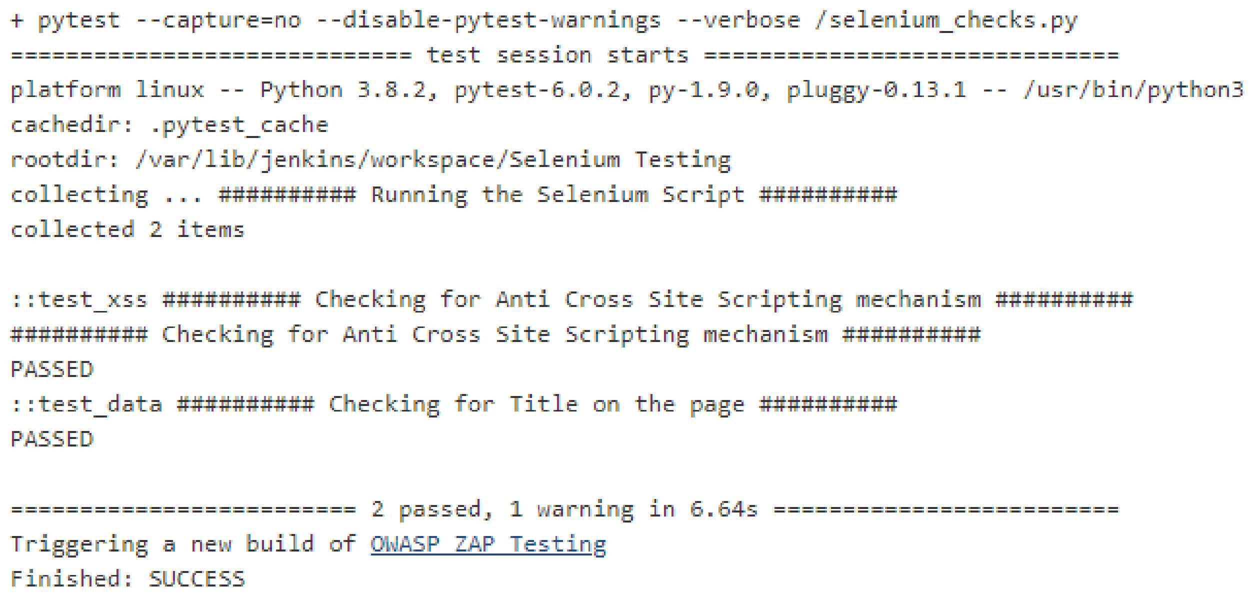
#2 Selenium Testing — Dec 11, 2020 7:00:16 AM — 6 min 49 sec

#1 OWASP ZAP Testing — Dec 11, 2020 7:07:11 AM — 16 sec

#1 ArcherySec - Scan — Dec 11, 2020 7:07:36 AM — 23 sec

#1 Inspec - Compliance — Dec 11, 2020 7:08:06 AM — 5.6 sec

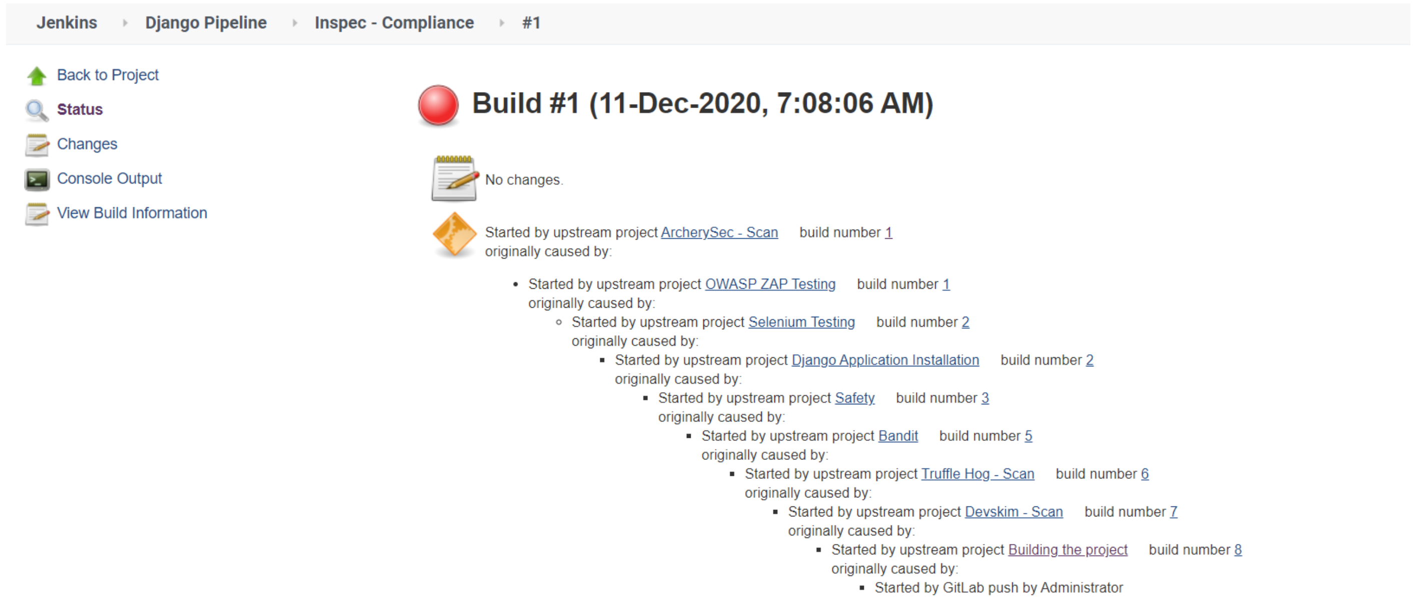
The Selenium Testing stage has passed successfully. The logs can be checked from clicking on the “Selenium Testing” and opening the console output.



```
+ pytest --capture=no --disable-pytest-warnings --verbose /selenium_checks.py
============================= test session starts ==============================
platform linux -- Python 3.8.2, pytest-6.0.2, py-1.9.0, pluggy-0.13.1 -- /usr/bin/python3
cachedir: .pytest_cache
rootdir: /var/lib/jenkins/workspace/Selenium Testing
collecting ... ########## Running the Selenium Script ##########
collected 2 items

::test_xss ########## Checking for Anti Cross Site Scripting mechanism ##########
########## Checking for Anti Cross Site Scripting mechanism ##########
PASSED
::test_data ########## Checking for Title on the page ##########
PASSED

========================= 2 passed, 1 warning in 6.64s =========================
Triggering a new build of OWASP ZAP Testing
Finished: SUCCESS
```

**Inspec Issue**

**Step 1:** Click on the 'Inspec' to check the job build page.



Build #1 (11-Dec-2020, 7:08:06 AM)

No changes.

Started by upstream project ArcherySec - Scan build number 1
originally caused by:

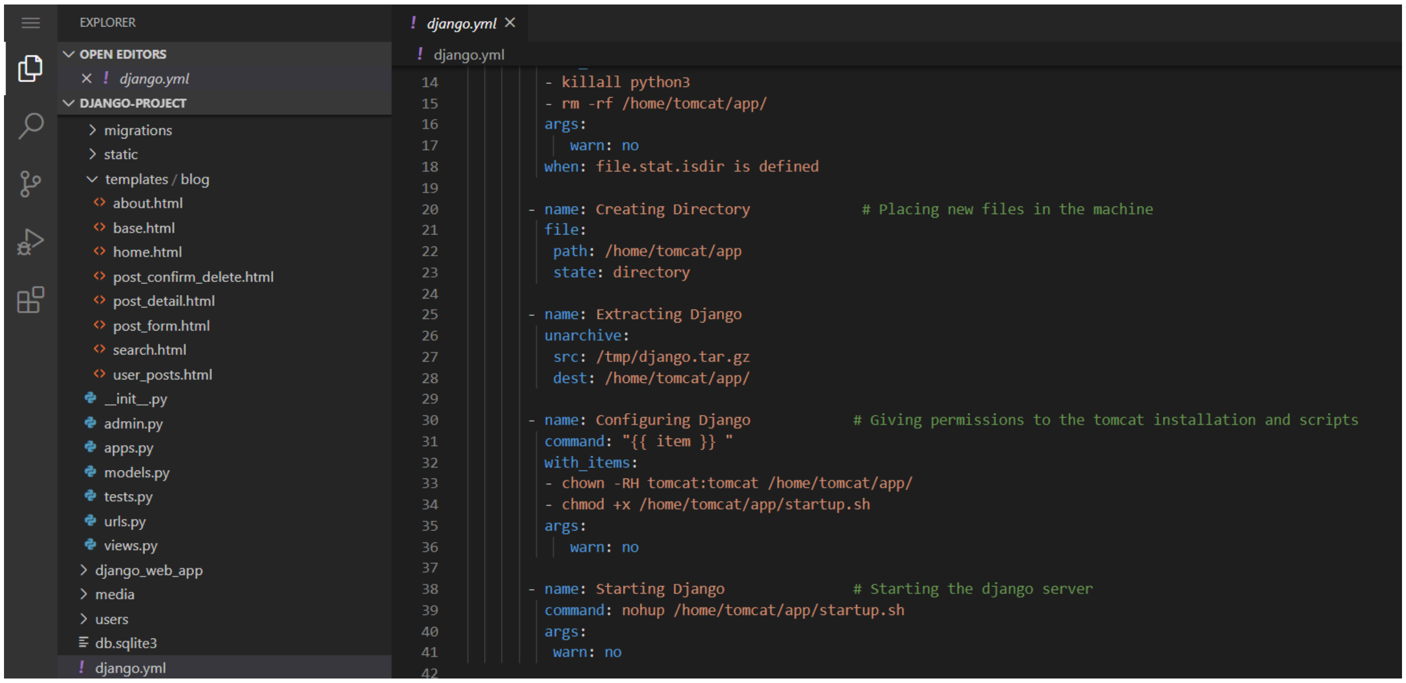
- Started by upstream project OWASP ZAP Testing build number 1
  originally caused by:
  - Started by upstream project Selenium Testing build number 2
    originally caused by:
    - Started by upstream project Django Application Installation build number 2
      originally caused by:
      - Started by upstream project Safety build number 3
        originally caused by:
        - Started by upstream project Bandit build number 5
          originally caused by:
          - Started by upstream project Truffle Hog - Scan build number 6
            originally caused by:
            - Started by upstream project Devskim - Scan build number 7
              originally caused by:
              - Started by upstream project Building the project build number 8
                originally caused by:
                - Started by GitLab push by Administrator

**Step 2:** Click on the "Console Output" to check the issues found in Inspec.

```
+ /inspec.sh

Profile: tests from /django.rb (tests from .django.rb)
Version: (not specified)
Target:  ssh://tomcat@test-server:22

▯[38;5;9m  ▯▯  tomcat.directories: Check for existence and correct permissions of logs file in application directory (3 failed)▯[0m
▯[38;5;9m     ▯▯  Directory /home/tomcat/app/logs is expected to be file
     expected `Directory /home/tomcat/app/logs.file?` to return true, got false▯[0m
▯[38;5;9m     ▯▯  Directory /home/tomcat/app/logs owner is expected to eq "tomcat"

     expected: "tomcat"
          got: nil

     (compared using ==)
▯[0m
▯[38;5;9m     ▯▯  Directory /home/tomcat/app/logs mode is expected to cmp == "0750"
     can't convert nil into Integer▯[0m


Profile Summary: 0 successful controls, ▯[38;5;9m1 control failure▯[0m, 0 controls skipped
Test Summary: 0 successful, ▯[38;5;9m3 failures▯[0m, 0 skipped
Build step 'Execute shell' marked build as failure
Finished: FAILURE
```

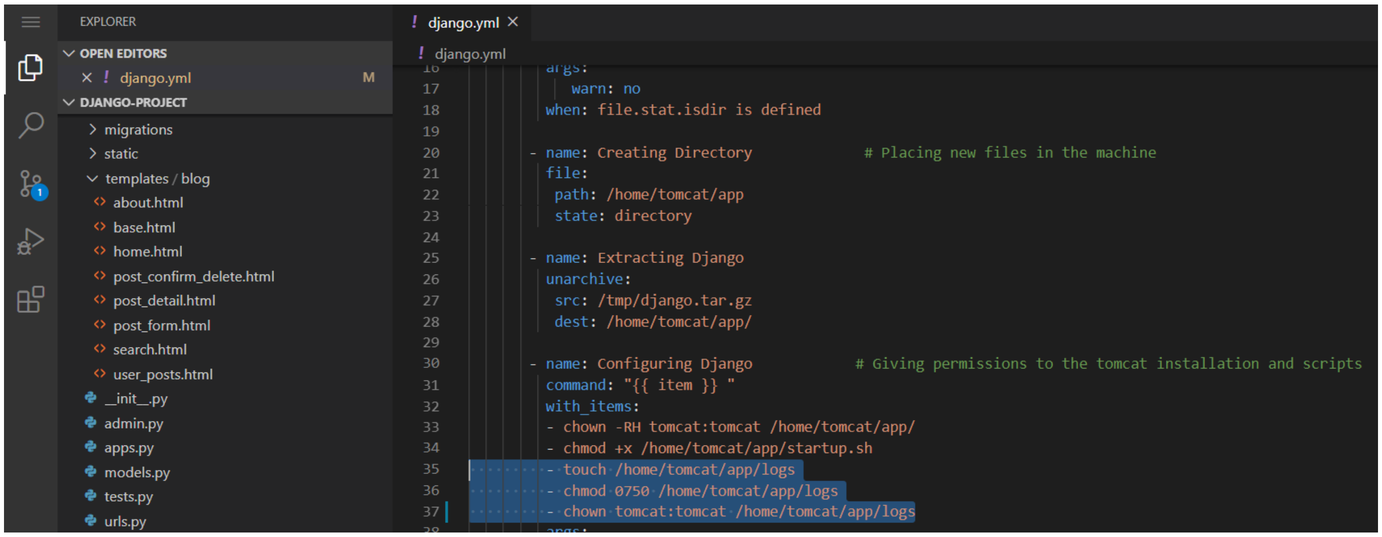
The file /home/tomcat/app/logs in the remote server need to exist with permissions 0750 but currently, the file does not exist.

**Step 3:** Open the Django.yml file which is used by ansible to deploy the application on the remote server.



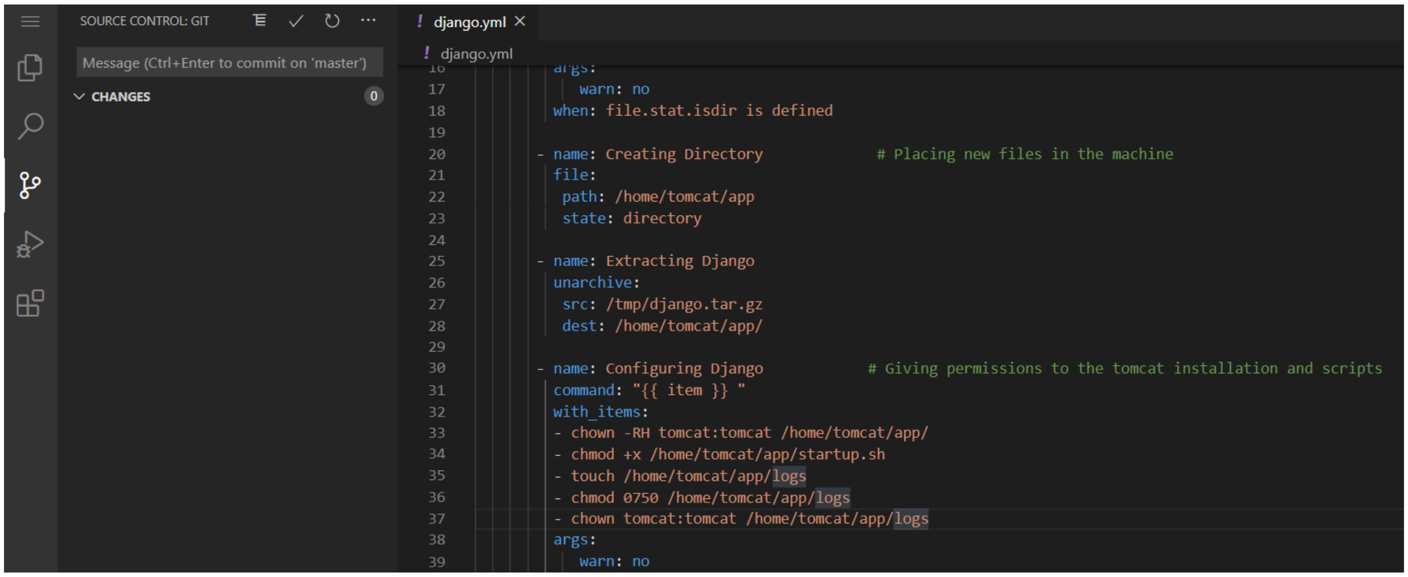
```
    - killall python3
    - rm -rf /home/tomcat/app/
    args:
      warn: no
    when: file.stat.isdir is defined

  - name: Creating Directory             # Placing new files in the machine
    file:
     path: /home/tomcat/app
     state: directory

  - name: Extracting Django
    unarchive:
     src: /tmp/django.tar.gz
     dest: /home/tomcat/app/

  - name: Configuring Django            # Giving permissions to the tomcat installation and scripts
    command: "{{ item }} "
    with_items:
    - chown -RH tomcat:tomcat /home/tomcat/app/
    - chmod +x /home/tomcat/app/startup.sh
    args:
      warn: no

  - name: Starting Django               # Starting the django server
    command: nohup /home/tomcat/app/startup.sh
    args:
     warn: no
```

**Step 4:** Place command to create and set the appropriate permissions on the 'logs' file.



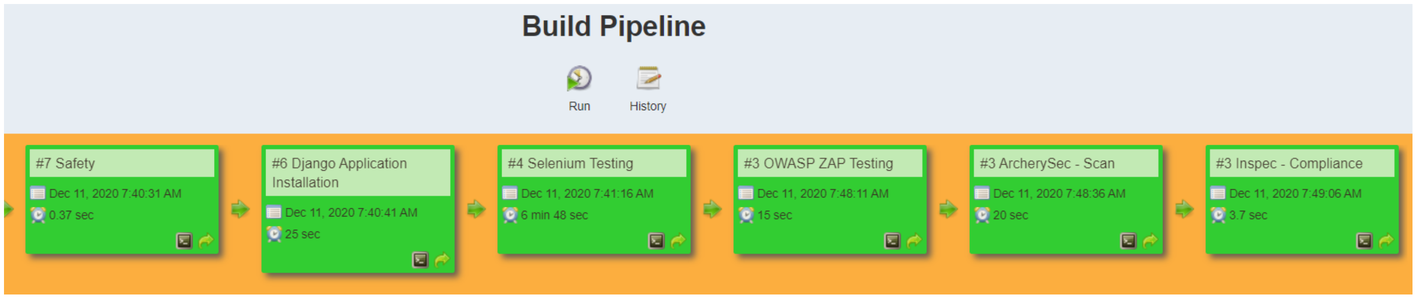
```
      args:
        warn: no
      when: file.stat.isdir is defined

    - name: Creating Directory              # Placing new files in the machine
      file:
       path: /home/tomcat/app
       state: directory

    - name: Extracting Django
      unarchive:
       src: /tmp/django.tar.gz
       dest: /home/tomcat/app/

    - name: Configuring Django           # Giving permissions to the tomcat installation and scripts
      command: "{{ item }} "
      with_items:
      - chown -RH tomcat:tomcat /home/tomcat/app/
      - chmod +x /home/tomcat/app/startup.sh
      - touch /home/tomcat/app/logs
      - chmod 0750 /home/tomcat/app/logs
      - chown tomcat:tomcat /home/tomcat/app/logs
      args:
```

Commit the changes and push the files to the remote repository.
(Same as Step 10 Onwards of the DevSkim section)



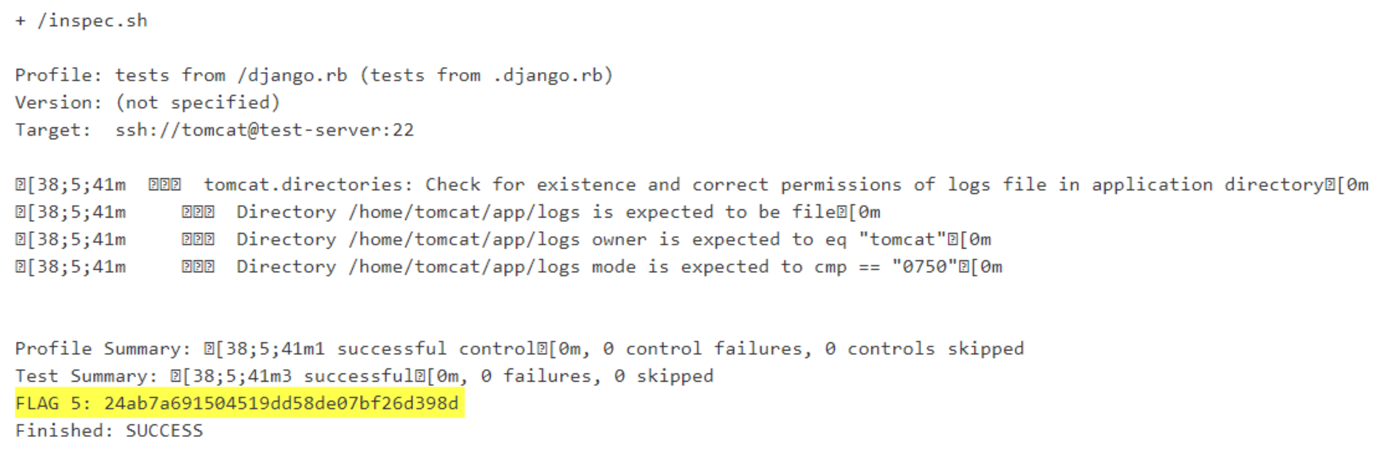
```
      args:
        warn: no
      when: file.stat.isdir is defined

    - name: Creating Directory              # Placing new files in the machine
      file:
       path: /home/tomcat/app
       state: directory

    - name: Extracting Django
      unarchive:
       src: /tmp/django.tar.gz
       dest: /home/tomcat/app/

    - name: Configuring Django           # Giving permissions to the tomcat installation and scripts
      command: "{{ item }} "
      with_items:
      - chown -RH tomcat:tomcat /home/tomcat/app/
      - chmod +x /home/tomcat/app/startup.sh
      - touch /home/tomcat/app/logs
      - chmod 0750 /home/tomcat/app/logs
      - chown tomcat:tomcat /home/tomcat/app/logs
      args:
        warn: no
```

**Step 5:** Check the pipeline for the changes



| #7 Safety | #6 Django Application Installation | #4 Selenium Testing | #3 OWASP ZAP Testing | #3 ArcherySec - Scan | #3 Inspec - Compliance |
|---|---|---|---|---|---|
| Dec 11, 2020 7:40:31 AM | Dec 11, 2020 7:40:41 AM | Dec 11, 2020 7:41:16 AM | Dec 11, 2020 7:48:11 AM | Dec 11, 2020 7:48:36 AM | Dec 11, 2020 7:49:06 AM |
| 0.37 sec | 25 sec | 6 min 48 sec | 15 sec | 20 sec | 3.7 sec |

The Inspec stage has passed successfully. The logs can be checked from clicking on the "Inspec" and opening the console output.



```
+ /inspec.sh

Profile: tests from /django.rb (tests from .django.rb)
Version: (not specified)
Target:  ssh://tomcat@test-server:22

[38;5;41m  ✔✔✔  tomcat.directories: Check for existence and correct permissions of logs file in application directory[0m
[38;5;41m     ✔✔✔  Directory /home/tomcat/app/logs is expected to be file[0m
[38;5;41m     ✔✔✔  Directory /home/tomcat/app/logs owner is expected to eq "tomcat"[0m
[38;5;41m     ✔✔✔  Directory /home/tomcat/app/logs mode is expected to cmp == "0750"[0m


Profile Summary: [38;5;41m1 successful control[0m, 0 control failures, 0 controls skipped
Test Summary: [38;5;41m3 successful[0m, 0 failures, 0 skipped
FLAG 5: 24ab7a691504519dd58de07bf26d398d
Finished: SUCCESS
```

**FLAG 5:** 24ab7a691504519dd58de07bf26d398d

## Learning

Working on a simple DevSecOps pipeline consisting of different components to fix the issues present in the pipeline