kura labs

COHORT 3

Deployment #1

Kerri Smith

27th September 2022

Table of Contents

Summary

CI/CD Pipeline

Continuous Integration and Continuous Deployment (CI/CD) pipelines use automation to improve software delivery throughout the software development life cycle. This deployment exercise demonstrated the steps for setting up a basic CI/CD pipeline consisting of four main stages: CODE, BUILD, TEST, and DEPLOY

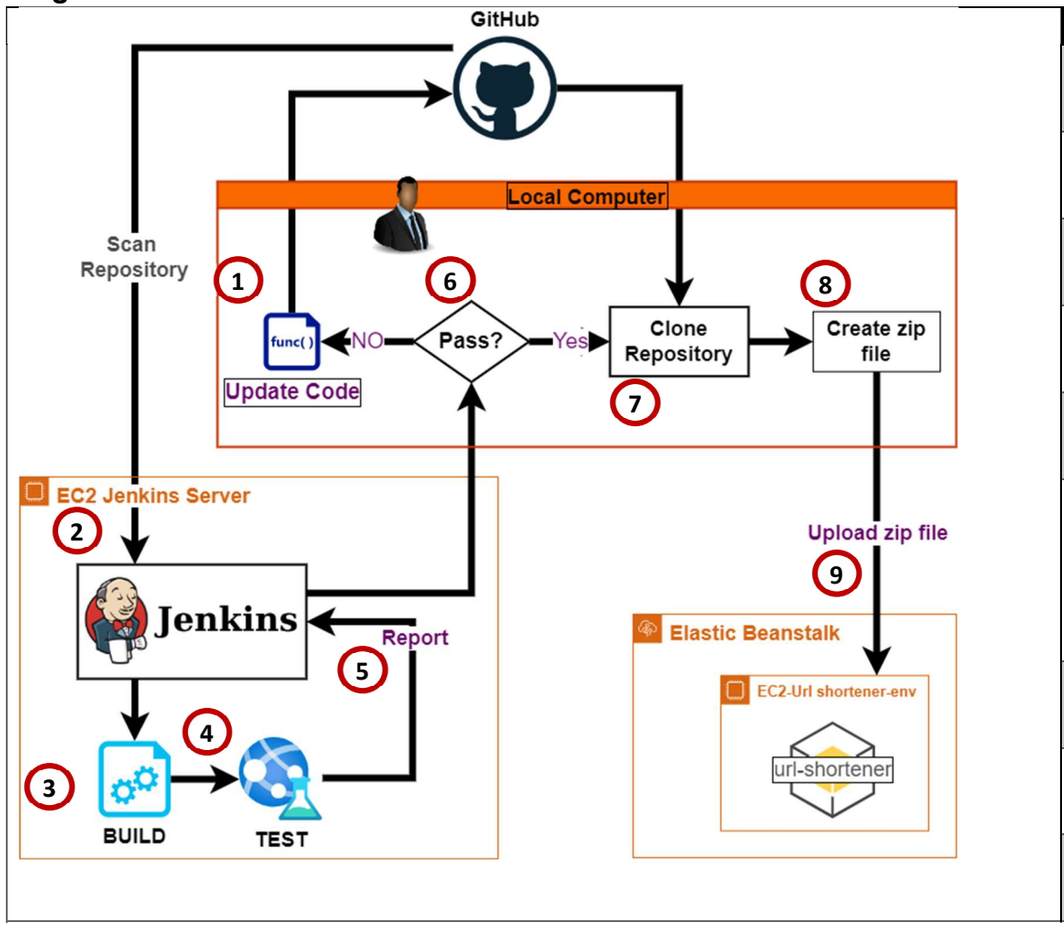
The software application used in this case was a Flask web application called “url shortener”

GitHub was used to manage the code and Jenkins was used to automate the build as well as the testing stages. The application was then manually deployed using AWS Elastic Beanstalk.

kerri.p.h.smith@gmail.com

Pipeline

Diagram

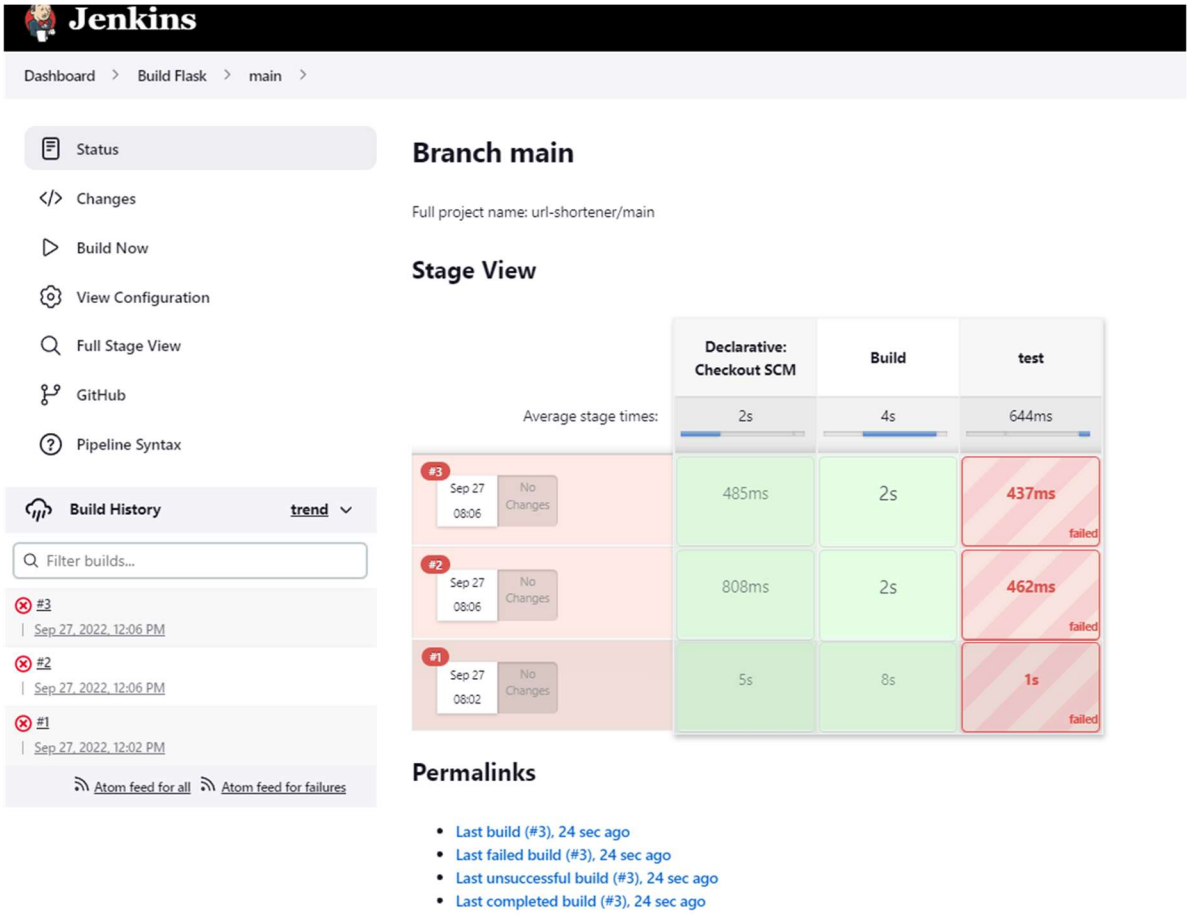


STEP	DESCRIPTION
1	The code is update by the developer and the changes and committed and pushed to GitHub
2	The Jenkins server scans the repository and retrieves the files
3	The Jenkins server builds the application
4	The Jenkins server tests the application
5	The results of the build and test stages are reported in the Jenkins server
6	If both stages are passed, then we proceed to step **7**. If there are errors the error logs and reports are reviewed to identify and isolate the issue. Any code related issued including errors in the Jenkins file are corrected and the updates committed and pushed to GitHub i.e. we proceed to step **1**.
7	The repository is cloned to the local computer
8	The cloned repository is compressed and stored as a zip file
9	The application is deployed using AWS Elastic beanstalk

Observations

Create a multibranch build

At this stage a few of the builds failed because of typos in the Jenkins file and dependencies that were not installed



Missing dependency python3.10-venv

The build failed because ***python3.10-venv*** was not installed see the report below:

Console Output

```
Started by user
12:06:43 Connecting to https://api.github.com using Hobsonkp/****** (GitHub Repo
Jenkins)
Obtained Jenkinsfile from c709cc3ce9f9cd440b84dca22572371844703231 [Pipeline]
Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/url-shortener_main
[Pipeline] { [Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
The recommended git tool is: NONE
using credential GitHubRepo
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/url-
shortener_main/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url
https://github.com/Hobsonkp/kuralabs_deployment_2.git # timeout=10
Fetching without tags
Fetching upstream changes from
https://github.com/Hobsonkp/kuralabs_deployment_2.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials GitHub Repo Jenkins
> git fetch --no-tags --force --progress --
https://github.com/Hobsonkp/kuralabs_deployment_2.git
+refs/heads/main:refs/remotes/origin/main # timeout=10
Checking out Revision c709cc3ce9f9cd440b84dca22572371844703231 (main)
> git config core.sparsecheckout # timeout=10
> git checkout -f c709cc3ce9f9cd440b84dca22572371844703231 # timeout=10 Commit
message: "Update
README.md"
> git rev-list --no-walk c709cc3ce9f9cd440b84dca22572371844703231 # timeout=10
[Pipeline] }
[Pipeline] // stage [Pipeline] withEnv [Pipeline] { [Pipeline] stage [Pipeline] {
(Build)
[Pipeline] sh
The virtual environment was not created successfully because ensurepip is not
available. On

Debian/Ubuntu systems, you need to install the python3-venv package using the
following command.

apt install python3.10-venv
```

```
You may need to use sudo with that command. After installing the python3-venv
package, recreate
your virtual environment.

Failing command: ['/var/lib/jenkins/workspace/url-
shortener_main/test3/bin/python3', '- Im',
'ensurepip', '--upgrade', '--default-pip']

/var/lib/jenkins/workspace/url-shortener_main@tmp/durable-65cb7b2d/script.sh: line
3:
test3/bin/activate: No such file or directory
Defaulting to user installation because normal site-packages is not writeable
Requirement already
satisfied: pip in /var/lib/jenkins/.local/lib/python3.10/site- packages (22.2.2)
Defaulting to user installation because normal site-packages is not writeable
Requirement already
satisfied: attrs==22.1.0 in
/var/lib/jenkins/.local/lib/python3.10/site-packages (from -r requirements.txt
(line 2))

(22.1.0)
Requirement already satisfied: click==8.1.3 in
/var/lib/jenkins/.local/lib/python3.10/site-packages (from -r requirements.txt
(line 3)) (8.1.3)
Requirement already satisfied: flask==2.2.2 in
/var/lib/jenkins/.local/lib/python3.10/site-packages (from -r requirements.txt
(line 4)) (2.2.2)
Requirement already satisfied: iniconfig==1.1.1 in
/var/lib/jenkins/.local/lib/python3.10/site-packages (from -r requirements.txt
(line 5)) (1.1.1)
Requirement already satisfied: itsdangerous==2.1.2 in
/var/lib/jenkins/.local/lib/python3.10/site-packages (from -r requirements.txt
(line 6)) (2.1.2)
Requirement already satisfied: jinja2==3.1.2 in
/var/lib/jenkins/.local/lib/python3.10/site-packages (from -r requirements.txt
(line 7)) (3.1.2)
Requirement already satisfied: markupsafe==2.1.1 in
/var/lib/jenkins/.local/lib/python3.10/site-packages (from -r requirements.txt
(line 8)) (2.1.1)
Requirement already satisfied: packaging==21.3 in
/var/lib/jenkins/.local/lib/python3.10/site-packages (from -r requirements.txt
(line 9)) (21.3)
Requirement already satisfied: pluggy==1.0.0 in
/var/lib/jenkins/.local/lib/python3.10/site-packages (from -r requirements.txt
(line 10)) (1.0.0)
Requirement already satisfied: py==1.11.0 in
/var/lib/jenkins/.local/lib/python3.10/site-packages (from -r requirements.txt
(line 11)) (1.11.0)
```

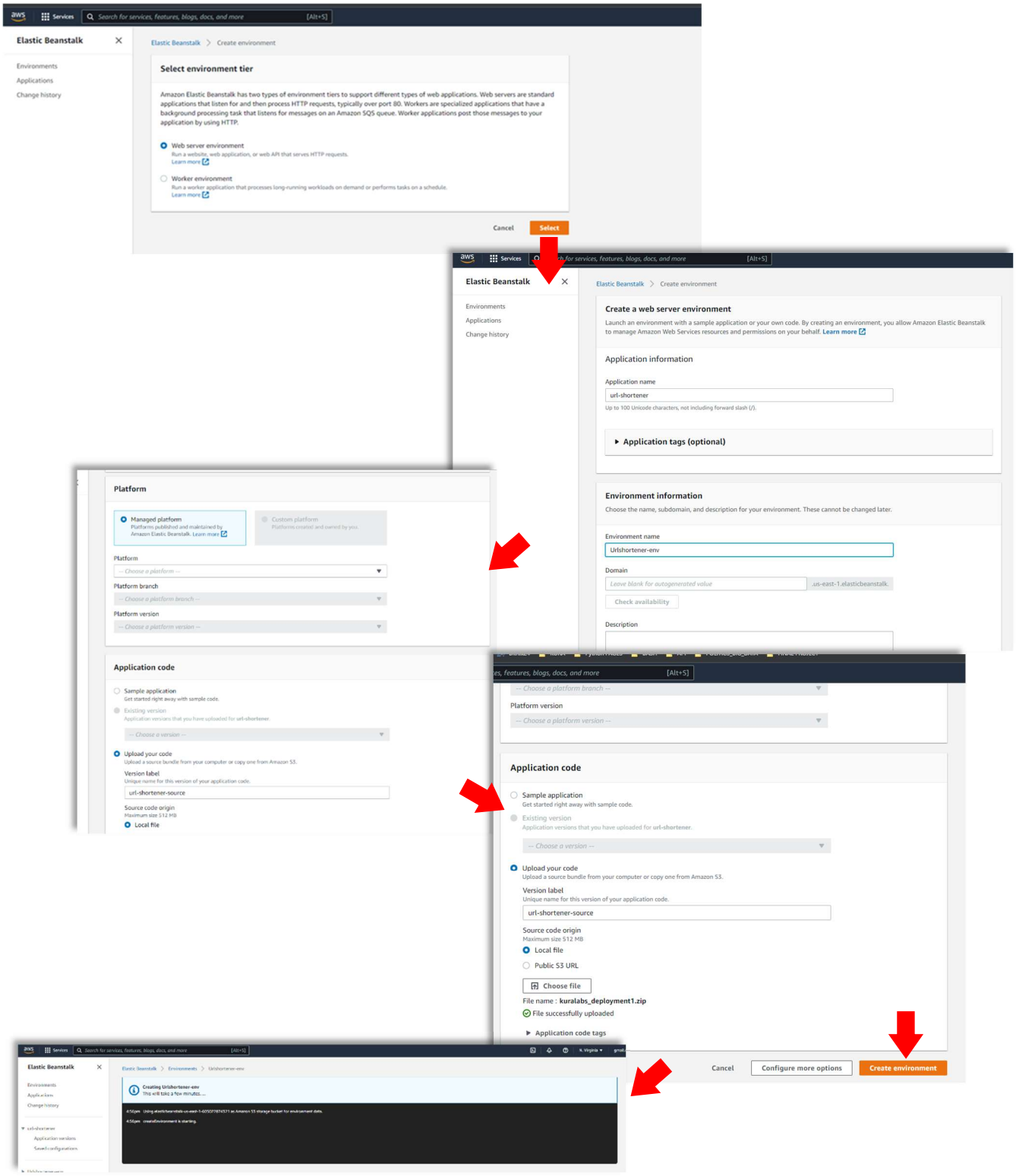
```
Requirement already satisfied: pyparsing==3.0.9 in
/var/lib/jenkins/.local/lib/python3.10/site-packages (from -r requirements.txt
(line 12)) (3.0.9)
Requirement already satisfied: pytest==7.1.2 in
/var/lib/jenkins/.local/lib/python3.10/site-packages (from -r requirements.txt
(line 13)) (7.1.2)
Requirement already satisfied: tomli==2.0.1 in
/var/lib/jenkins/.local/lib/python3.10/site-packages (from -r requirements.txt
(line 14)) (2.0.1)
Requirement already satisfied: werkzeug==2.2.2 in
/var/lib/jenkins/.local/lib/python3.10/site-packages (from -r requirements.txt
(line 15)) (2.2.2)
/var/lib/jenkins/workspace/url-shortener_main@tmp/durable-65cb7b2d/script.sh: line
7: flask: command not found
[Pipeline] } [Pipeline] // stage [Pipeline] stage [Pipeline] { (test)
[Pipeline] sh
/var/lib/jenkins/workspace/url-shortener_main@tmp/durable-c5eae38a/script.sh: line
2: test3/bin/activate: No such file or directory/var/lib/jenkins/workspace/url-
shortener_main@tmp/durable-c5eae38a/script.sh: line 3: py.test: command not found
Post stage
[Pipeline] junit
Recording test results
No test report files were found. Configuration error?
Error when executing always post condition:hudson.AbortException: No test report
files were found. Configuration error?
at hudson.tasks.junit.JUnitParser$ParseResultCallable.invoke(JUnitParser.java:184)
at hudson.FilePath.act(FilePath.java:1192)
at hudson.FilePath.act(FilePath.java:1175)
at hudson.tasks.junit.JUnitParser.parseResult(JUnitParser.java:118)
at hudson.tasks.junit.JUnitResultArchiver.parse(JUnitResultArchiver.java:157)
at
hudson.tasks.junit.JUnitResultArchiver.parseAndSummarize(JUnitResultArchiver.java:
251)
at
hudson.tasks.junit.pipeline.JUnitResultsStepExecution.run(JUnitResultsStepExecutio
n.java:63)
at
hudson.tasks.junit.pipeline.JUnitResultsStepExecution.run(JUnitResultsStepExecutio
n.java:29)
at
org.jenkinsci.plugins.workflow.steps.SynchronousNonBlockingStepExecution.lambda$st
art$0(SynchronousNonBlockingStepExecution.java:47)
at
java.base/java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:515)
at java.base/java.util.concurrent.FutureTask.run(FutureTask.java:264)
at
java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.jav
a:1128)
```

```
at
java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.ja
va:628)
at java.base/java.lang.Thread.run(Thread.java:829)
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 127
Could not update commit status, please check if your scan credentials belong to a
member of the organization or a collaborator of the repository and repo:status
scope is selected
GitHub has been notified of this commit’s build result
Finished: FAILURE
```

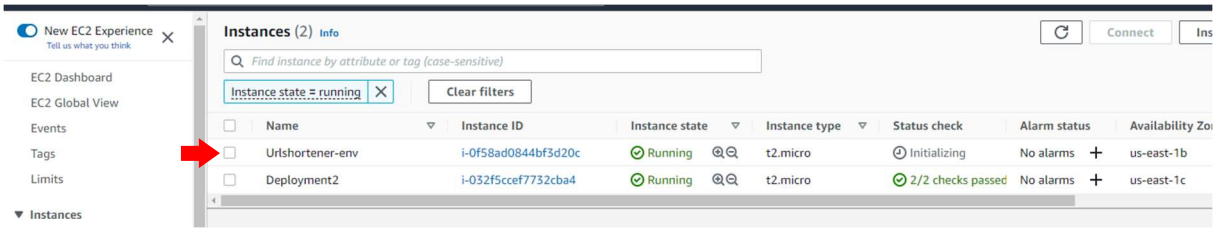
Create a new environment in AWS Elastic Beanstalk

Screenshots of Elastic beanstalk deployment

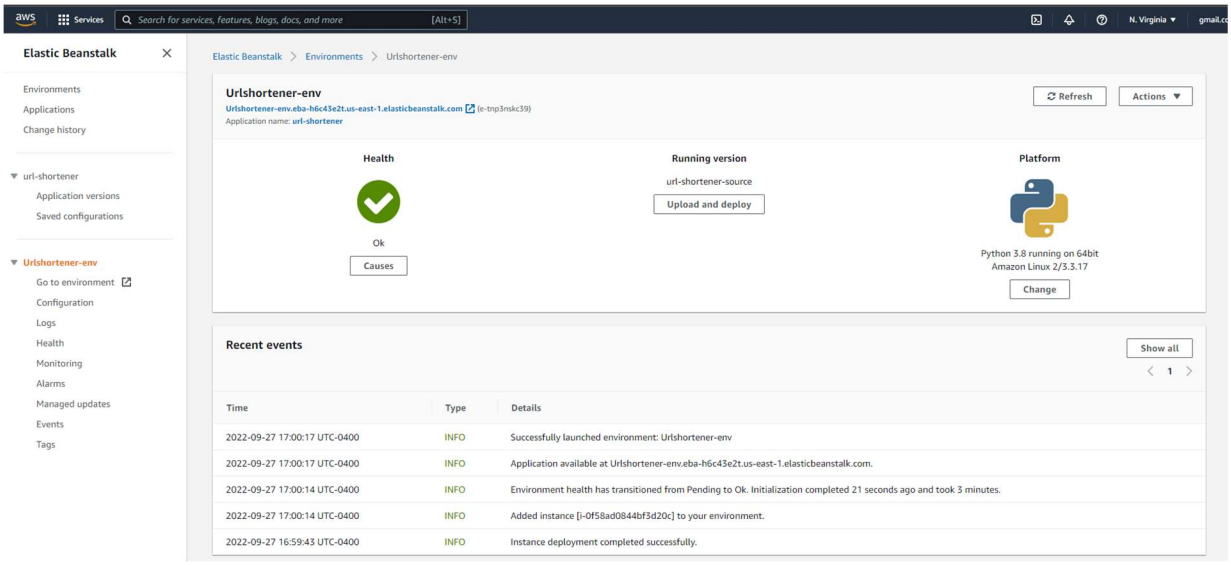
Creating new environment



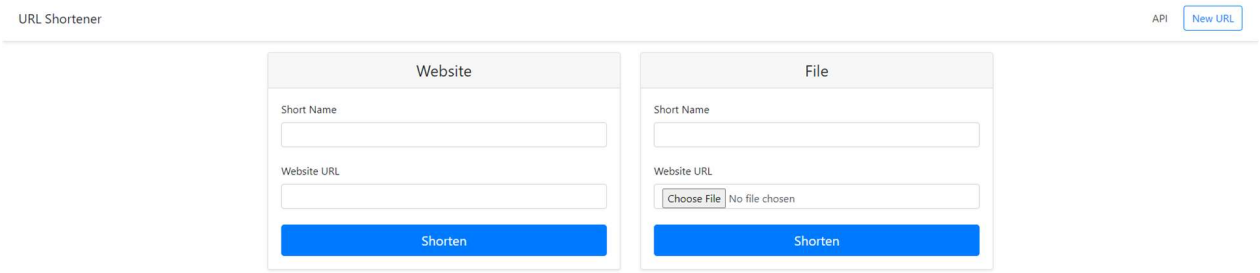
EC2 instance created



Environment status



Application running



Proposed Improvements

The pipeline could be improved in the following ways:

1. Automating or scheduling builds in Jenkins based on commits or updates to the GitHub repository.
2. Automated alerts via email to the developer
3. Automating the deployment stage
4. Including additional testing of the application at the test stage
5. Include automated monitoring and reporting after deployment

Appendix A – List of Terms & Acronyms

TERM	DEFINITION
AWS	Amazon Web Services
CI/CD	Continuous Integration and Continuous Deployment (CI/CD) pipelines use automation to improve software delivery throughout the software development life cycle.
Amazon EC2 or EC2	Amazon Elastic Compute Cloud (EC2) is a part of Amazon Web Services (AWS) cloud-computing platform. It allows users to rent virtual computers on which to run their own computer applications. EC2 encourages scalable deployment of applications by providing a web service through which a user can boot and configure a virtual machine called an "instance". A user can create, launch, and terminate server-instances as needed.
Git	Git is free and open-source software for distributed version control for coordinating work among programmers collaboratively developing source code during software development.
GitHub	GitHub, Inc., is an Internet hosting service for software development and version control using Git. It provides the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis.
Jenkins	Jenkins is an open-source automation server. It can automate the parts of software development related to building, testing, and deploying. The Jenkins server facilitates continuous integration and continuous delivery.
Python3	Python is a high-level, general-purpose programming language.
PIP	Package Installer for Python (pip) is a package-management system written in Python and is used to install and manage software packages
Flask	Flask is a Web Server Gateway Interface (WSGI) web application framework written in Python. It is classified as a microframework because it does not require particular tools or libraries
Elastic Beanstalk or EB	AWS Elastic Beanstalk is an orchestration service offered by Amazon Web Services for deploying applications which orchestrates various AWS services, including EC2, S3, Simple Notification Service (SNS), CloudWatch, autoscaling, and Elastic Load Balancers.