There were plenty of manual steps in creating this server. The first thing I noticed is that we are not at the point where we are using the aws cli to build resources. The console is fine for this type of project, but I'm looking forward to the day where we play around with the cli and write scripts in lieu of what we did for this assignment.

I made the mistake of not having the instance installed before Wednesday class and the act of not having completed the Jenkins deploy before we started creating snapshots and new volumes disoriented me. So, I started everything over and this time I did 2 things differently, 1) I created a script that automatically installs the server and 2) I documented the procedure for creating a new instance from a snapshot. I also realized I hadn't referenced the updated deployment document and wondered where it mentioned Elastic Beanstalk anywhere.

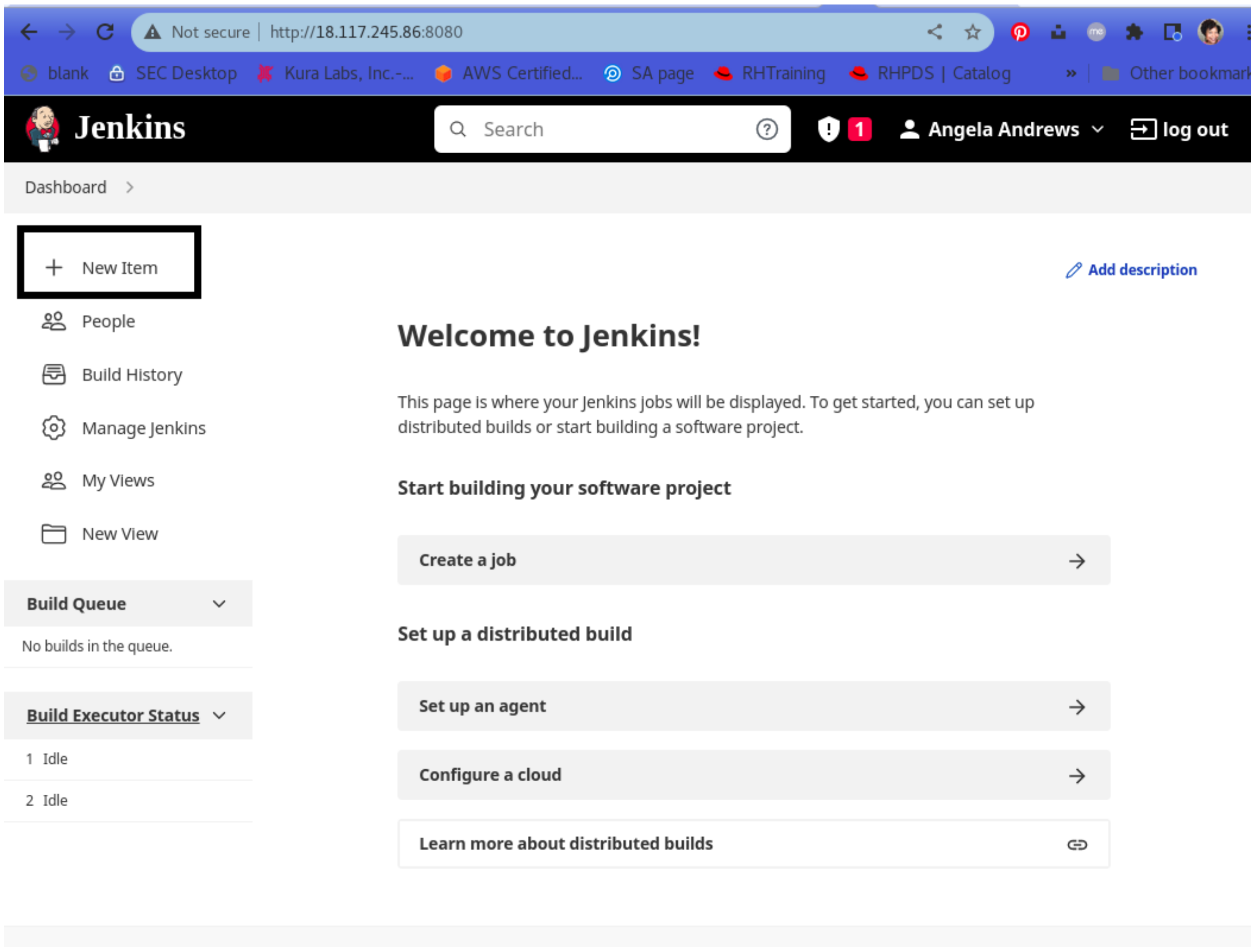
There were many resources I used to get my new server built from the snapshot as well as debugging the install script.

Once I'd created the new instance, I started building a script that would update the server, install the dependencies, start the service and return the DNS name, IP address and initial admin password to me.

Since bash is not idempotent, the script would fail at some places. I figured out how to get to the bottom of the script before I started getting end of file errors. I debugged some more and realized that a late addition if statement wasn't closed with fi. Once that was solved and the system running, I spun up another instance and tested the script on a vanilla install. It worked as expected. Jenkins was started and running and I was able to access it from the public IP and create the admin account.

Here are a list of resources that proved beneficial:
Unexpected End of file error
How to install pip and venv
How to install Jenkins on Ubuntu 22.04
Retrieve Instance metadata from EC2 instance
No installation candidate error
How to move an instance to a different AZ

Once I created my personal access token, I began working on the build.


Not secure | http://18.117.245.86:8080

Jenkins

Search

Angela Andrews log out

Dashboard >

New Item

People

Build History

Manage Jenkins

My Views

New View

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Add description

Welcome to Jenkins!

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

Start building your software project

Create a job

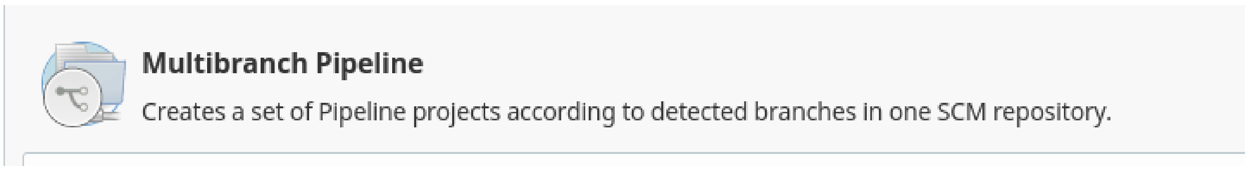
Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

The documentation is lacking here. It’s asking for an item name, but nothing instructing you on what.

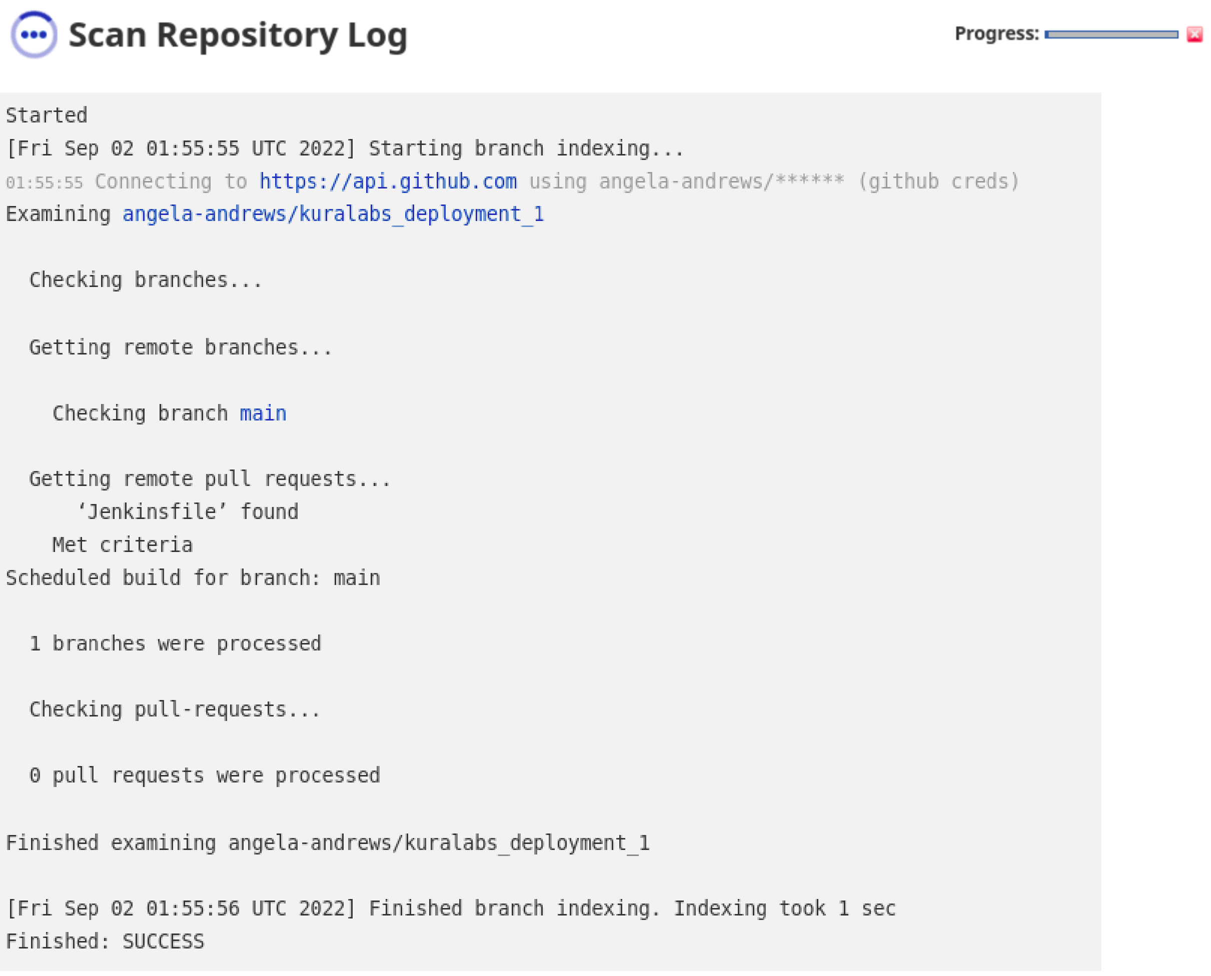

Multibranch Pipeline

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

I clicked on New Item and entered an item name: url-shortner-app. Next I clicked on Branch Sources and selected url-shortener-app. In the dialog box, I entered my github username and my personal access token in the password field and ‘github creds’ in the description field. Clicked Add.

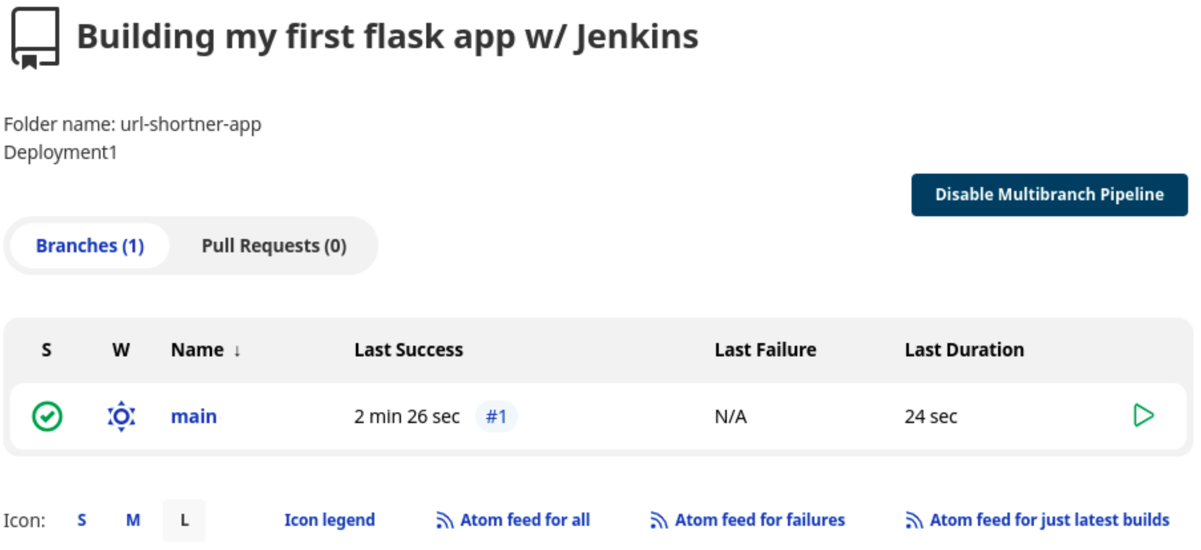
I could now select my github creds, so I pasted in the forked repo URL and clicked validate. The message “Credentials OK” was returned. Under the build configuration section, by Jenkinsfile was in the mode and Script Path was Jenkinsfile mirroring the deployment pdf. I clicked save.

The scan repository log view appeared and it showed a successful branch indexing, but the progress bar was not progressing.



```
Started
[Fri Sep 02 01:55:55 UTC 2022] Starting branch indexing...
01:55:55 Connecting to https://api.github.com using angela-andrews/****** (github creds)
Examining angela-andrews/kuralabs_deployment_1

  Checking branches...

  Getting remote branches...

    Checking branch main

  Getting remote pull requests...
      ‘Jenkinsfile’ found
    Met criteria
Scheduled build for branch: main

  1 branches were processed

  Checking pull-requests...

  0 pull requests were processed

Finished examining angela-andrews/kuralabs_deployment_1

[Fri Sep 02 01:55:56 UTC 2022] Finished branch indexing. Indexing took 1 sec
Finished: SUCCESS
```

I clicked on status and saw my build.

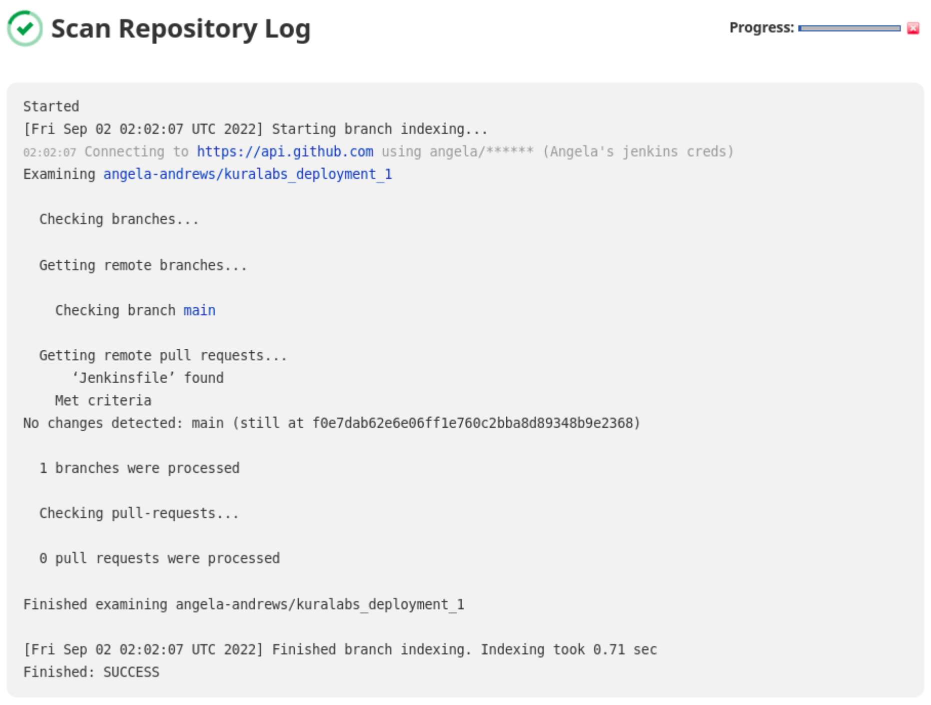

Building my first flask app w/ Jenkins
Disable Multibranch Pipeline
Branches (1)
Pull Requests (0)
Icon: S M L
Icon legend
Atom feed for all
Atom feed for failures
Atom feed for just latest builds

Folder name: url-shortner-app
Deployment1

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☼	main	2 min 26 sec #1	N/A	24 sec

I noticed in the doc that it had another branch source image. I went back to the build and added the Jenkins credential of my admin user created previously then clicked add then save. This document needs page numbers, I'd scrolled too far and did neve notice. Also, the images need to be smaller to fit more information on the page.

The scan repository log screen showed finished success.



Scan Repository Log

Progress:

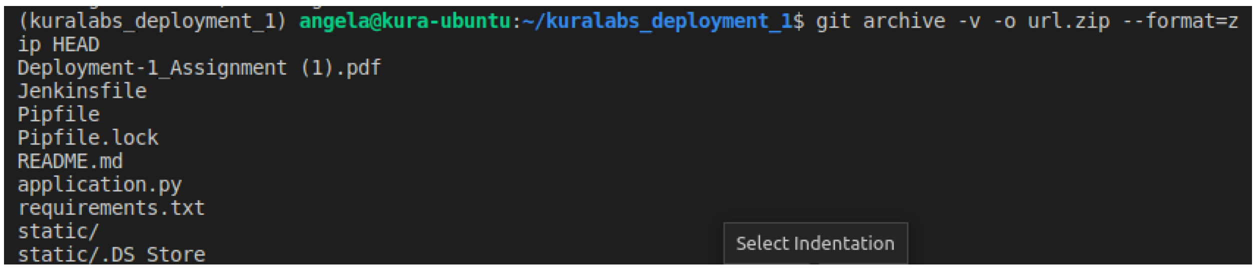
```
Started
[Fri Sep 02 02:02:07 UTC 2022] Starting branch indexing...
02:02:07 Connecting to https://api.github.com using angela/****** (Angela's jenkins creds)
Examining angela-andrews/kuralabs_deployment_1

  Checking branches...

  Getting remote branches...

    Checking branch main

  Getting remote pull requests...
      'Jenkinsfile' found
    Met criteria
No changes detected: main (still at f0e7dab62e6e06ff1e760c2bba8d89348b9e2368)

  1 branches were processed

  Checking pull-requests...

  0 pull requests were processed

Finished examining angela-andrews/kuralabs_deployment_1

[Fri Sep 02 02:02:07 UTC 2022] Finished branch indexing. Indexing took 0.71 sec
Finished: SUCCESS
```

Elastic Beanstalk

The repo files have been cloned prior to this step, so the first bullet point is a bit redundant. We needed the url-shortner repo to configure Jenkins. I then followed the directions to create an application source bundle:

I used the git archive command:

```
$ git archive -v -o url.zip --format=zip HEAD
```



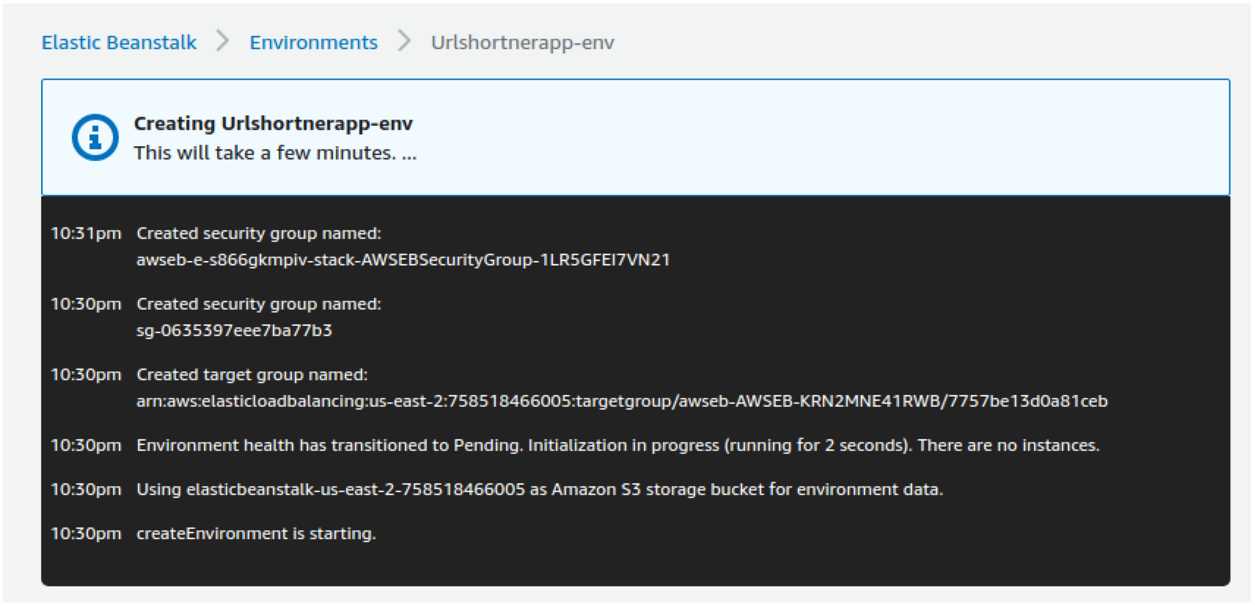
```
(kuralabs_deployment_1) angela@kura-ubuntu:~/kuralabs_deployment_1$ git archive -v -o url.zip --format=z
ip HEAD
Deployment-1_Assignment (1).pdf
Jenkinsfile
Pipfile
Pipfile.lock
README.md
application.py
requirements.txt
static/
static/.DS Store
```

Now that my repo is archived. Log into Elastic Beanstalk.

This is the second time my vm locked up (can't switch between browser, vscode or terminal) I've bumped the memory up to 4GB, but it's still freezing.

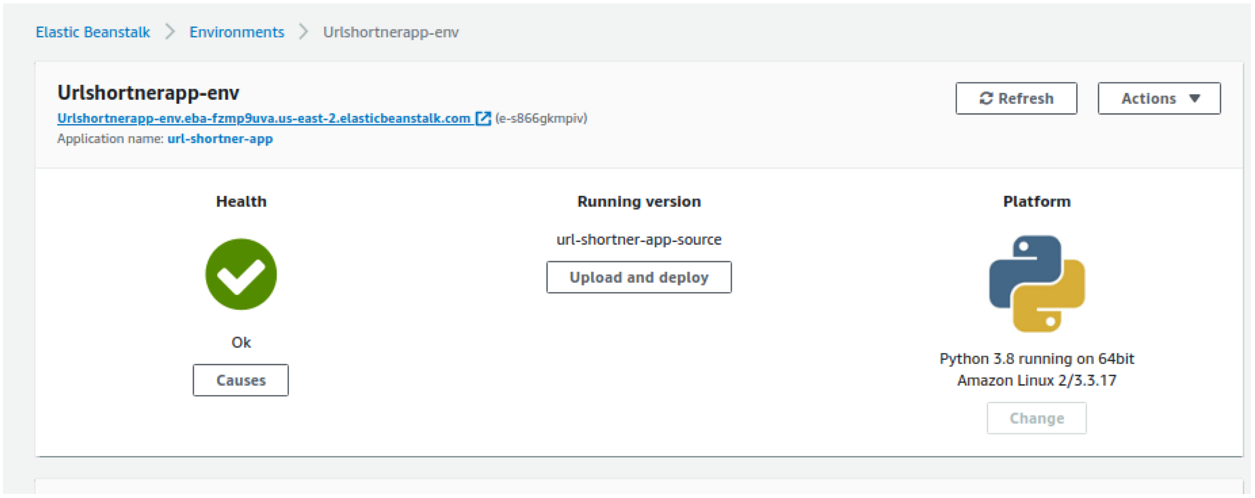
Clicked on create application and filled out the form:

- Application Name: url-shortner-app
- Application tags: key - Purpose Value - Deployment1
- Platform: Python remaining default
- Application code: Upload your code
- Source code origin: Local file, uploaded url.zip
- Clicked create application



Creating Urlshortnerapp-env
This will take a few minutes. ...

```
10:31pm  Created security group named:
         awseb-e-s866gkmpiv-stack-AWSEBSecurityGroup-1LR5GFEI7VN21

10:30pm  Created security group named:
         sg-0635397eee7ba77b3

10:30pm  Created target group named:
         arn:aws:elasticloadbalancing:us-east-2:758518466005:targetgroup/awseb-AWSEB-KRN2MNE41RWB/7757be13d0a81ceb

10:30pm  Environment health has transitioned to Pending. Initialization in progress (running for 2 seconds). There are no instances.

10:30pm  Using elasticbeanstalk-us-east-2-758518466005 as Amazon S3 storage bucket for environment data.

10:30pm  createEnvironment is starting.
```



Urlshortnerapp-env
Urlshortnerapp-env.eba-fzmp9uva.us-east-2.elasticbeanstalk.com (e-s866gkmpiv)
Application name: url-shortner-app

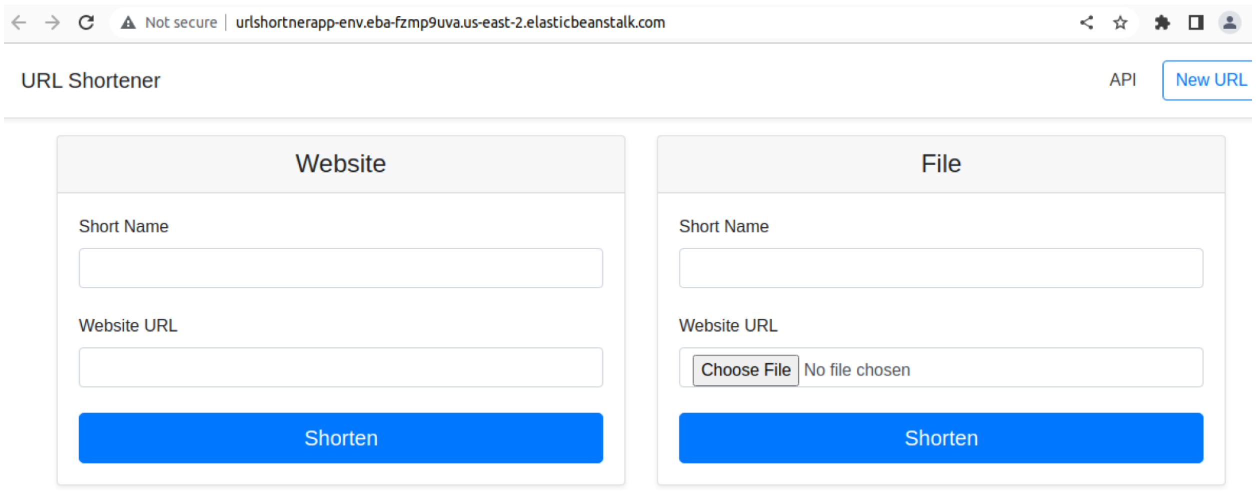
Health: Ok
Running version: url-shortner-app-source
Platform: Python 3.8 running on 64bit Amazon Linux 2/3.3.17

http://urlshortnerapp-env.eba-fzmp9uva.us-east-2.elasticbeanstalk.com/

This is a simple app and could very well run in a container. That would remove the need for Elastic Beanstalk, creating a zip file and the complexity of Jenkins. Although Jenkins can be used for containerized applications, the examples we've seen so far are verbose and hard to follow.

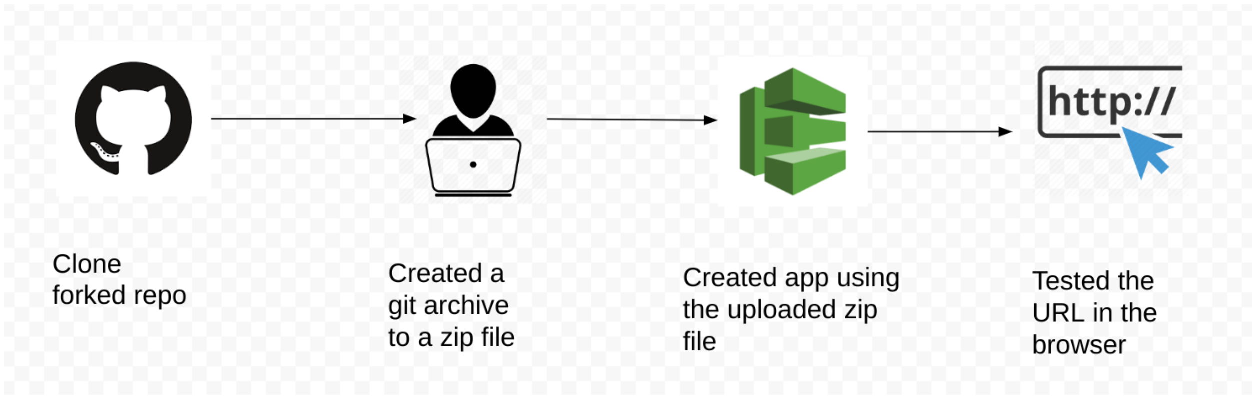
We could have:

- downloaded a ubuntu container image
- created a Containerfile and built the image with the code from the repo
- pushed the image to Docker Hub or Elastic Container Registry
- Deployed the image to ECS or EKS


Not secure | urlshortnerapp-env.eba-fzmp9uva.us-east-2.elasticbeanstalk.com
URL Shortener
API
New URL
Website
Short Name
Website URL
Shorten
File
Short Name
Website URL
Choose File No file chosen
Shorten

I haven't made the connection between creating a jenkins server and deploying code from a zip file that I created manually to upload to Elastic Beanstalk.

Pipeline Diagram


Clone forked repo
Created a git archive to a zip file
Created app using the uploaded zip file
http://
Tested the URL in the browser