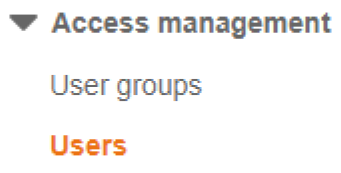
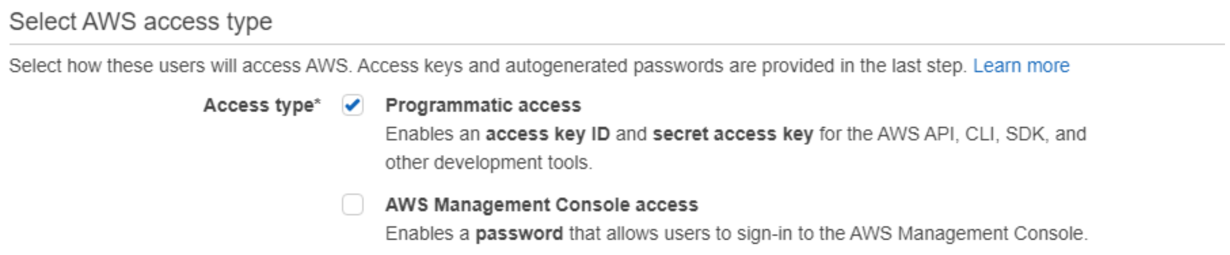
Deployment 4 - Documentation

First I started my EC2 Instance that has Jenkins

I then logged into my AWS account w/ Admin Privileges and **created an IAM user**.
I went to IAM aws service and selected users section



I then added a new user
I called it "**Jenkins-user**" and gave it **Programmatic access**

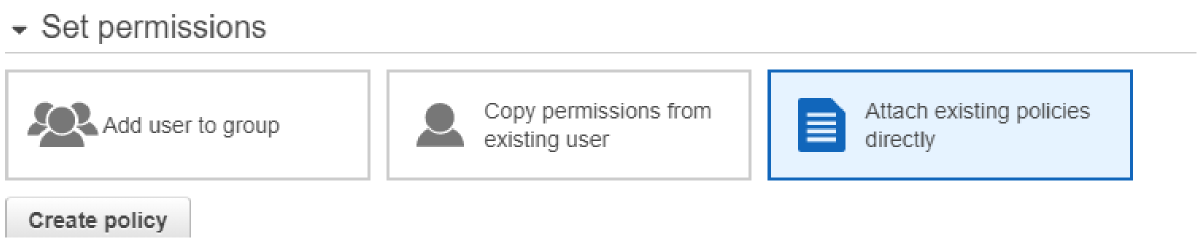

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

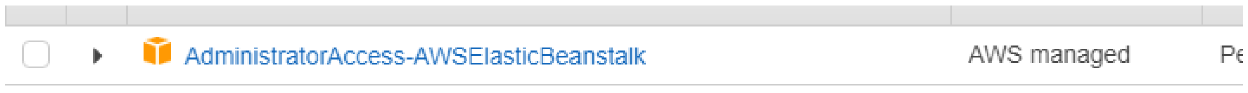
Access type* [✓] **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

[ ] **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

I then selected Attach existing policies directly

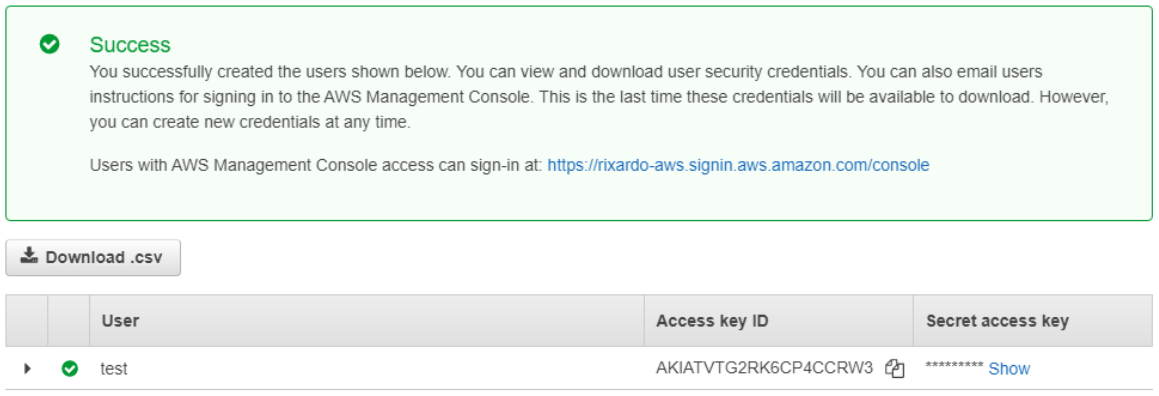

Set permissions
Add user to group
Copy permissions from existing user
Attach existing policies directly
Create policy

And Selected


[ ] AdministratorAccess-AWSElasticBeanstalk AWS managed Pe

Tags are not necessary

Once you create the user, **download the .csv** that has the user name, Access Key ID and secret access key



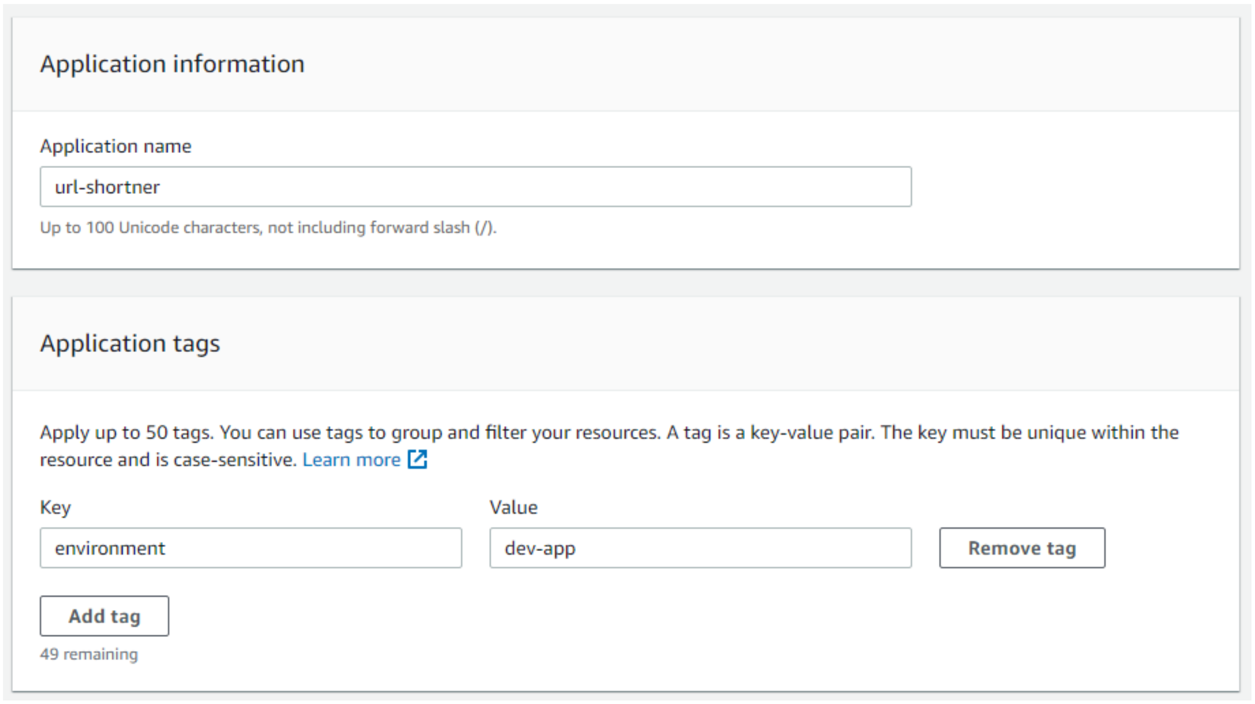
Success
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: https://rixardo-aws.signin.aws.amazon.com/console

Download .csv

		User	Access key ID	Secret access key
▸	✓	test	AKIATVTG2RK6CP4CCRW3	********* Show

Create an Elastic Beanstalk next



Application information

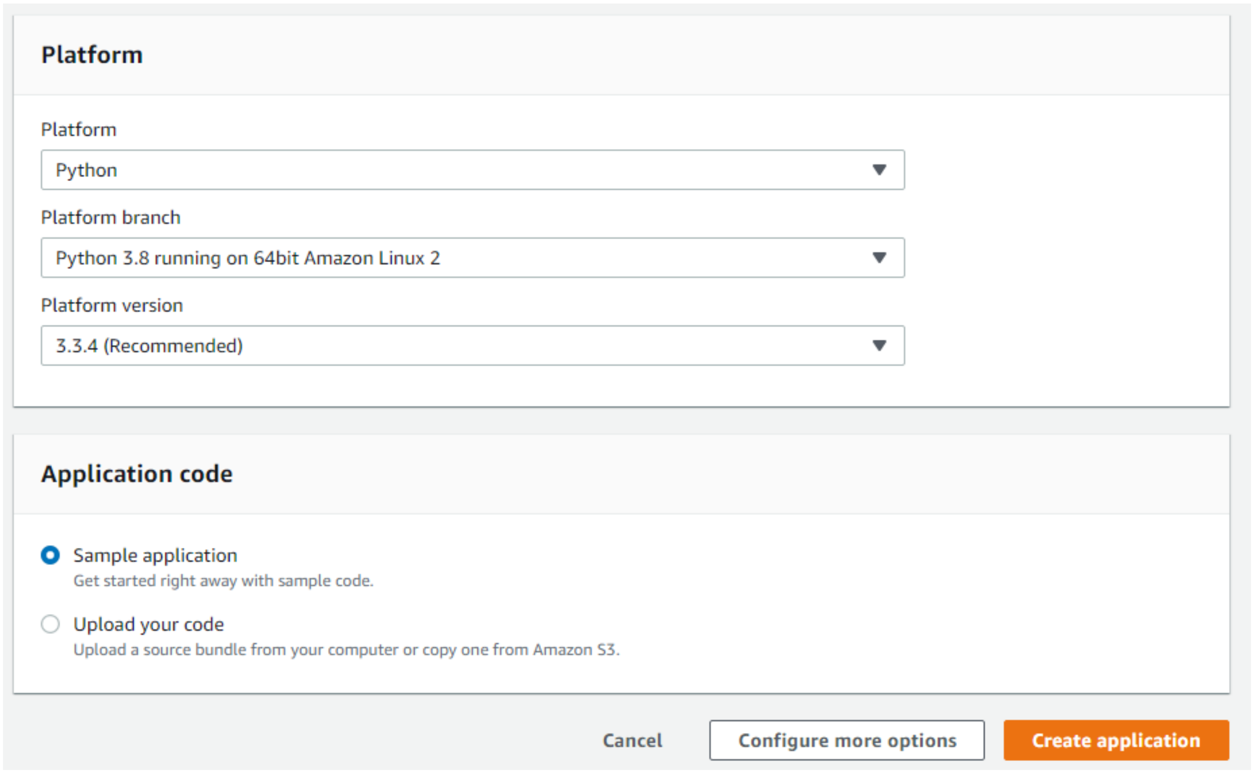
Application name
url-shortner
Up to 100 Unicode characters, not including forward slash (/).

Application tags

Apply up to 50 tags. You can use tags to group and filter your resources. A tag is a key-value pair. The key must be unique within the resource and is case-sensitive. Learn more

Key	Value	
environment	dev-app	Remove tag

Add tag
49 remaining



Platform

Platform: Python

Platform branch: Python 3.8 running on 64bit Amazon Linux 2

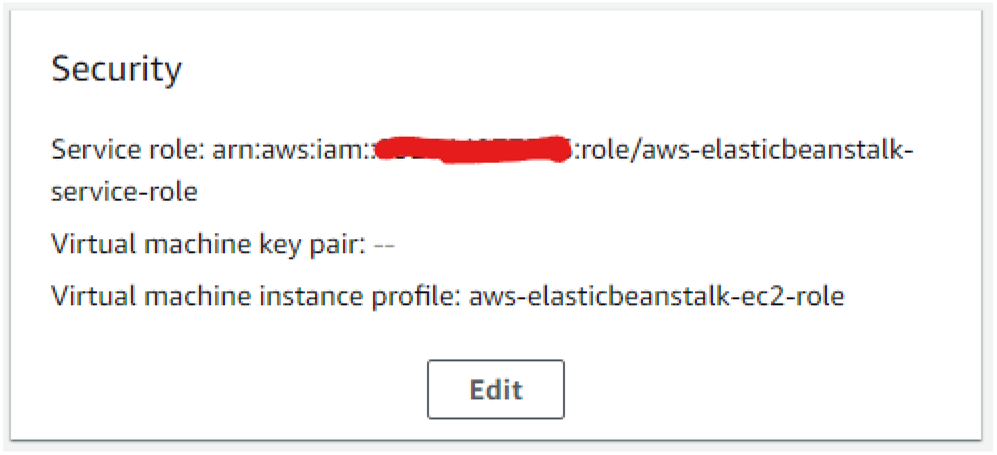
Platform version: 3.3.4 (Recommended)

Application code

(●) Sample application
Get started right away with sample code.

( ) Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.

OPTIONAL
Configure more options
Select configure more options and select Security. Add your Key pair to SSH into beanstalk EC2

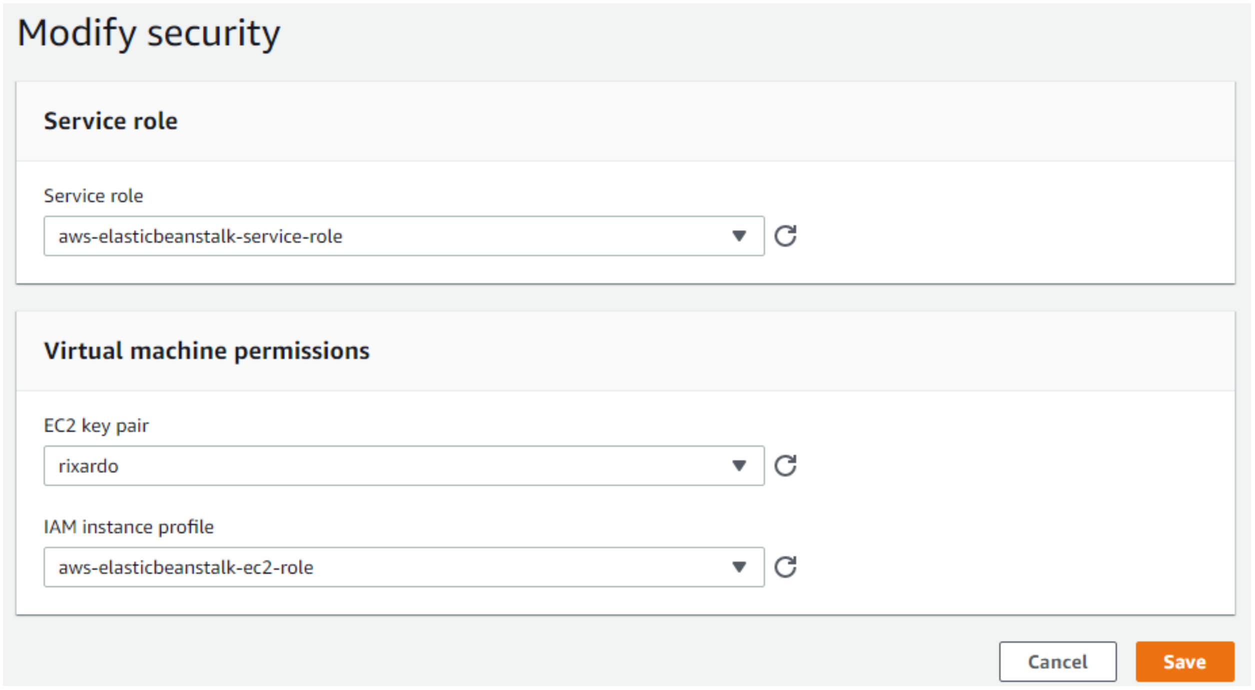


Security

Service role: arn:aws:iam::[redacted]:role/aws-elasticbeanstalk-service-role

Virtual machine key pair: --

Virtual machine instance profile: aws-elasticbeanstalk-ec2-role


Modify security

Service role

Service role: aws-elasticbeanstalk-service-role

Virtual machine permissions

EC2 key pair: rixardo

IAM instance profile: aws-elasticbeanstalk-ec2-role

Save and then press create APP

Go into Jenkins and manage Jenkins -> manage Plugins
Install the following plugins
AWSEB Deployment Plugin
CloudBees Credentials Plugin

I then created a Deployment 4 folder that has 3 separate folders inside with each name based on the application.

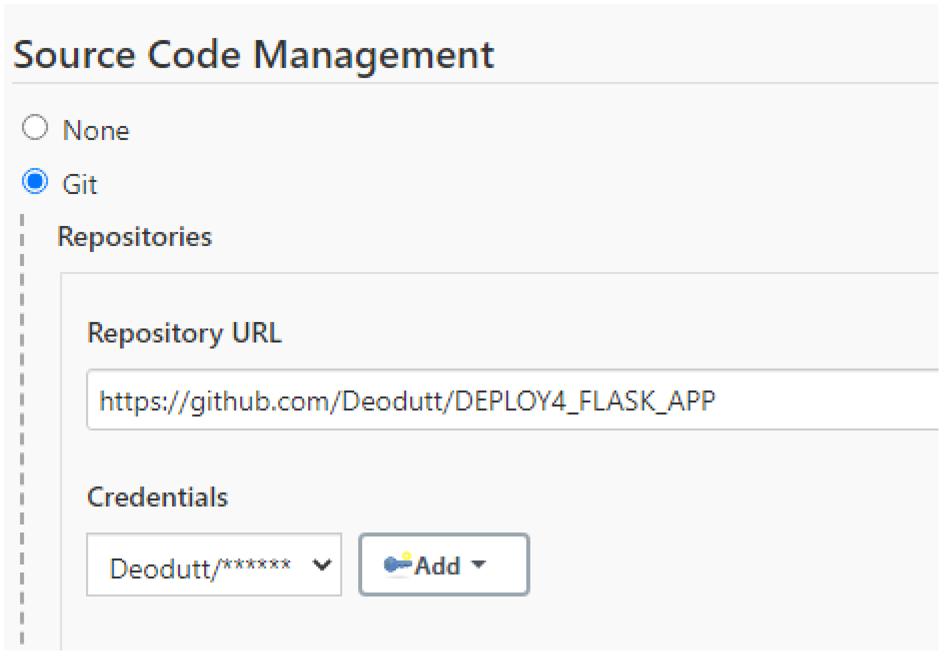
Useful Commands

```
python3 -m venv venv
pip freeze > requirements.txt
source ./Scripts/Activate

FLASK_APP=app.py flask run

Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Unrestricted
.\venv\Scripts\activate
$env:FLASK_ENV="development"
$env:FLASK_APP = "yourappname.py"
flask run
```

Now create a Freestyle Project inside the corresponding application folder.


Source Code Management

None

[✓ Git]

Repositories

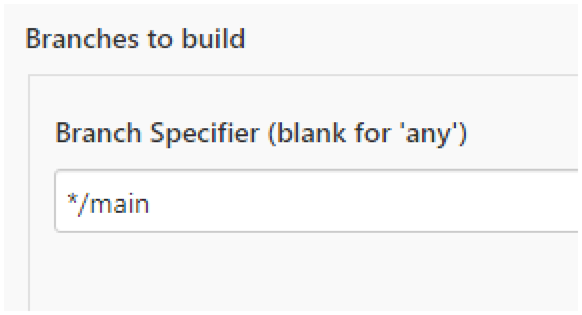
Repository URL

https://github.com/Deodutt/DEPLOY4_FLASK_APP

Credentials

Deodutt/******

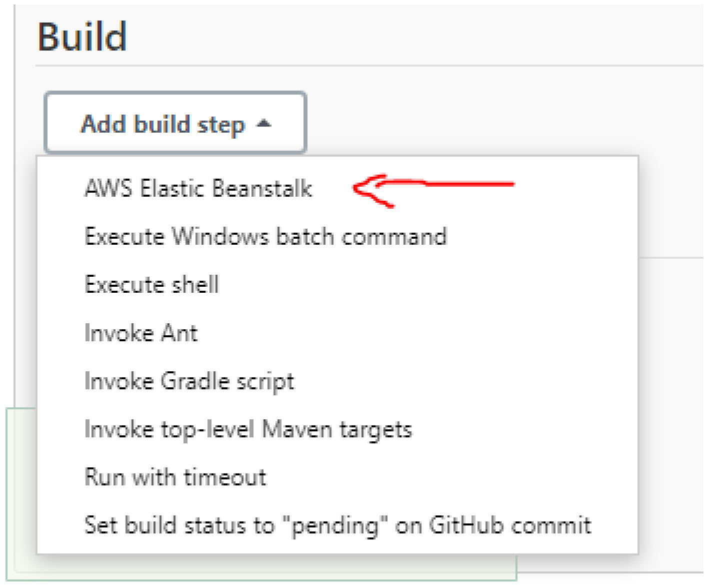
Make sure the branch is set correctly. (If its master, change it to master)


Branches to build

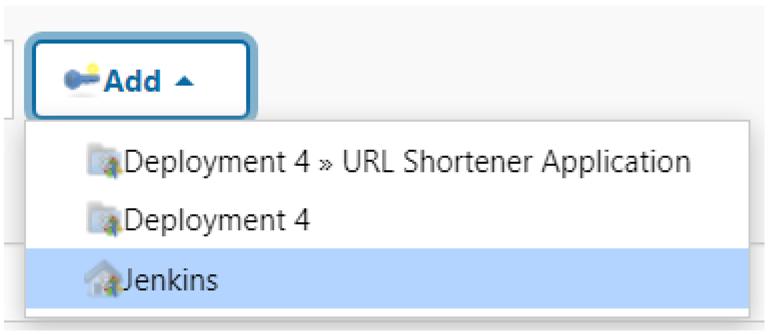
Branch Specifier (blank for 'any')

*/main

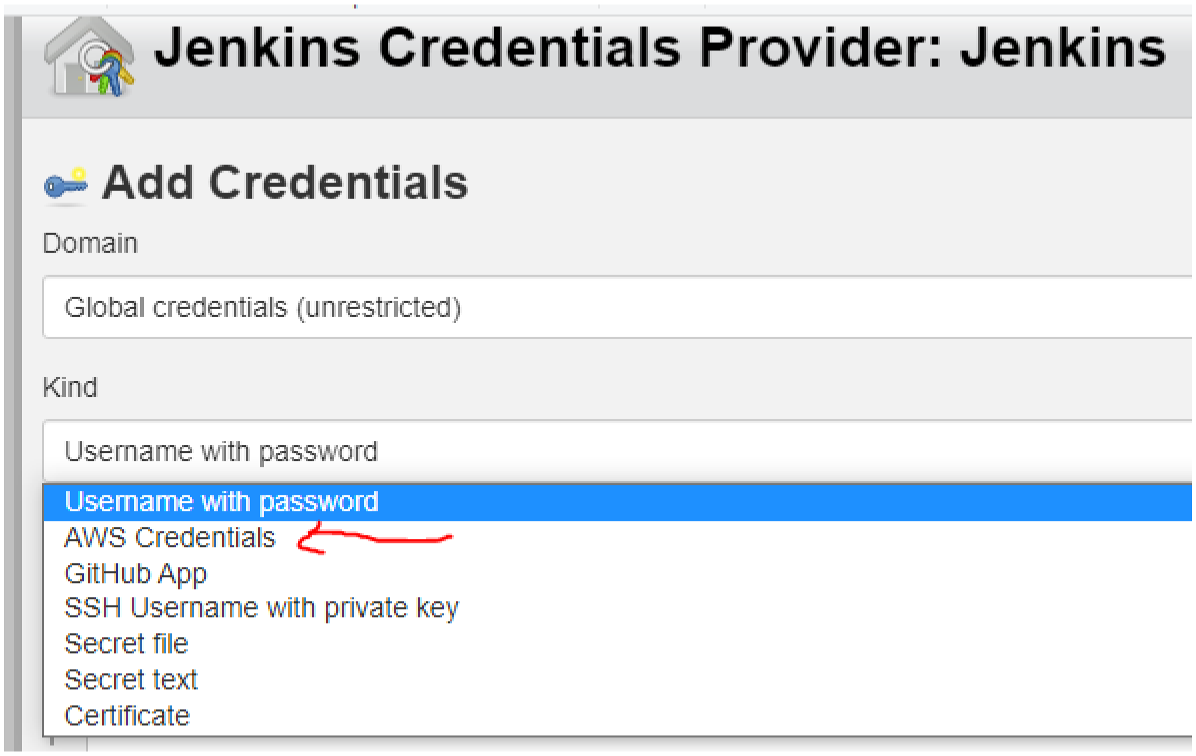
Scroll down to Build and select AWS Elastic Beanstalk


Build
Add build step
AWS Elastic Beanstalk
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Run with timeout
Set build status to "pending" on GitHub commit

To create a new credentials select Jenkins



Under kind, select AWS Credentials



Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

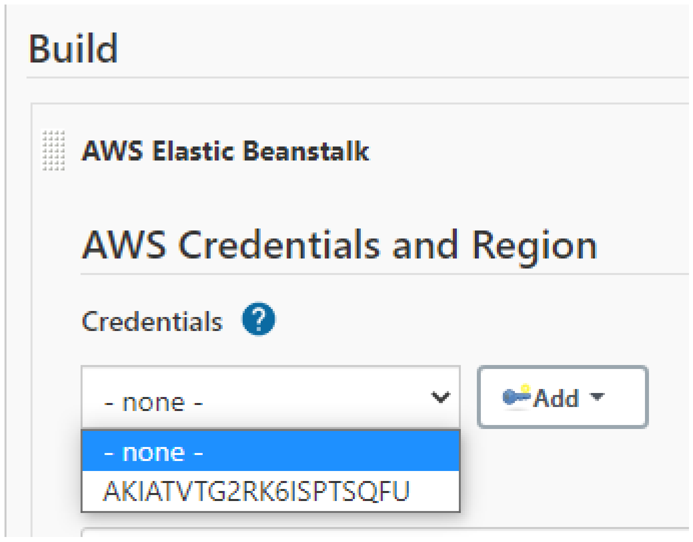
Username with password

Username with password
AWS Credentials
GitHub App
SSH Username with private key
Secret file
Secret text
Certificate

For ID just put a "**Jenkins-user**"

For **Access KEY ID** and **Secret Access Key** - Use the credentials from the IAM user you downloaded.

Press Add and now select the credentials from the dropdown



Build

AWS Elastic Beanstalk

AWS Credentials and Region

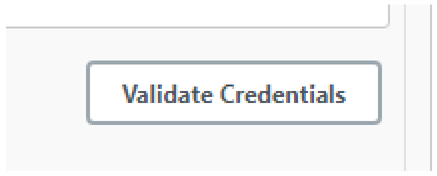
Credentials

- none -

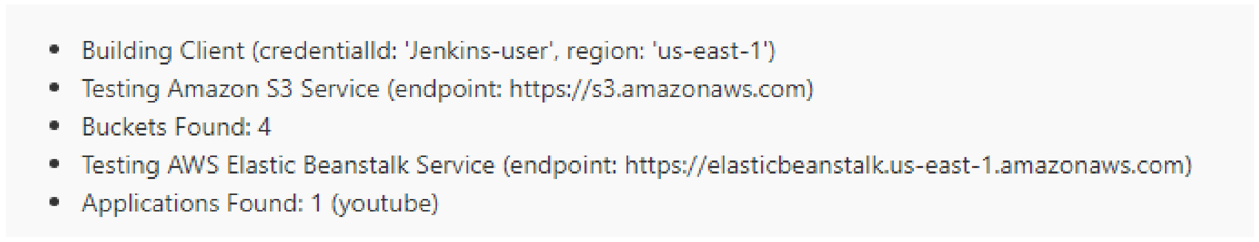
Add

- none -
AKIATVTG2RK6ISPTSQFU

Select the AWS Region where the Elastic BeanStalk was created
Then press **Validate Credentials** in the bottom right

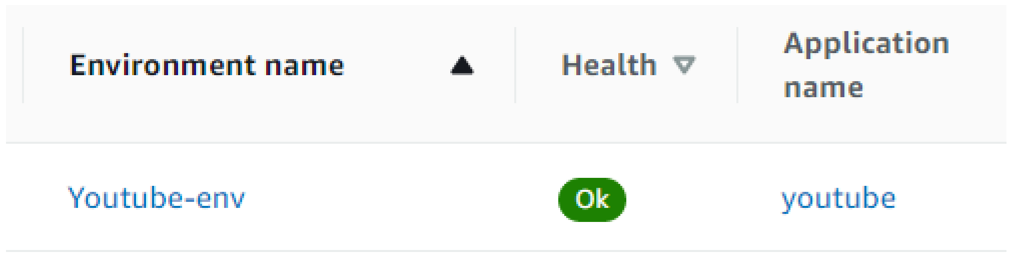


You should see a response like this…



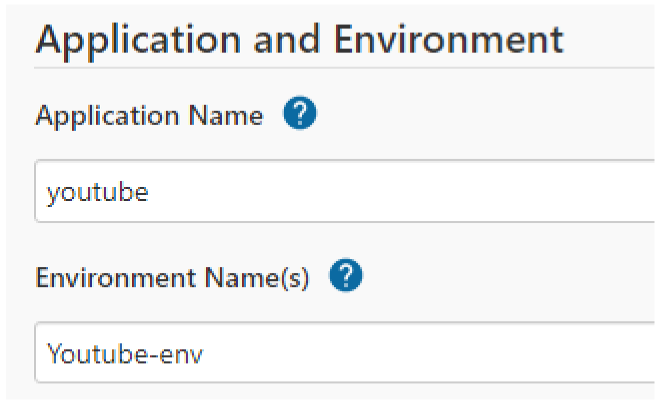
```
• Building Client (credentialId: 'Jenkins-user', region: 'us-east-1')
• Testing Amazon S3 Service (endpoint: https://s3.amazonaws.com)
• Buckets Found: 4
• Testing AWS Elastic Beanstalk Service (endpoint: https://elasticbeanstalk.us-east-1.amazonaws.com)
• Applications Found: 1 (youtube)
```

Go back to AWS Beanstalk and find the environment you created. Make a note of the **Environment name** and **application name**



Environment name ▲	Health ▽	Application name
Youtube-env	Ok	youtube

Scroll down to Application and Environment and enter the correct info from AWSEB,

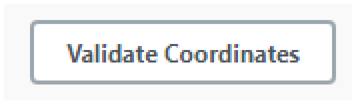


Application and Environment

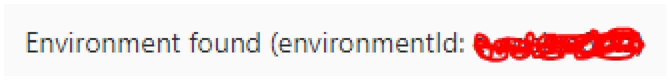
Application Name: youtube

Environment Name(s): Youtube-env

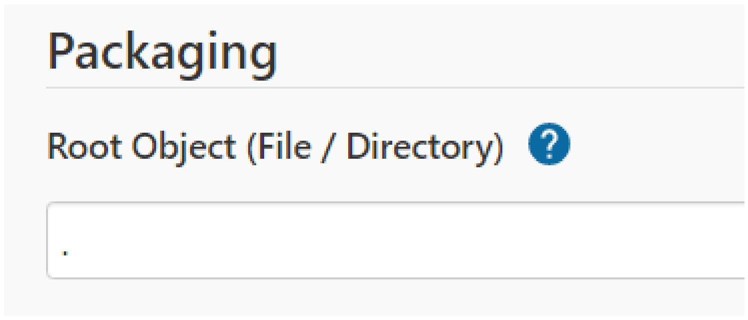
Then press validate coordinates



You should see something like this..

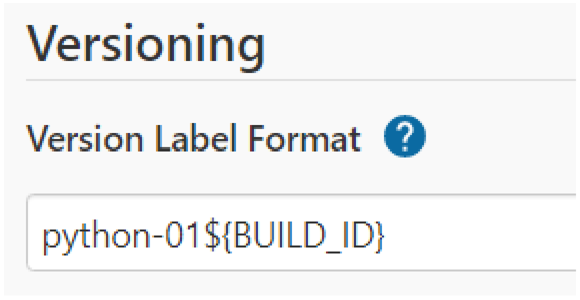

Environment found (environmentId: [redacted]

In Packaging, add a "." to the Root object



.

In versioning put **python-01${BUILD_ID}**



python-01${BUILD_ID}

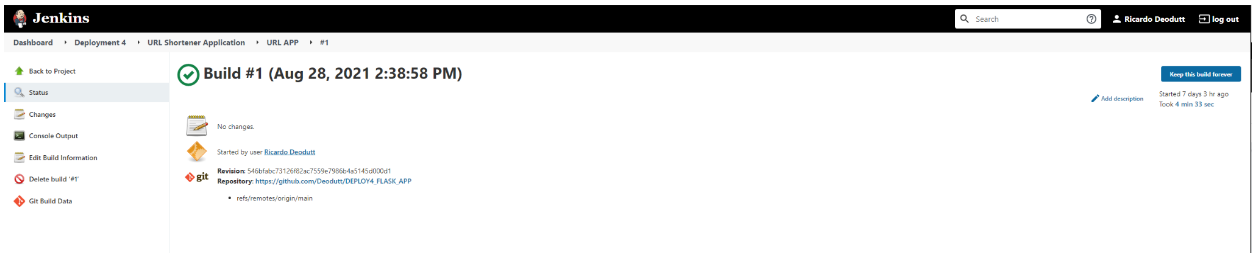
Then press SAVE and **BUILD NOW**

Should take about ~5 minutes to build successfully

Once you have a successful build, Go back to AWS Elastic Bean and click on the **URL** for your environment.

URL Shortener
Souce: https://github.com/Deodutt/DEPLOY4_FLASK_APP

Successful Build:



Build #1 (Aug 28, 2021 2:38:58 PM)

Started 7 days 3 hr ago
Took 4 min 33 sec

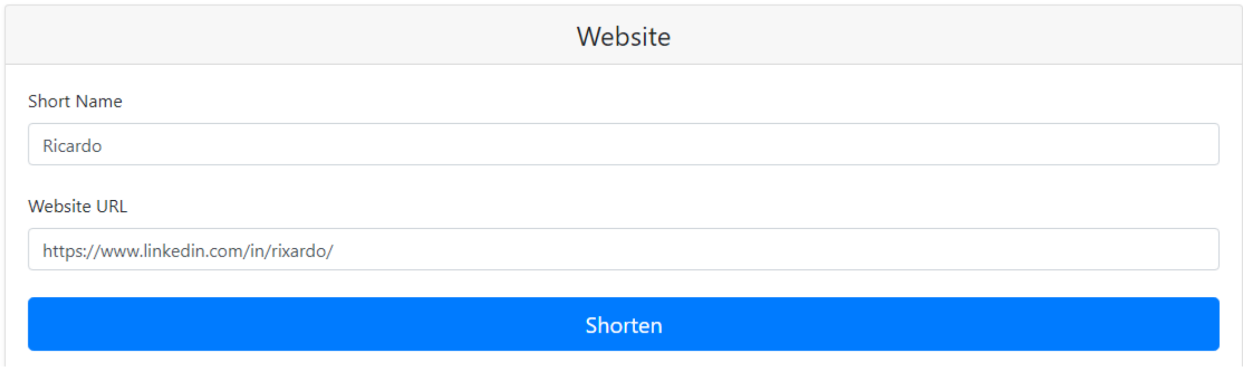
No changes.

Started by user Ricardo Deodutt

Revision: 546bfabc73126f82ac7559e7986b4a5145d000d1
Repository: https://github.com/Deodutt/DEPLOY4_FLASK_APP

- refs/remotes/origin/main

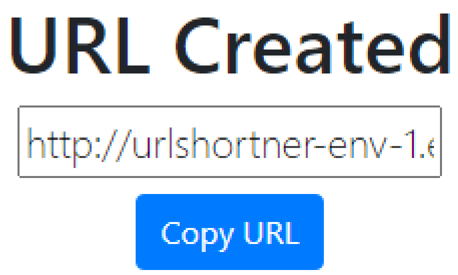
Default Web Page:



Short Name: Ricardo

Website URL: https://www.linkedin.com/in/rixardo/

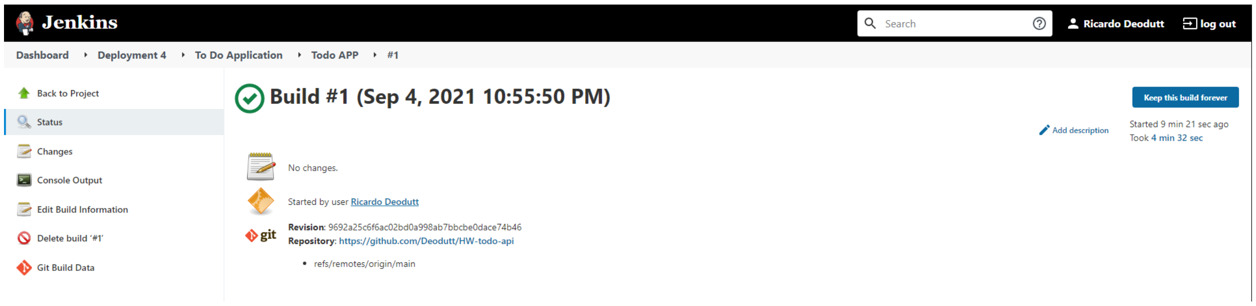
Web page once I enter information and press shorten



This takes me to my Linkedin Page

Todo
Source: https://github.com/Deodutt/HW-todo-api

Successful Build:



Build #1 (Sep 4, 2021 10:55:50 PM)

Started 9 min 21 sec ago
Took 4 min 32 sec

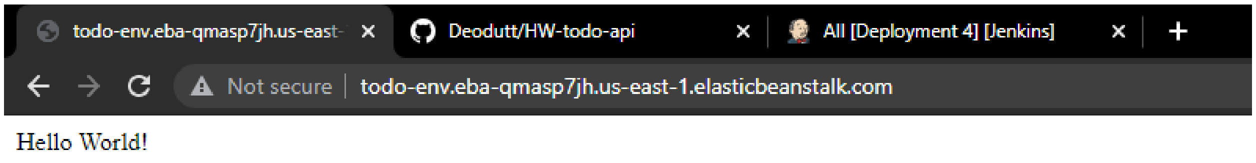
No changes.

Started by user Ricardo Deodutt

Revision: 9692a25c6f6ac02bd0a998ab7bbcbe0dace74b46
Repository: https://github.com/Deodutt/HW-todo-api

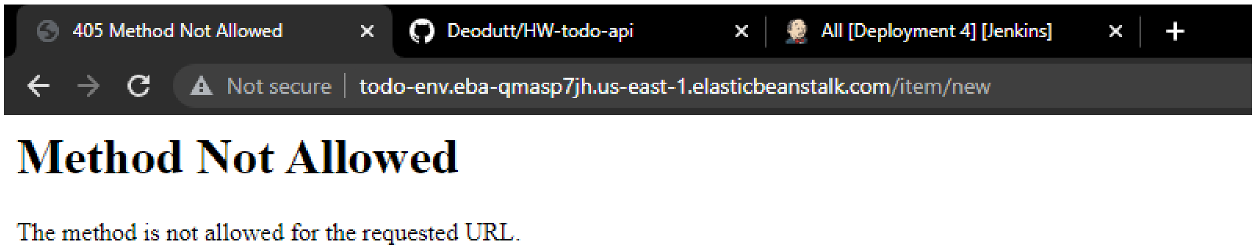
- refs/remotes/origin/main

Route: /


todo-env.eba-qmasp7jh.us-east-
Deodutt/HW-todo-api
All [Deployment 4] [Jenkins]
Not secure
todo-env.eba-qmasp7jh.us-east-1.elasticbeanstalk.com

Hello World!

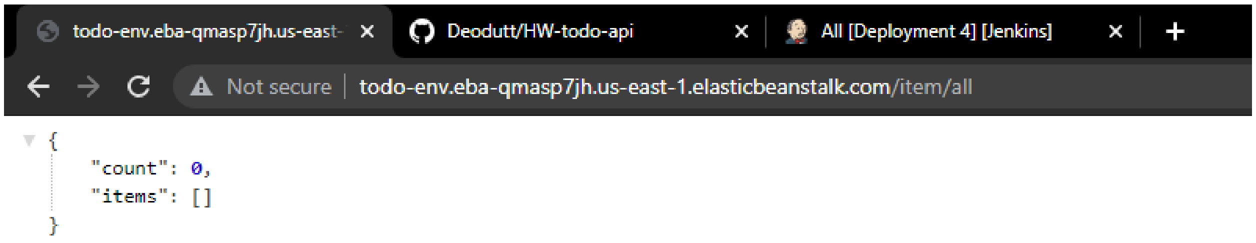
Route: /item/new


405 Method Not Allowed
Deodutt/HW-todo-api
All [Deployment 4] [Jenkins]
Not secure
todo-env.eba-qmasp7jh.us-east-1.elasticbeanstalk.com/item/new

Method Not Allowed

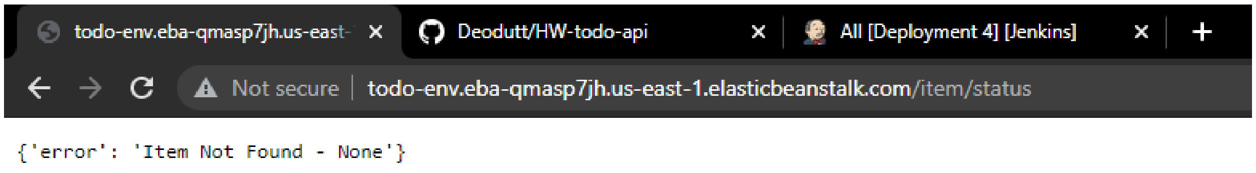
The method is not allowed for the requested URL.

Route: /item/all


todo-env.eba-qmasp7jh.us-east-
Deodutt/HW-todo-api
All [Deployment 4] [Jenkins]
Not secure
todo-env.eba-qmasp7jh.us-east-1.elasticbeanstalk.com/item/all

```
{
    "count": 0,
    "items": []
}
```

Route: /item/status

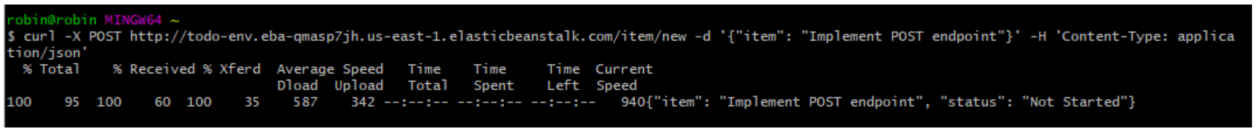


todo-env.eba-qmasp7jh.us-east-1.elasticbeanstalk.com/item/status

```
{'error': 'Item Not Found - None'}
```

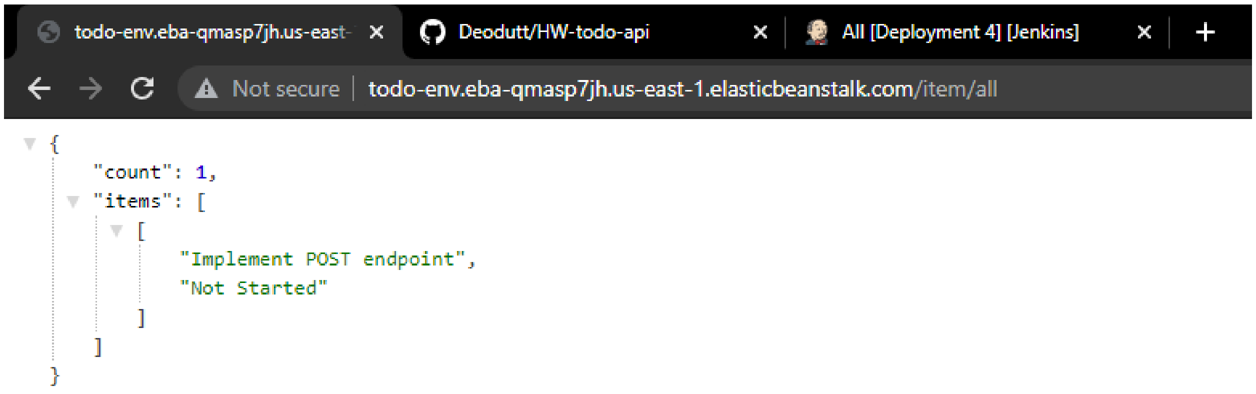
To update the database run the following commands...

```
curl -X POST
http://todo-env.eba-qmasp7jh.us-east-1.elasticbeanstalk.com/item/new -d
'{"item": "Implement POST endpoint"}' -H 'Content-Type: application/json'
```



```
robin@robin MINGW64 ~
$ curl -X POST http://todo-env.eba-qmasp7jh.us-east-1.elasticbeanstalk.com/item/new -d '{"item": "Implement POST endpoint"}' -H 'Content-Type: applica
tion/json'
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    95  100    60  100    35    587    342 --:--:-- --:--:-- --:--:--   940{"item": "Implement POST endpoint", "status": "Not Started"}
```

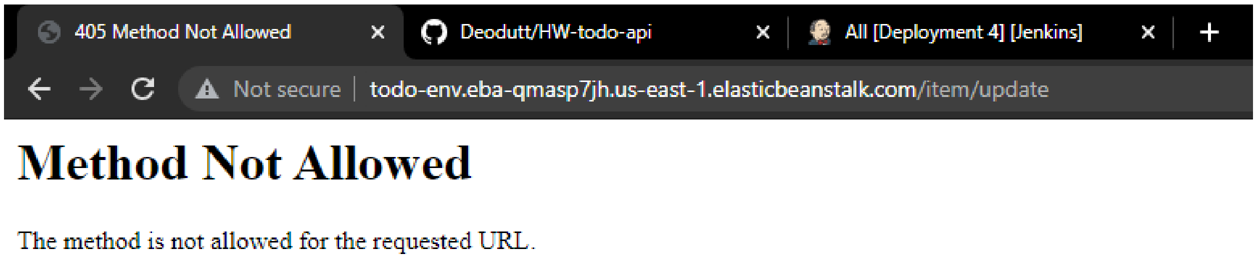
Route: /item/all



todo-env.eba-qmasp7jh.us-east-1.elasticbeanstalk.com/item/all

```
{
    "count": 1,
    "items": [
        [
            "Implement POST endpoint",
            "Not Started"
        ]
    ]
}
```

Route: /item/update

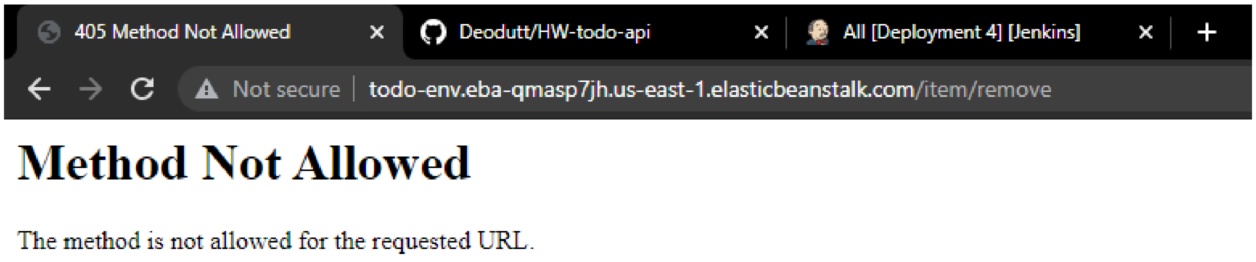


todo-env.eba-qmasp7jh.us-east-1.elasticbeanstalk.com/item/update

Method Not Allowed

The method is not allowed for the requested URL.

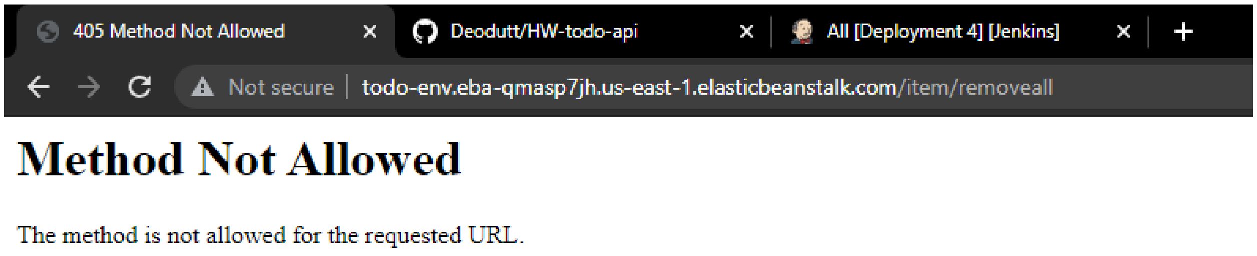
Route: /item/remove


405 Method Not Allowed
Deodutt/HW-todo-api
All [Deployment 4] [Jenkins]
Not secure | todo-env.eba-qmasp7jh.us-east-1.elasticbeanstalk.com/item/remove

Method Not Allowed

The method is not allowed for the requested URL.

Route: /item/removeall


405 Method Not Allowed
Deodutt/HW-todo-api
All [Deployment 4] [Jenkins]
Not secure | todo-env.eba-qmasp7jh.us-east-1.elasticbeanstalk.com/item/removeall

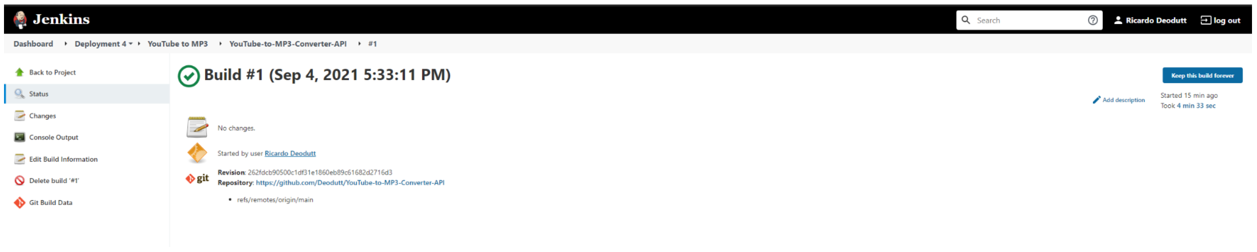
Method Not Allowed

The method is not allowed for the requested URL.

YouTube

Source: https://github.com/Deodutt/YouTube-to-MP3-Converter-API

Successful Build



Build #1 (Sep 4, 2021 5:33:11 PM)

Started 15 min ago
Took 4 min 33 sec

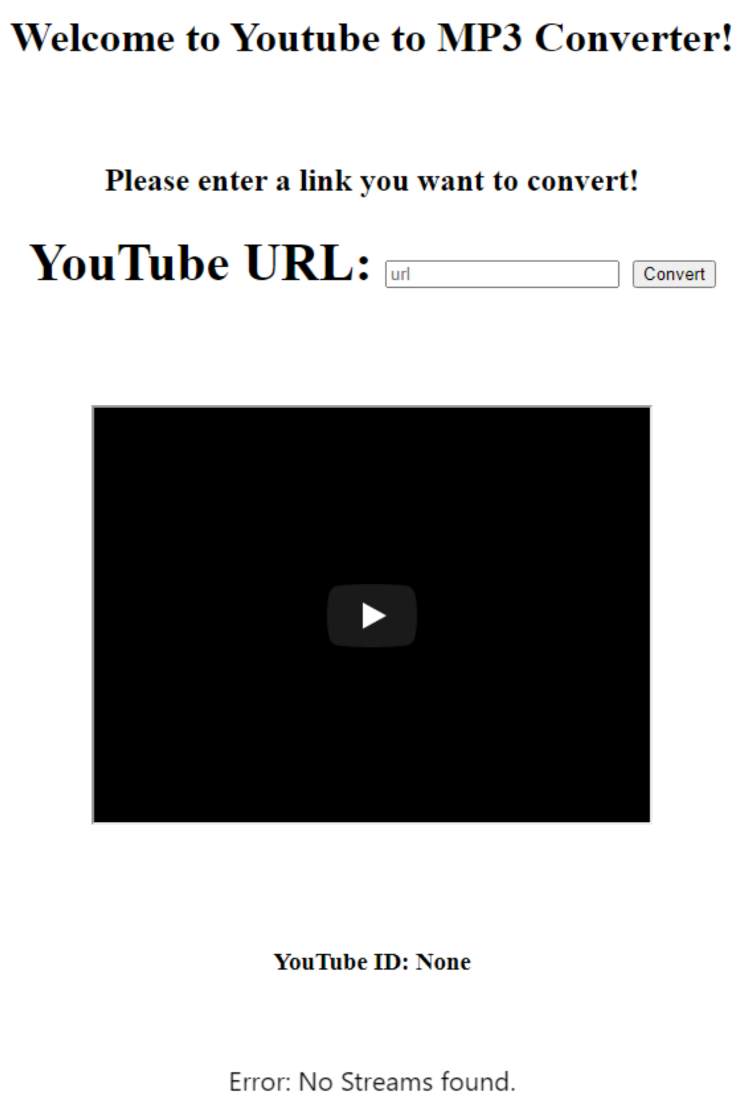
No changes.

Started by user Ricardo Deodutt

Revision: 262fdcb90500c1df31e1860eb89c61682d2716d3
Repository: https://github.com/Deodutt/YouTube-to-MP3-Converter-API

- refs/remotes/origin/main

Default web page when I click on the URL from Elastic Beanstalk



Welcome to Youtube to MP3 Converter!

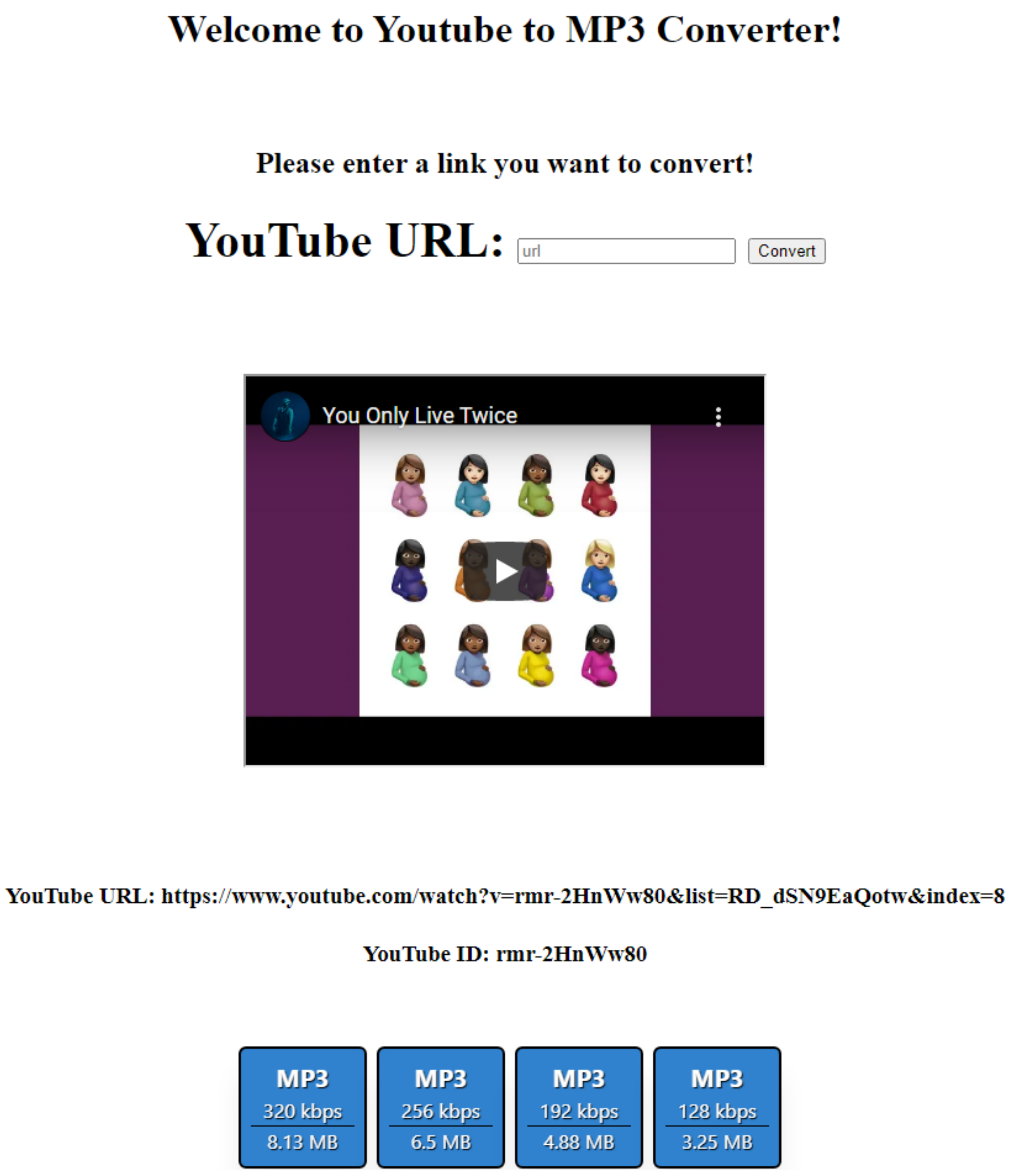
Please enter a link you want to convert!

YouTube URL:

YouTube ID: None

Error: No Streams found.

Web page once I enter an URL and press convert.



Welcome to Youtube to MP3 Converter!

Please enter a link you want to convert!

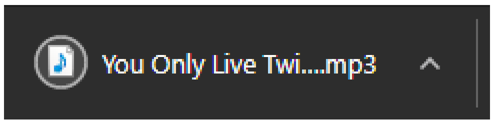
YouTube URL: url Convert

YouTube URL: https://www.youtube.com/watch?v=rmr-2HnWw80&list=RD_dSN9EaQotw&index=8

YouTube ID: rmr-2HnWw80

MP3	MP3	MP3	MP3
320 kbps	256 kbps	192 kbps	128 kbps
8.13 MB	6.5 MB	4.88 MB	3.25 MB

Successful download after selecting a download option



You Only Live Twi....mp3

YouTube Converter

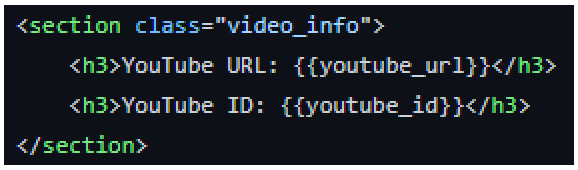
(This is for educational purposes)

As a person who listens to music a lot, I wanted to make this application to make my life a little easier. I prefer to work on projects that benefit me. For example, I made a VPN application so I can use it. I prefer to download music online and put it on my phone rather than using streaming services. I do not like paywalls so I thought this was a great project for me. It's a weird way of listening to music but I have been doing it for a long time since Limewire days.

While looking up ideas, I faced some issues.I tried looking up official YouTube APIs to download videos but I found out it's not supported because it's against their terms of services so I had to look for different options. I came across this repository and thought it would be great to add on. The repo creator supplies an API that I could use to embed into a website. It gives the user different options to download videos in different qualities. While creating the application, I used Ibrahima's repository as a template. His repository was a great way to understand how to write to HTML files and use variables in them. Everything was simple and easy to understand. While creating the application, I found this really great way of getting the YouTube ID and thought it would be great to incorporate it.

For my application, once a user enters the URL inside the web page and presses convert, it would do a POST method. This would then call the converter() function inside of application.py and that would get the string the user entered. The application will then call a id_grabber function from the helper.py file, and that will use urlparse to parse the URL and extract the youtube ID from the URL. It goes to many different cases to incase the user enters a different youtube link format. Once it gets the ID it will return the ID in a variable called youtube_id. The youtube_id will then concat with a string to the API and be assigned to a variable called api_converter_link. This is then returned into a function called render_templates() which basically renders a template from the template folder which has an index.html file. This will basically let the HTML page use variables.

example

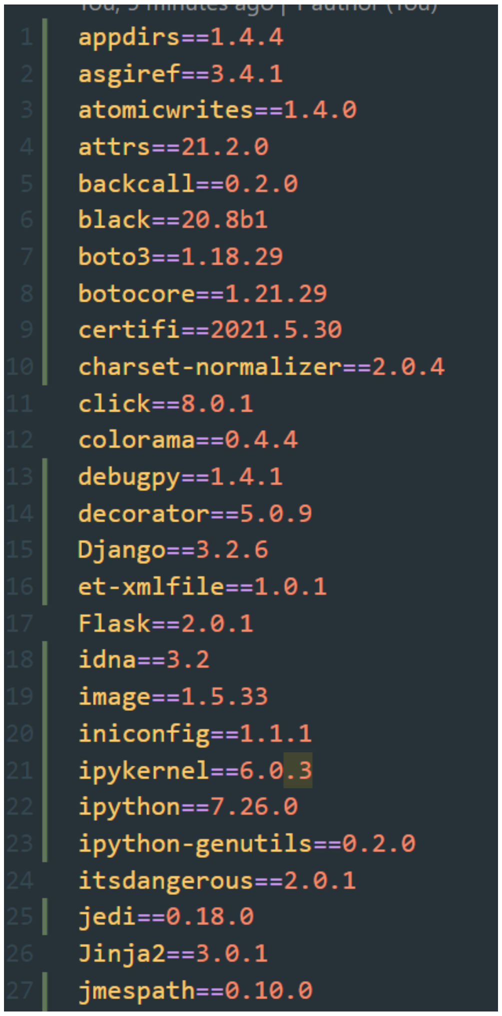


```
<section class="video_info">
    <h3>YouTube URL: {{youtube_url}}</h3>
    <h3>YouTube ID: {{youtube_id}}</h3>
</section>
```

Errors

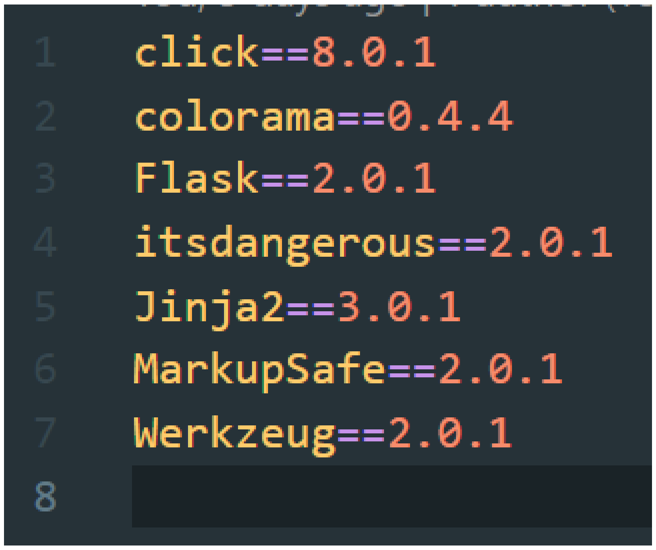
When I was inside my virtual environment, I kept trying to make the requirement.txt file but it kept bugging out. I didn't notice it at all because I assumed it would make the pip list correctly made. I noticed my elastic beanstalk kept degrading and the build would take long in jenkins. I made sure all my settings in jenkins and file formats (such as application.py and requirements.txt) were correct.

I then eventually checked my requirement.txt file and noticed a long list that was different from the other applications for deployment4.



```
appdirs==1.4.4
asgiref==3.4.1
atomicwrites==1.4.0
attrs==21.2.0
backcall==0.2.0
black==20.8b1
boto3==1.18.29
botocore==1.21.29
certifi==2021.5.30
charset-normalizer==2.0.4
click==8.0.1
colorama==0.4.4
debugpy==1.4.1
decorator==5.0.9
Django==3.2.6
et-xmlfile==1.0.1
Flask==2.0.1
idna==3.2
image==1.5.33
iniconfig==1.1.1
ipykernel==6.0.3
ipython==7.26.0
ipython-genutils==0.2.0
itsdangerous==2.0.1
jedi==0.18.0
Jinja2==3.0.1
jmespath==0.10.0
```

So I had to make sure the requirement.txt file had only the required dependencies in it.



```
click==8.0.1
colorama==0.4.4
Flask==2.0.1
itsdangerous==2.0.1
Jinja2==3.0.1
MarkupSafe==2.0.1
Werkzeug==2.0.1
```

Also I had basic errors such as my requirements.txt was missing the s.

I also had a problem trying to link my stylesheet to my html file. It took me a while to figure it out but I found out you need a static folder with the stylesheet inside of it.

Plans

I plan to eventually update this application by finding different ways to download videos than depending on an API that I cannot control. Along with that I plan to expand this project by trying out MySQL and trying Node.js in the future. Overall, I enjoy this project because I will be using this application for my own use.

URL Shortener Application: https://github.com/Deodutt/DEPLOY4_FLASK_APP
To Do Application: https://github.com/Deodutt/HW-todo-api
YouTube Converter Application: https://github.com/Deodutt/YouTube-to-MP3-Converter-API