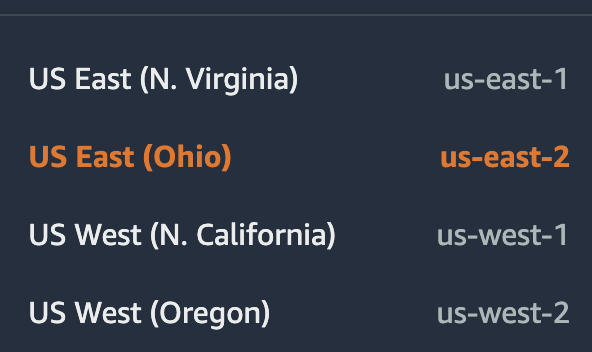
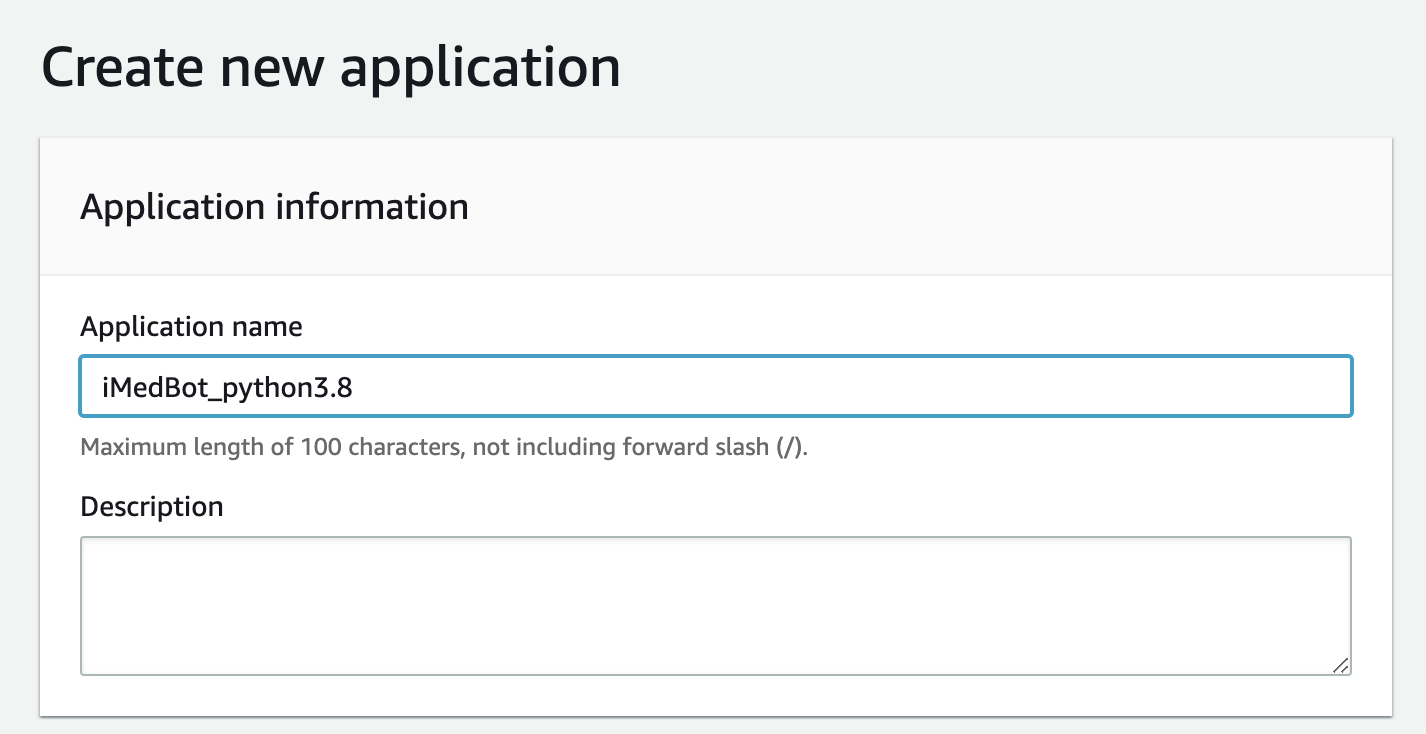
Procedure to Setup Website using AWS.

1. Choose the aws area



1. Create an application in Elastic Beanstalk
2. Go to application. Question: What is the difference between application and environment? Answer: Application can have multiple environments.



Tag is not necessary.

1. Once the application is created, click on the “Create a new enironment”.

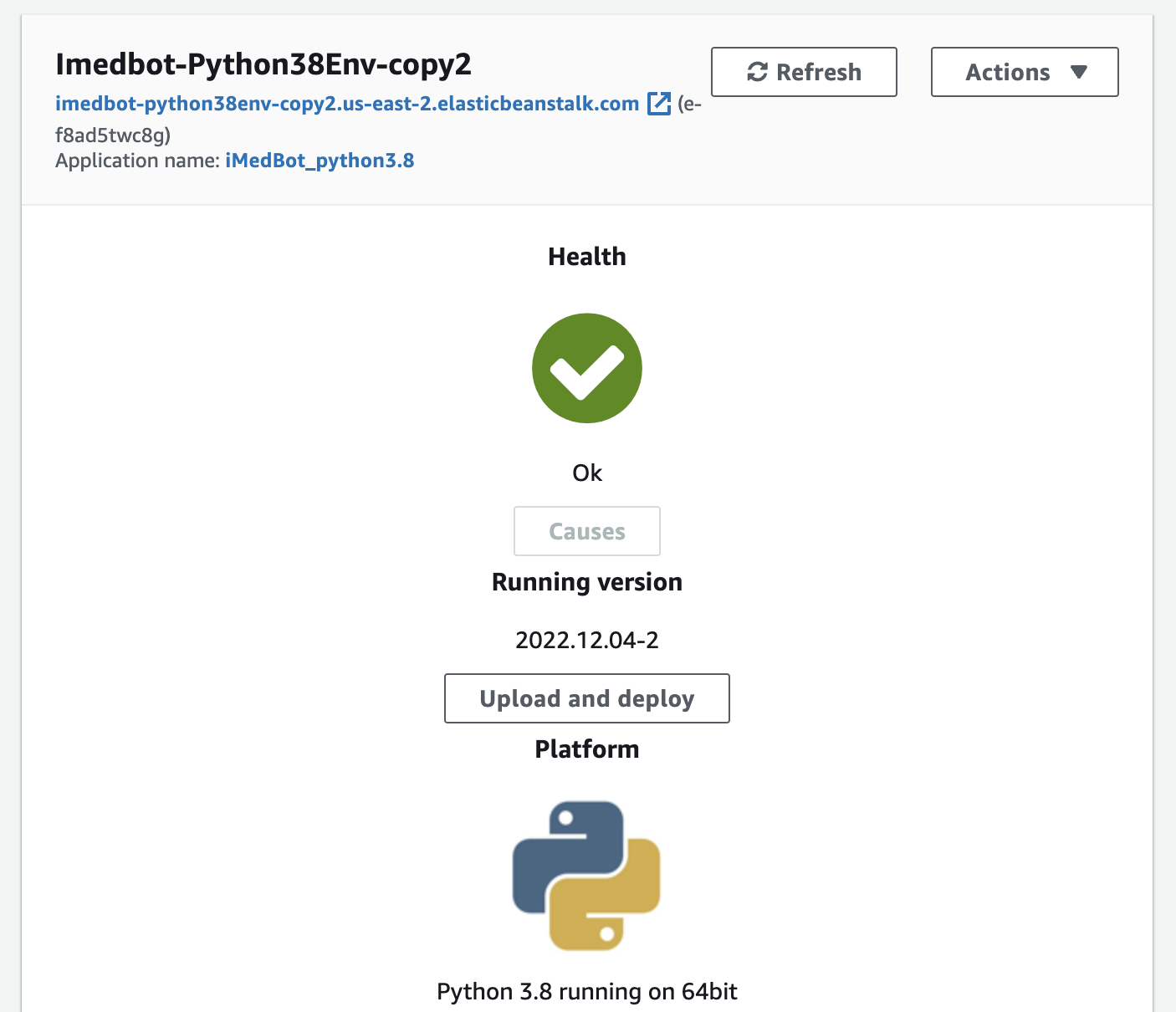
It will create an environment automatically for this application, which includes an ec2 instance named as application name + “env”

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated



1. Go to configuration (in the environment, edit security) to add the key pair to the environment

Key pair

Graphical user interface, application

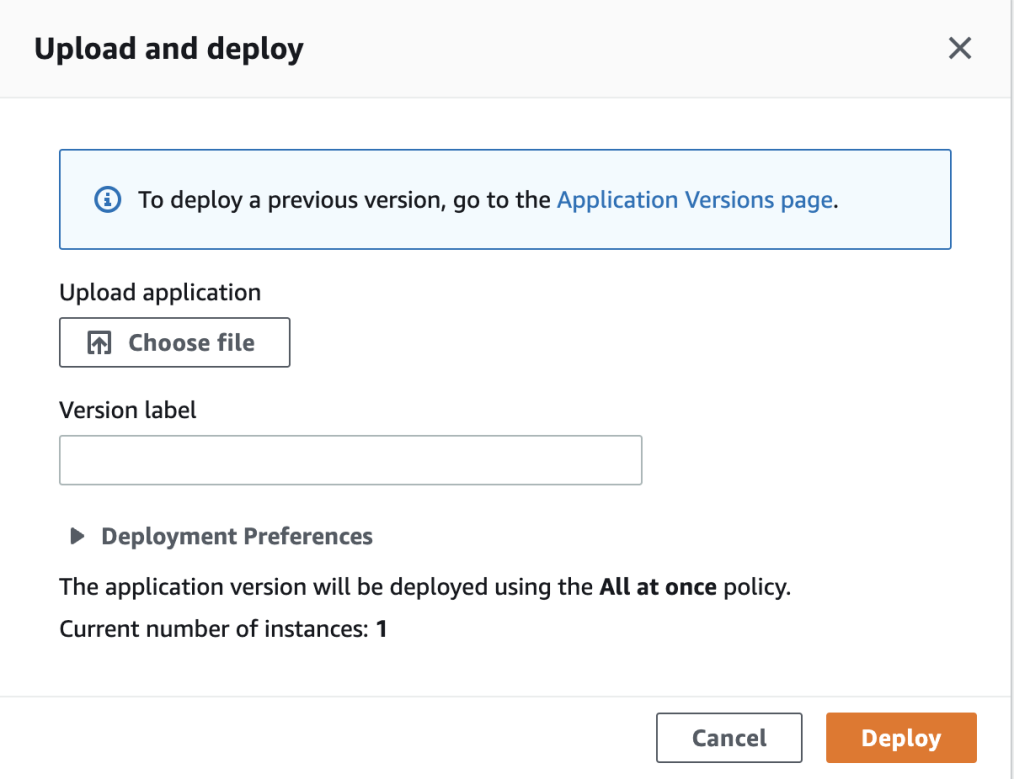
Description automatically generated

Graphical user interface, text, application, email

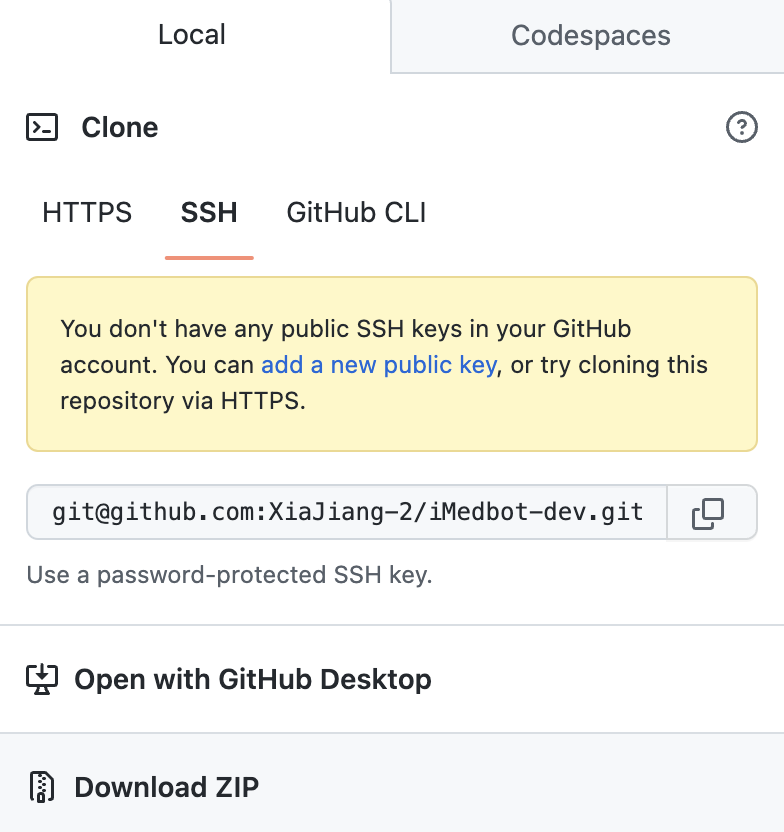
Description automatically generated

1. Upload code and deploy application in Elastic Beanstalk

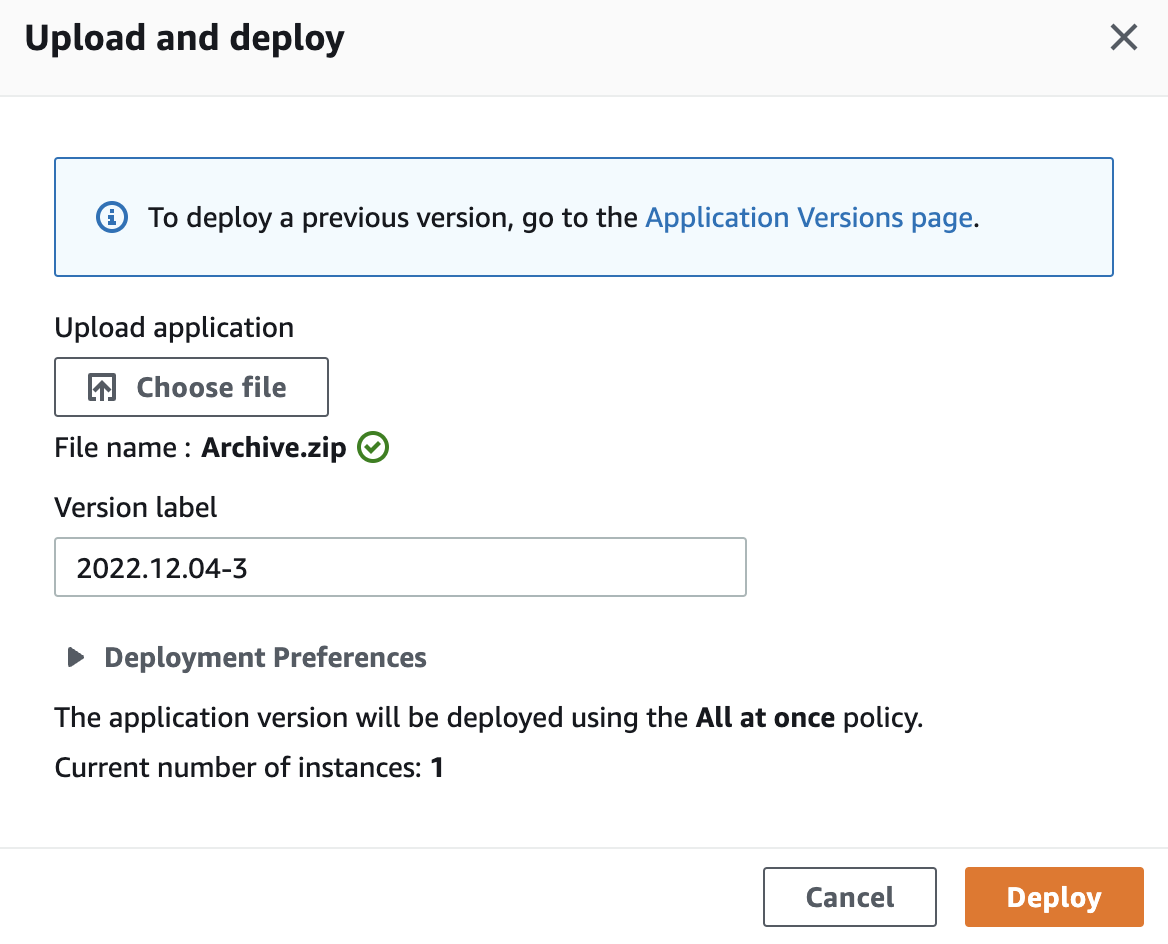
A.click upload and deploy in application



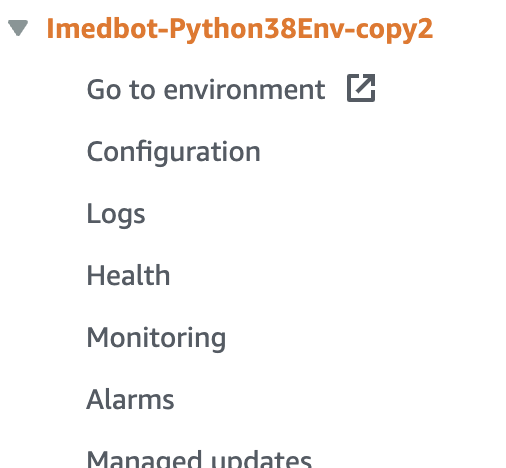
B.download the project from github(download zip)

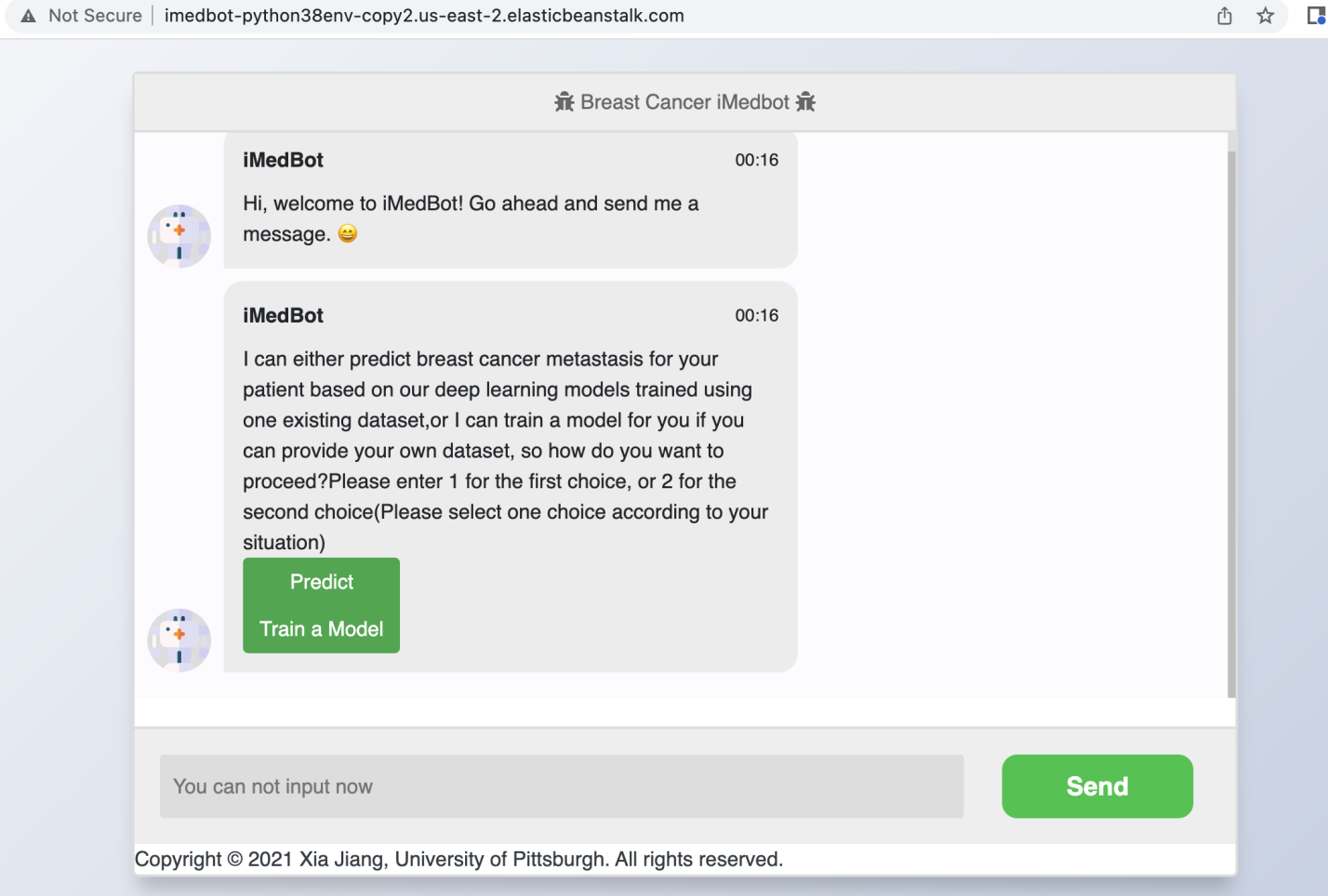


C.after downloading the whole project, unzip it and compress all the files under imedbot-dev directory(sometimes after unzip it extracts a folder which can not be directly uploaded to aws)

D. Edit the version label, the label should be unique and cannot be the same as previous version label of this application. 

E.the application can be checked in Elastic Beanstalk in “go to environment”

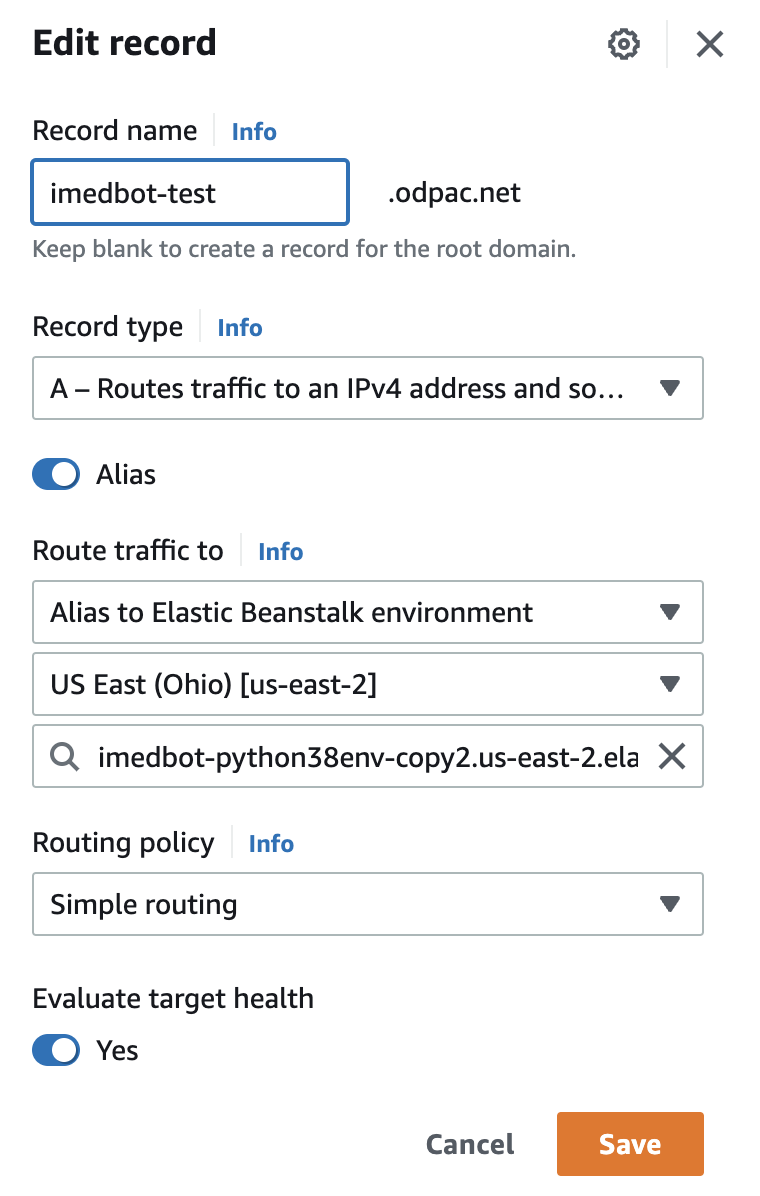




This url is a beanstalk url and it can be added to route53 later.

1. Connect the instance to domain in route53

Locate the hosted zones, then create record or edit existing record (in which case, you need to select the new environment).



Link the record to application in Elastic Beanstalk

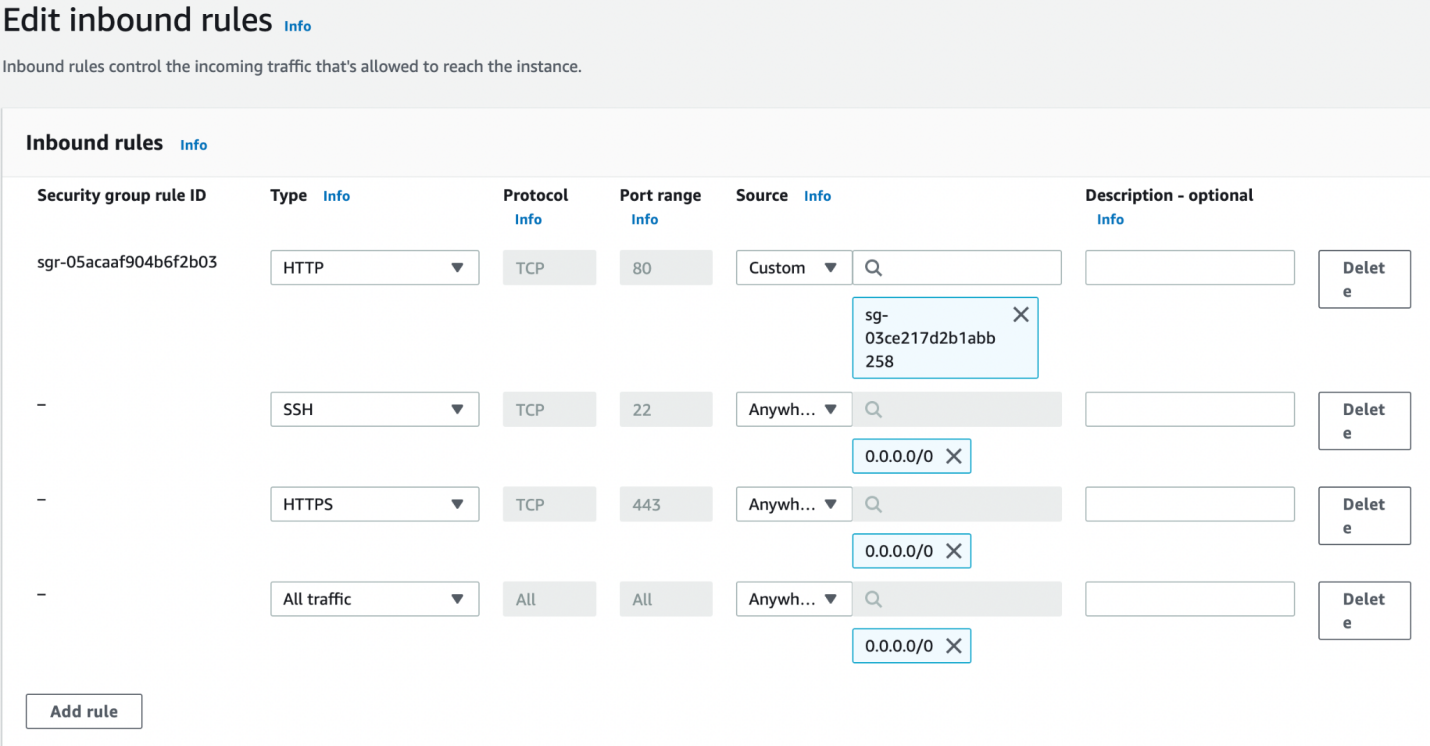
# Use ssh to open server and find out application

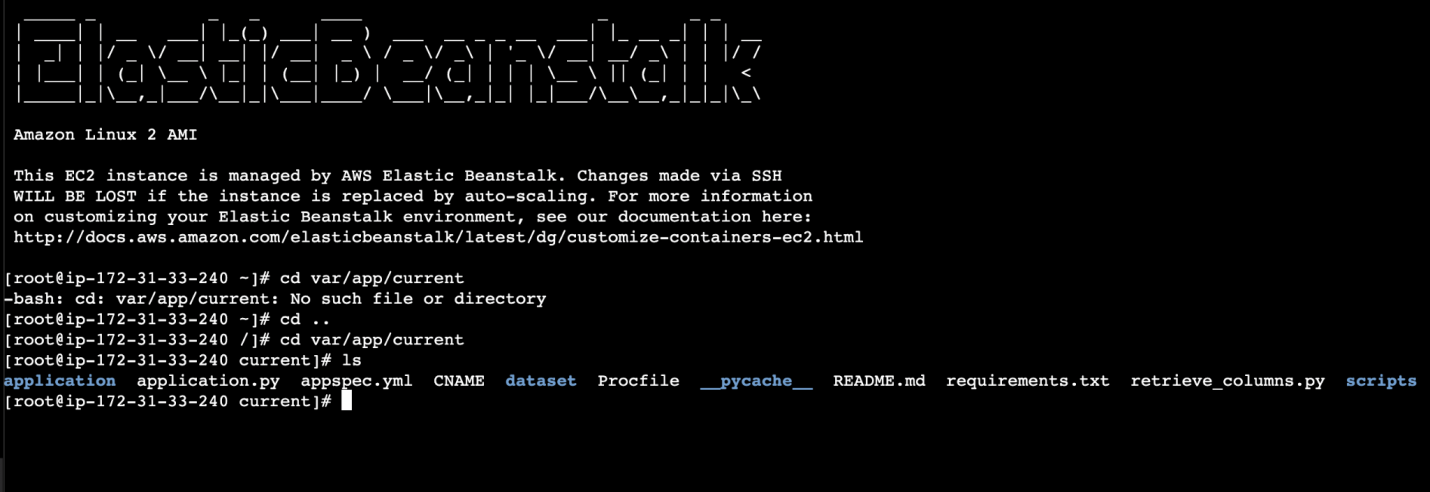
Go to EC2,

Find the instance for the current application,

Click the instance ID from the list, then click Connect.

Click the connect in the new window (See the second screenshot below) will lead to the ssh shell.

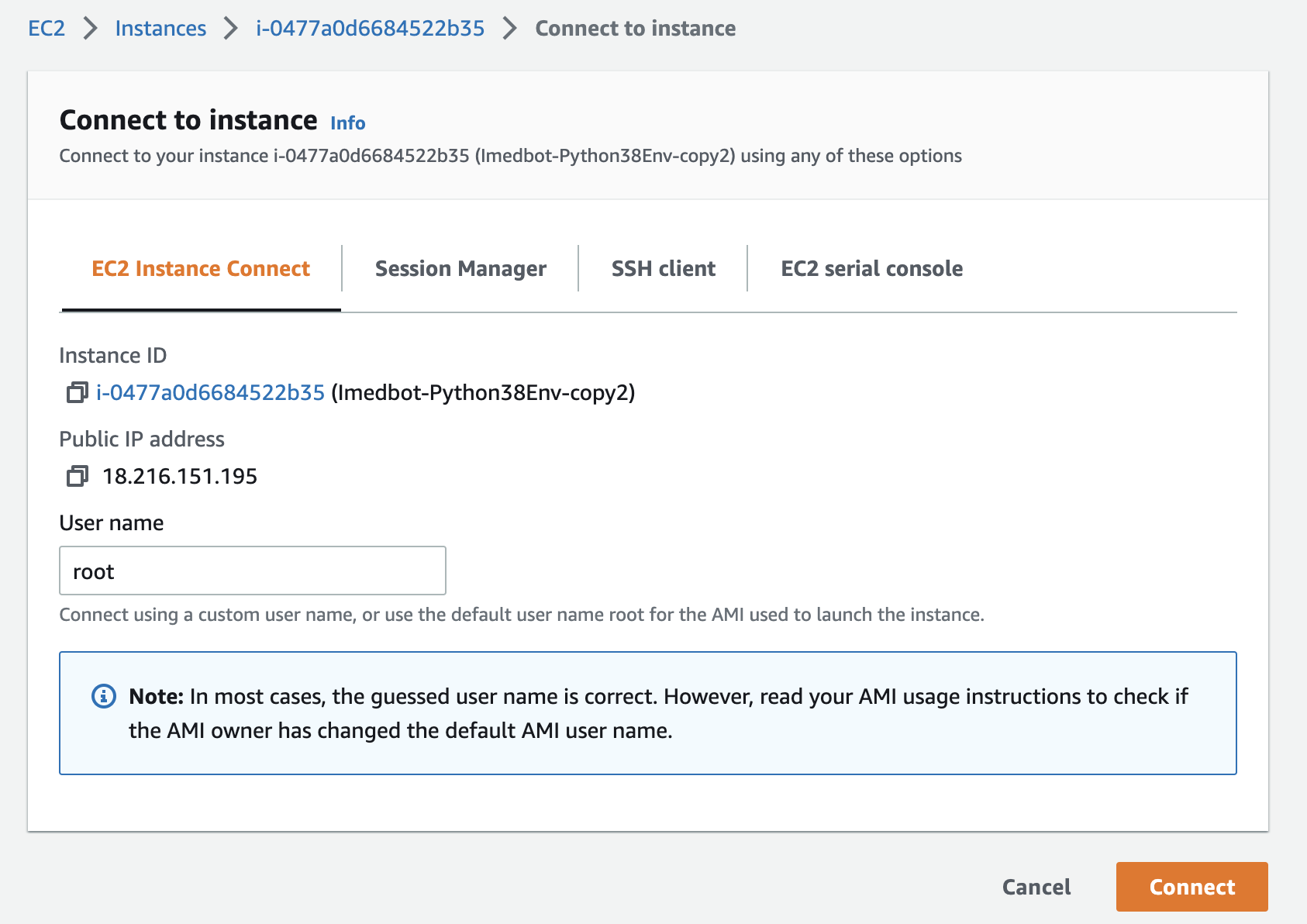


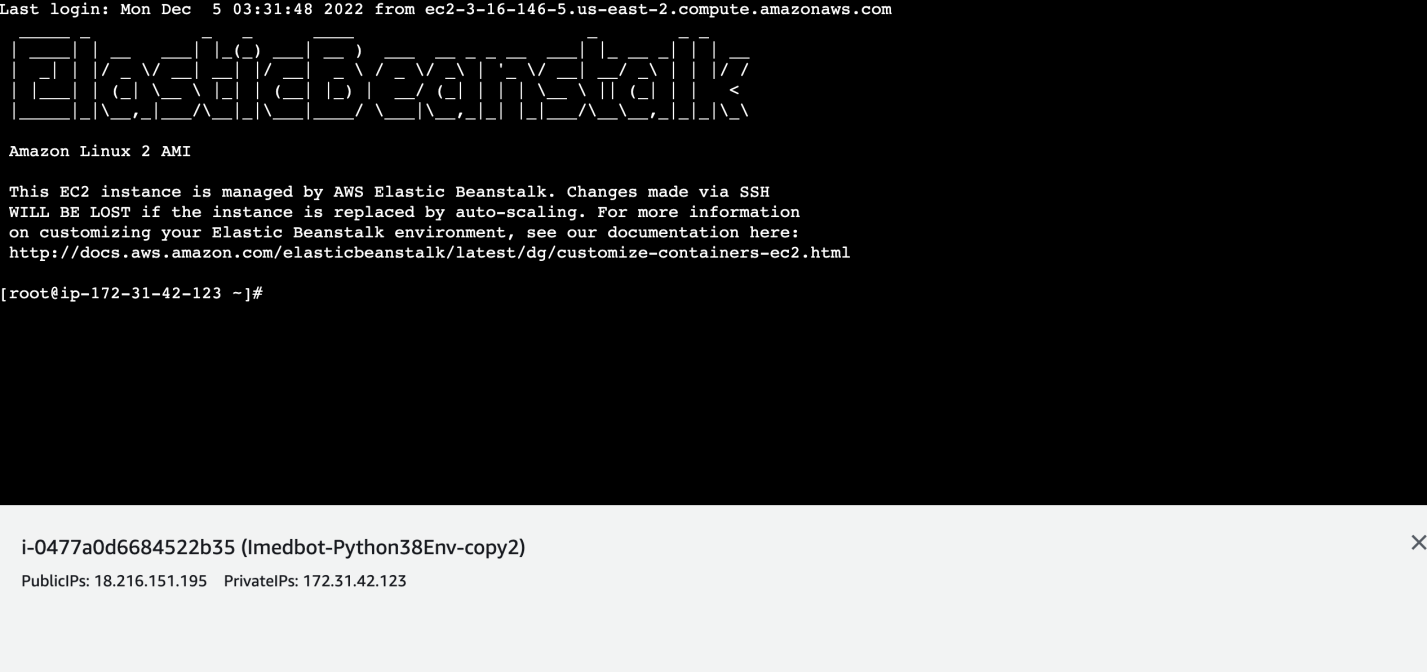


## Note: the commands in this terminal are tried lines, which are not required. See the commands useful in section B and C below.

## Use ssh to get access to ec2 instance

1. you can directly open terminal in aws. Open corresponding ec2 instance.

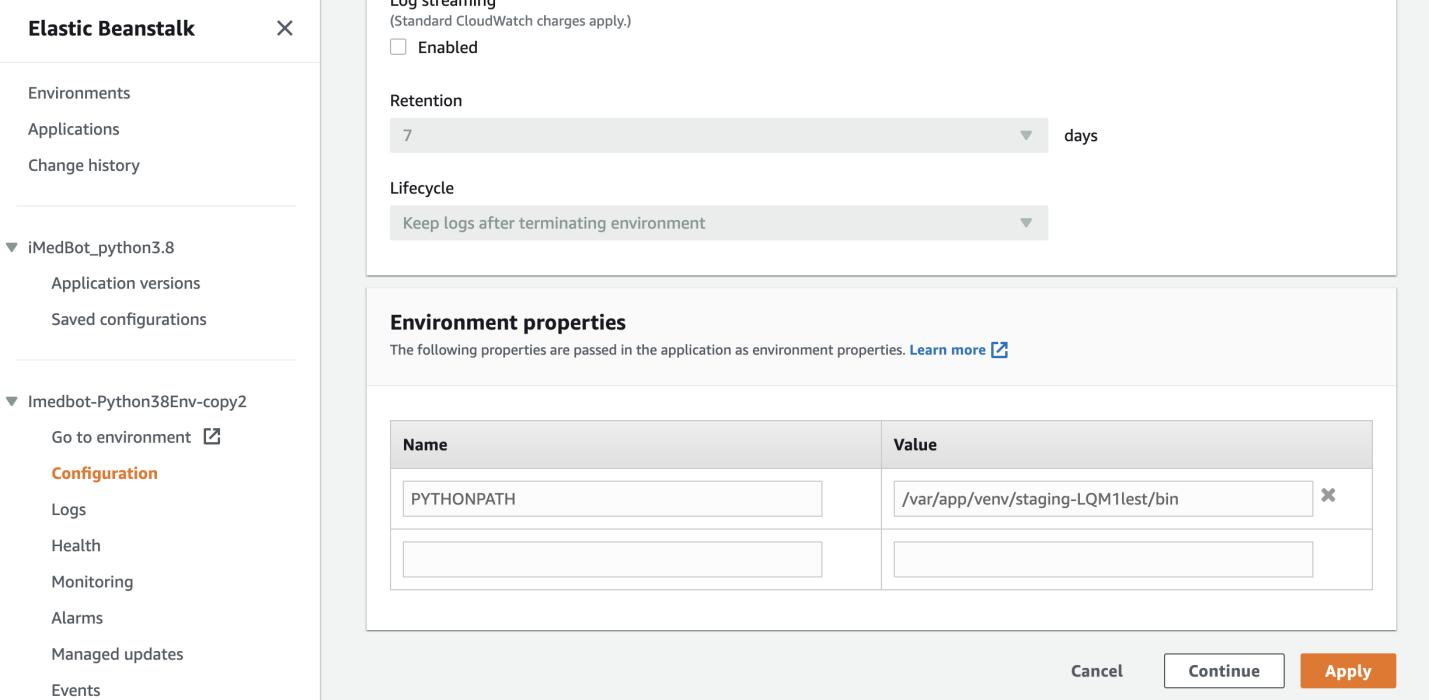


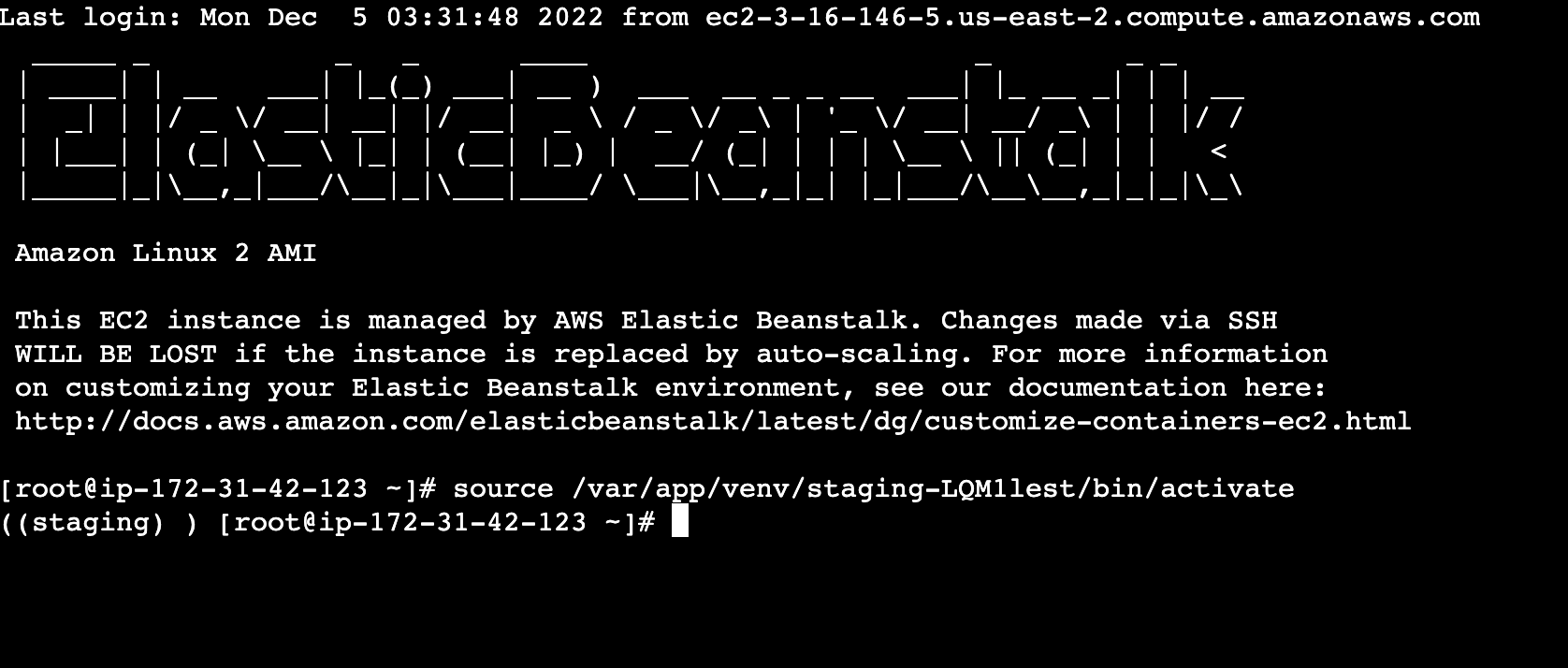


(this instance can open directly)

B. If you want to install python packages, activate the environment

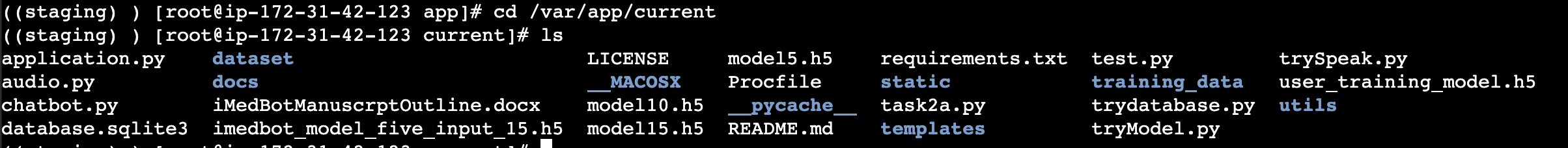
You can find the PYTHONPATH from the configuration page from Elastic Beanstalk environments





Then use pip install to install packages

C.if you want to see the code directly in ec2



The whole project is located in /var/app/current

If you want to see the code, use **nano application.py** to edit file in ec2

If you want to run the application in ec2, use **python application.py,** that will help to find the problem (why the internet is not working).

During the deployment during today’s meeting (2022.12.06), the iMedbot did not work, and we fixed the problem by going into the EC2 instance, following the procedure described above. See screenshot for the error we found and the fix. We first ran: python application.py, we found the following error.

Text

Description automatically generated

We just ran the following code to fix it:

pip install dnspython

2023.5.1

Redeployed iMedbot to AWA after we created a lab trelloboard space (copied the old board).

1. We created a new application at AWS.
2. Graphical user interface, application

   Description automatically generated
3. We created a new environment at AWS, we choose a larger instance see screenshot below.

Graphical user interface, text, application, email

Description automatically generated

We had problems the first time, and we download the first 100 log and have ChapGTP to translate it, and we found the following error:

Based on the log provided, there is a conflict in the package dependencies of **numpy**, **keras**, and **tensorflow**. The conflict is caused by the following:

* The user requested **numpy==1.20**
* **keras 2.4.3** depends on **numpy>=1.9.1**
* **tensorflow 2.5.0** depends on **numpy~=1.19.2**

To resolve the issue, the user could try to:

1. loosen the range of package versions specified
2. remove package versions to allow pip attempt to solve the dependency conflict

The log also shows that the deployment failed because of this error. The cleanup logic was executed after the failure, and the platform engine finished execution on the **app-deploy** command. The log then continues with the **tail-log** command to retrieve the tail logs.

Top of Form

Regenerate response

Bottom of Form

3). We then reverse the numpy version to the previous version in the requirements.txt to numpy 1.19.5.

4). We then regenerate the archive. Zip (the procedure is to download zip from code of github, unzip, remove the doc folder, then rezip. Even though we still received the warning (see below) at the new environment at AWS, but it worked this time. Hope the warning will be resolved.

Graphical user interface, text, application

Description automatically generated

5). We edit the records in the Route 53 at AWS to link the imedbot.odpac.net with the new environment.