Procedure to Setup Website using AWS.

Connect github to ec2 instance

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.

Graphical user interface, text, application, email

Description automatically generated

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

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

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

Key pair

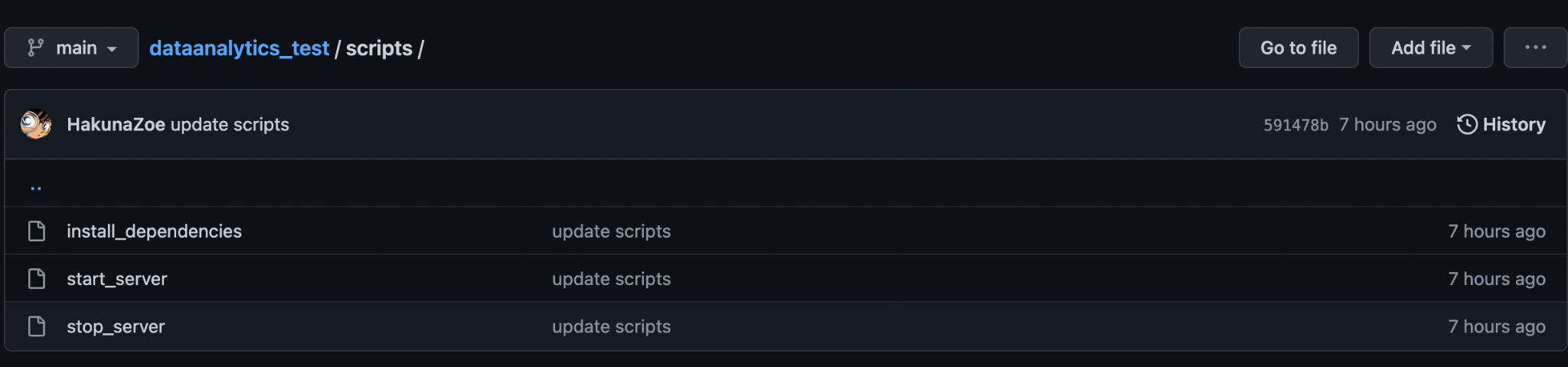
Graphical user interface, application

Description automatically generated

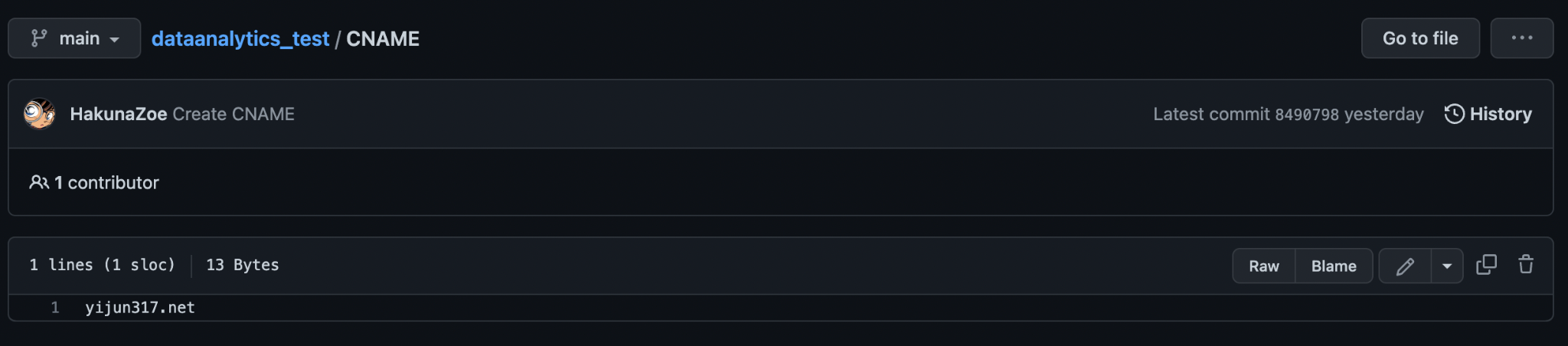
Graphical user interface, text, application, email

Description automatically generated

1. Add configuration in github
2. Add scripts



1. Add cname



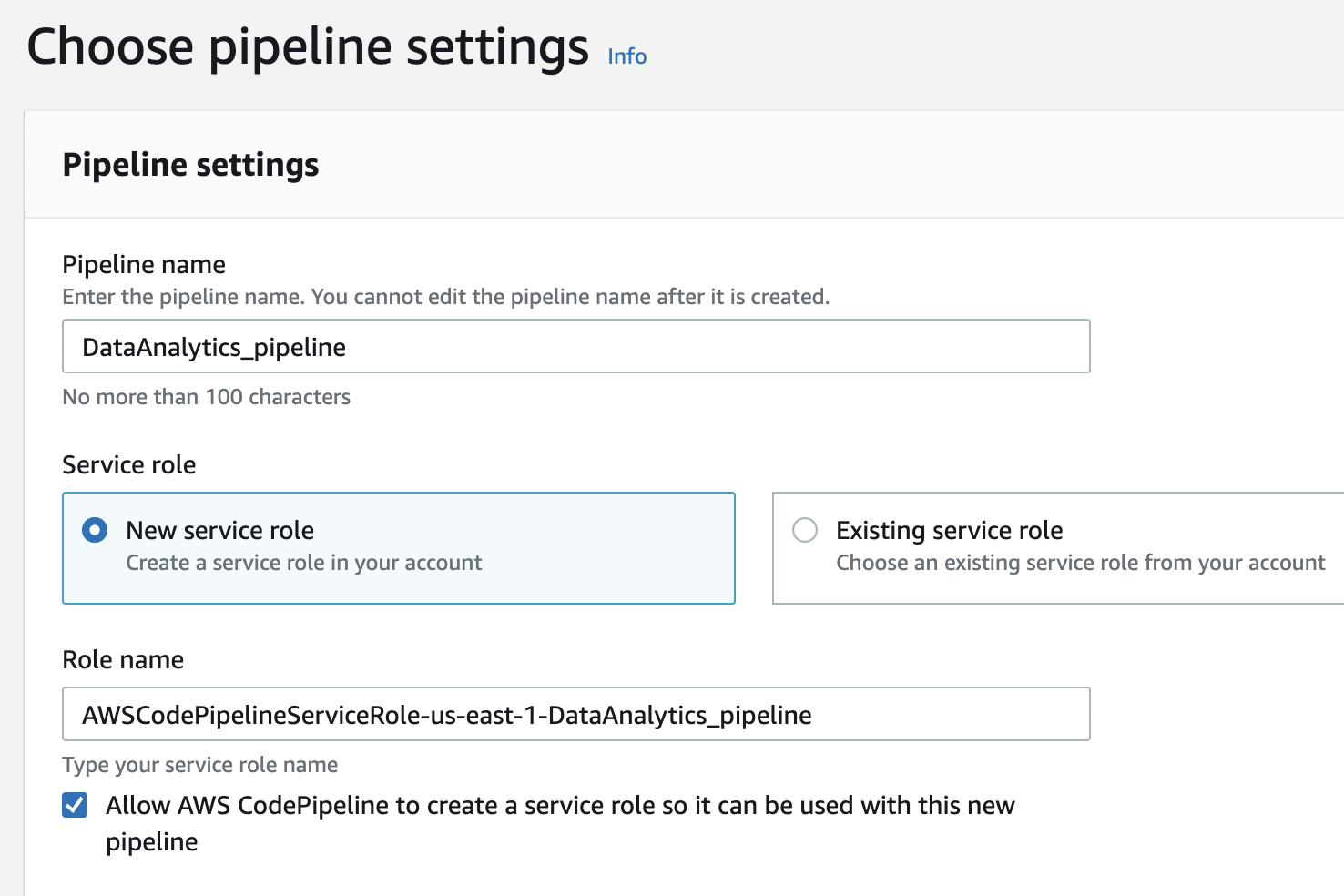
1. Add appspec.yml



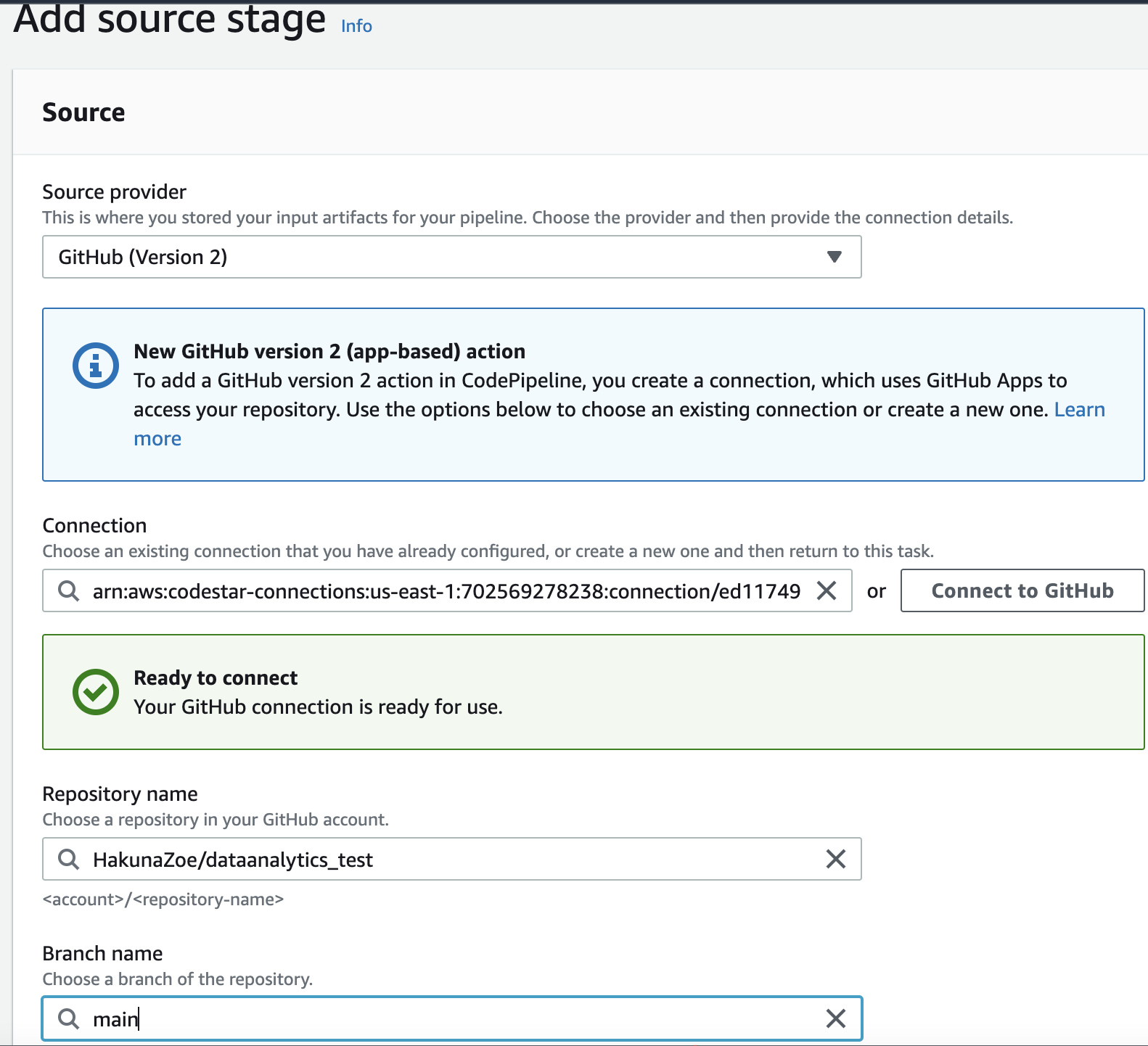
If the scripts already exist, we can just edit it. Question: Why we don’t need this to deploy “Charts”, see our previous “Charts” repo.

For the new application we just created, we only revised appspec.yml, by changing the name of application.

1. Create a code pipeline at AWS.
2. Question: what is the difference between the new service role and the existing service role? Can we use the existing service role for the new code pipeline settings? Does creating new role cause more money charged by AWS?



1. Add source stage

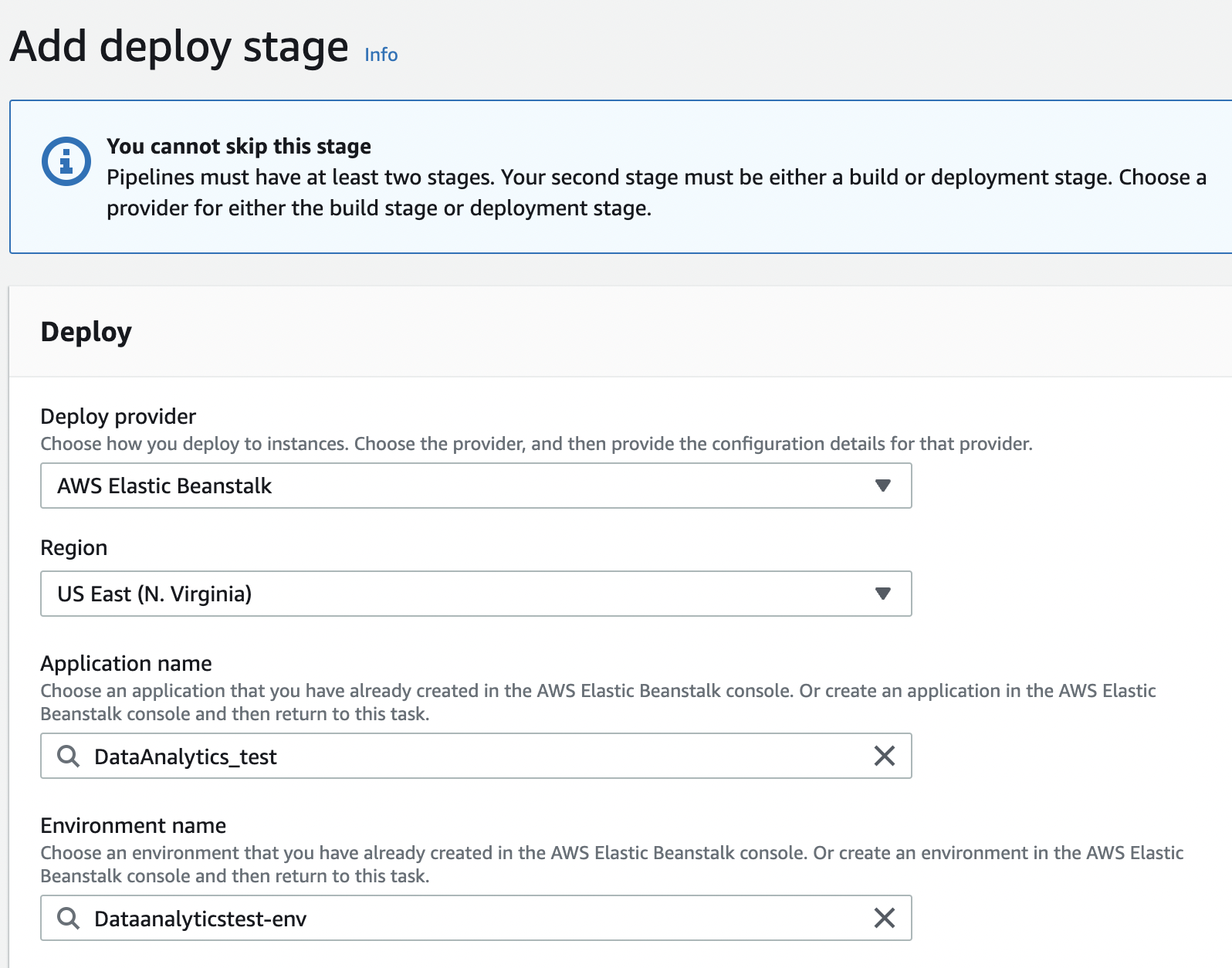


Graphical user interface, text, application, email

Description automatically generated

These choices may help avoiding the charge associated with the push in the past.

1. Add build stage( could skip)
2. Add deploy stage



Graphical user interface, text, application

Description automatically generated

1. Connect the instance to domain in route53

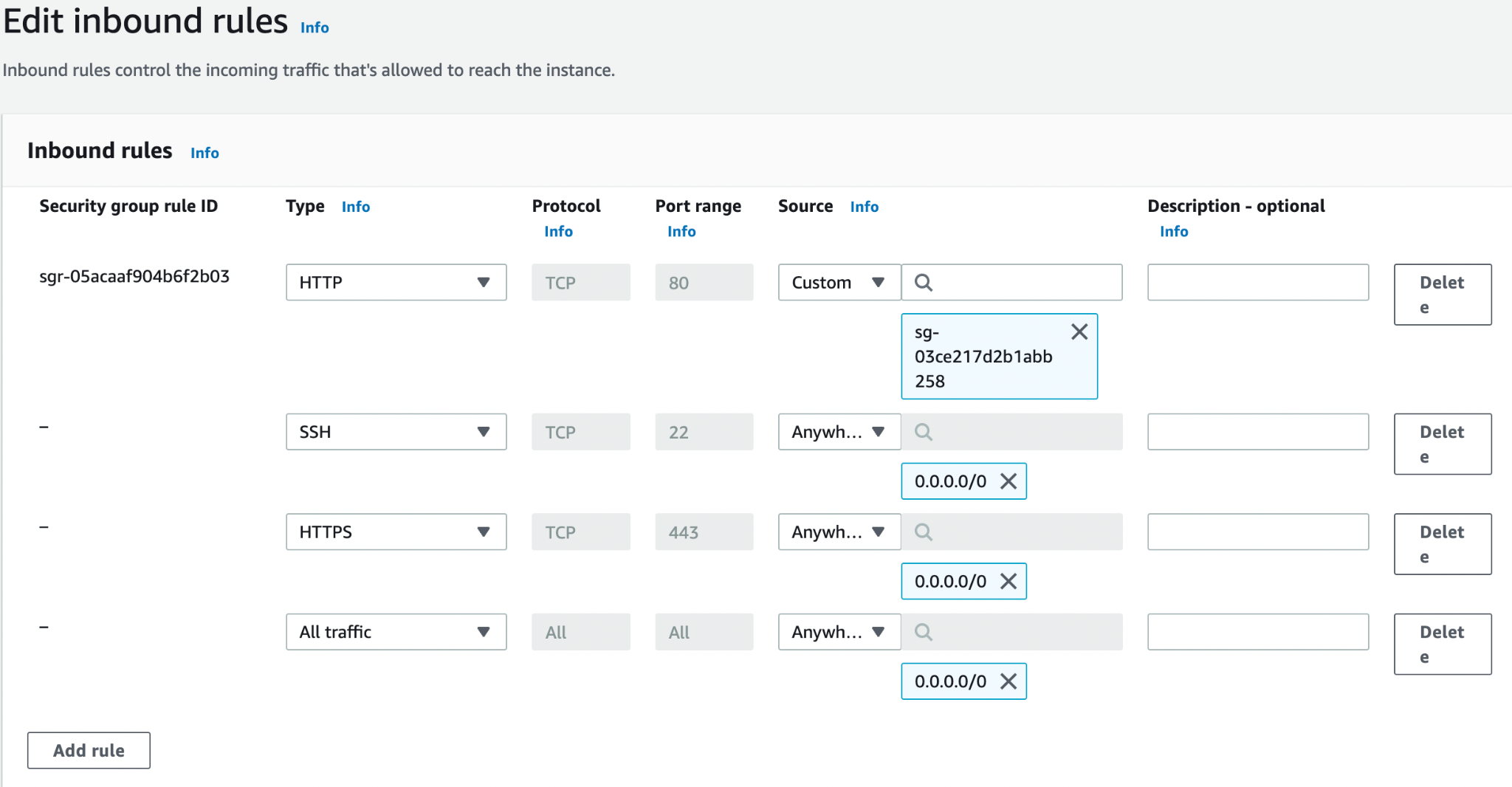
Locate the hosted zones, then create record or edit existing record (in which case, you need to select the new environment).

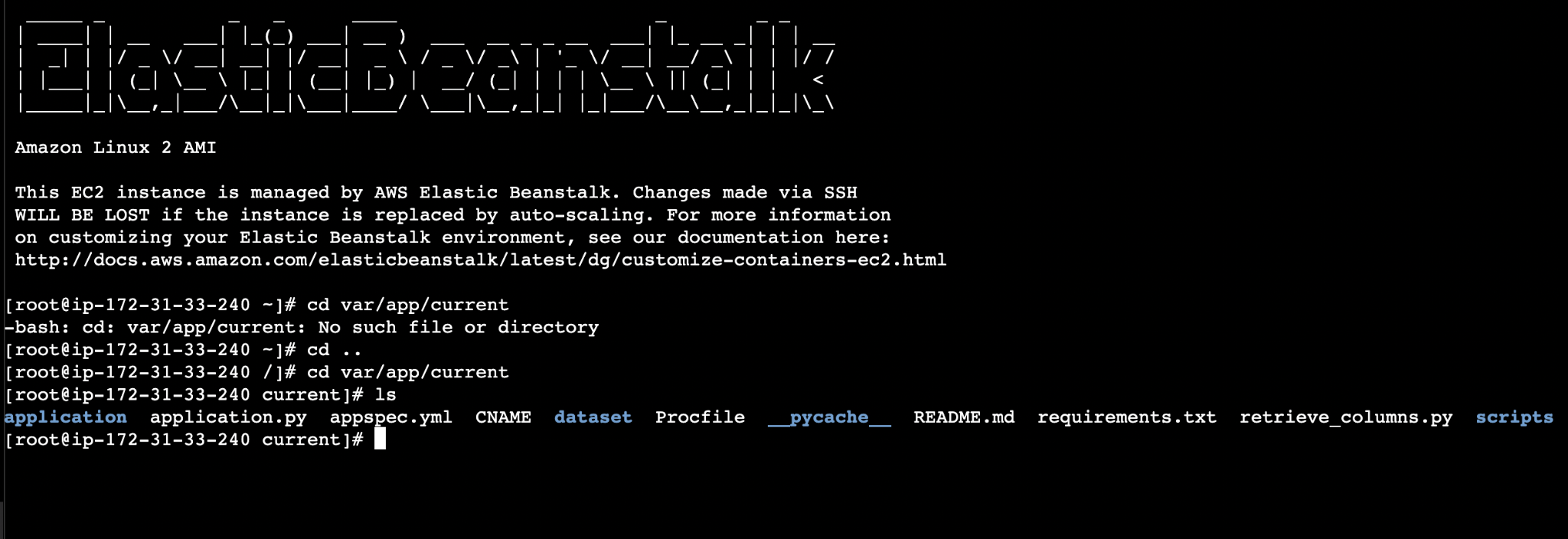
Graphical user interface, text, application, email

Description automatically generated



# Use ssh to open server and find out application





## 

## Use ssh to get access to ec2 instance

ssh -i "yijun-zhou-key.pem" ec2-user@ec2-44-202-57-201.compute-1.amazonaws.com

ssh -L 5901:localhost:5901 -i "yijun-zhou-key.pem" ec2-user@ec2-54.161.45.196.compute-1.amazonaws.com

Use tigervnc to open the remote desktop of ec2 instance

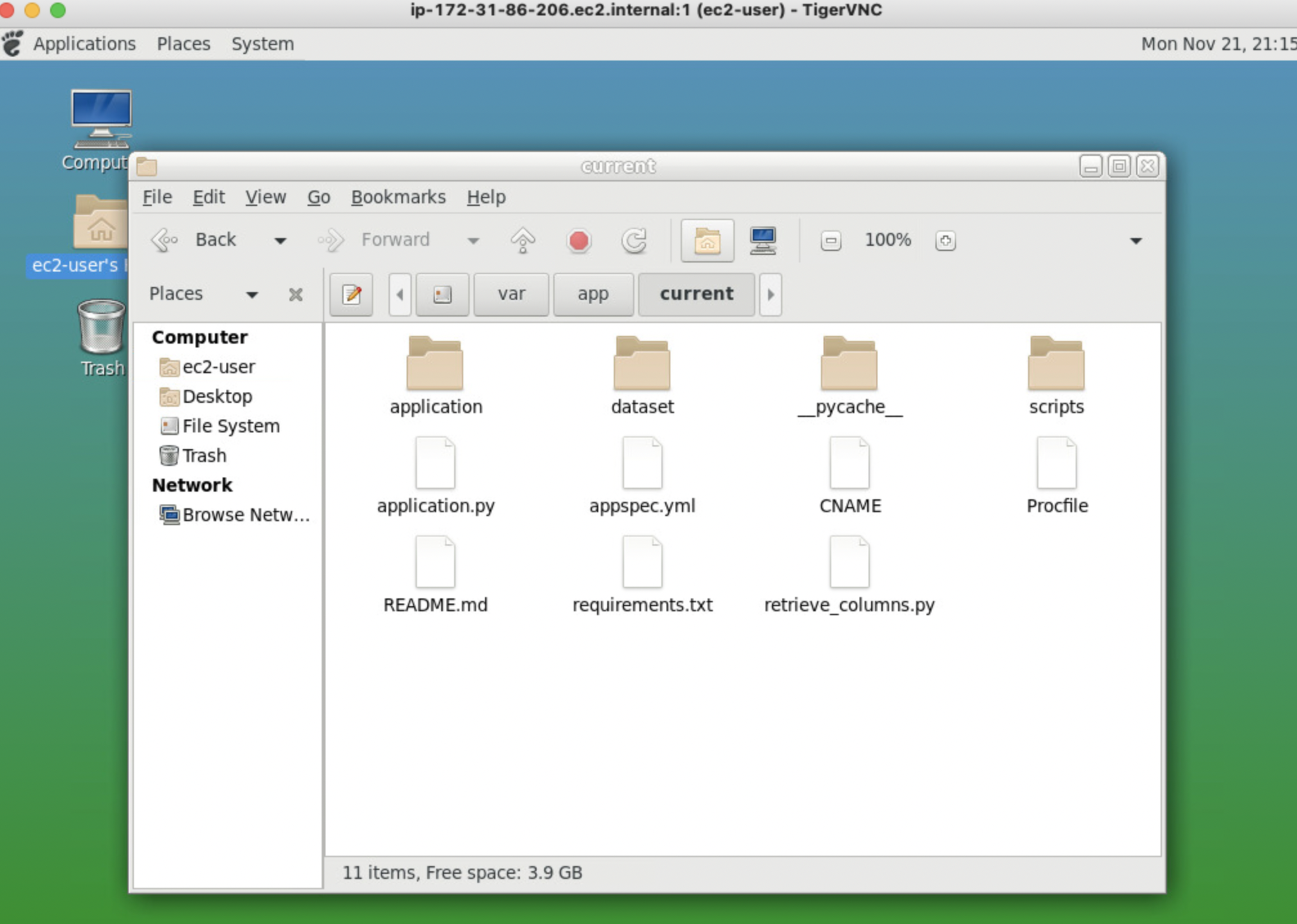
1. Put yijun-zhou-key.pem in the root directory(hakuna)
2. chmod 400 yijun-zhou-key.pem
3. ssh -i "yijun-zhou-key.pem" ec2-user@ec2-44-202-57-201.compute-1.amazonaws.com
4. Set password for ec2-user

Sudo passwd ec2-user

1. sudo bash -c 'echo PREFERRED=/usr/bin/mate-session > /etc/sysconfig/desktop'
2. sudo yum install tigervnc-server
3. Set vncpasswd (choose no view-only password)

vncpasswd

1. Sudo mkdir /etc/tigervnc
2. sudo bash -c 'echo localhost > /etc/tigervnc/vncserver-config-mandatory'
3. sudo cp /lib/systemd/system/vncserver@.service /etc/systemd/system/vncserver@.service
4. sudo sed -i 's/<USER>/ec2-user/' /etc/systemd/system/vncserver@.service
5. sudo systemctl daemon-reload
6. sudo systemctl enable vncserver@:1
7. sudo systemctl start vncserver@:1
8. Then open tigervnc, and type in localhost:1 in the host name, click connect



tutorial : <https://www.youtube.com/watch?v=DJ5gedaI4EY&t=326s>

Change the maximum uploading file size

1. sudo nano /etc/nginx/nginx.conf
2. Add client\_max\_body\_size 50M;
3. sudo service nginx restart

https://medium.com/fredwong-it/increase-the-upload-file-size-to-aws-ec2-a9df70d770cb