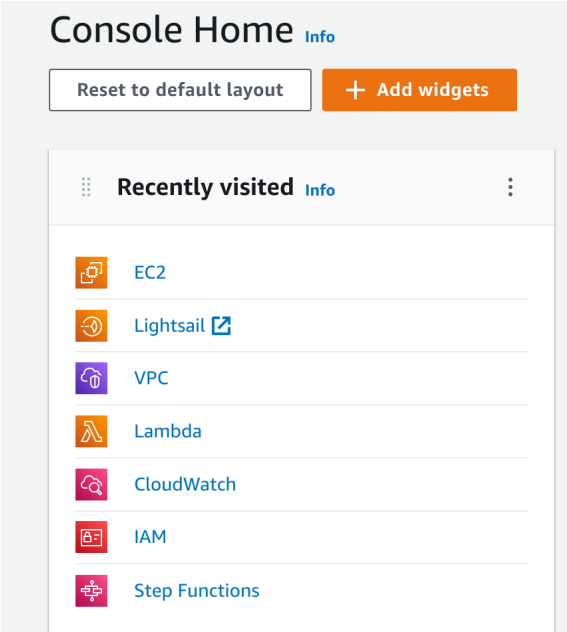
aws_public_ec2_setup

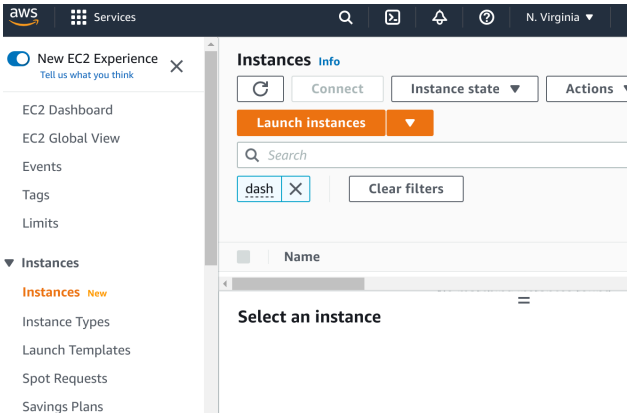
Instructions to set up public AWS EC2
e.g. to host a flask server, dashboard, REST api endpoint, etc.
(See picture version as pdf in repo, pictures may help!)

Go to: AWS
https://us-east-1.console.aws.amazon.com



Go to: EC2
https://us-east-1.console.aws.amazon.com/ec2/

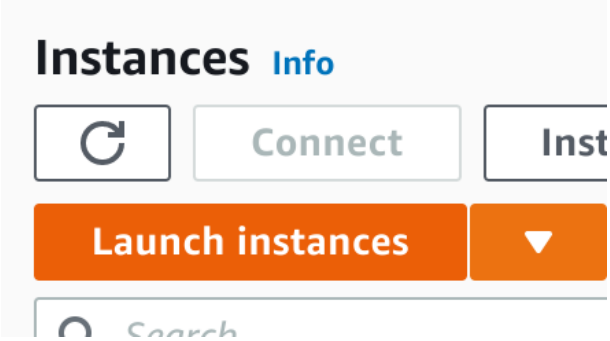
Go to: instances
the instances tab



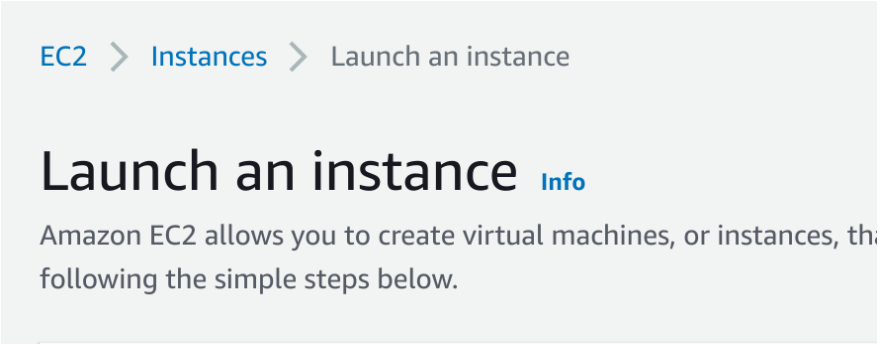
Launch Instance:

Hit the big orange button that says "Launch instances"

(plural...for some reason...which of course takes to you "launch an instance" singular)



 # "launch an instance" singular



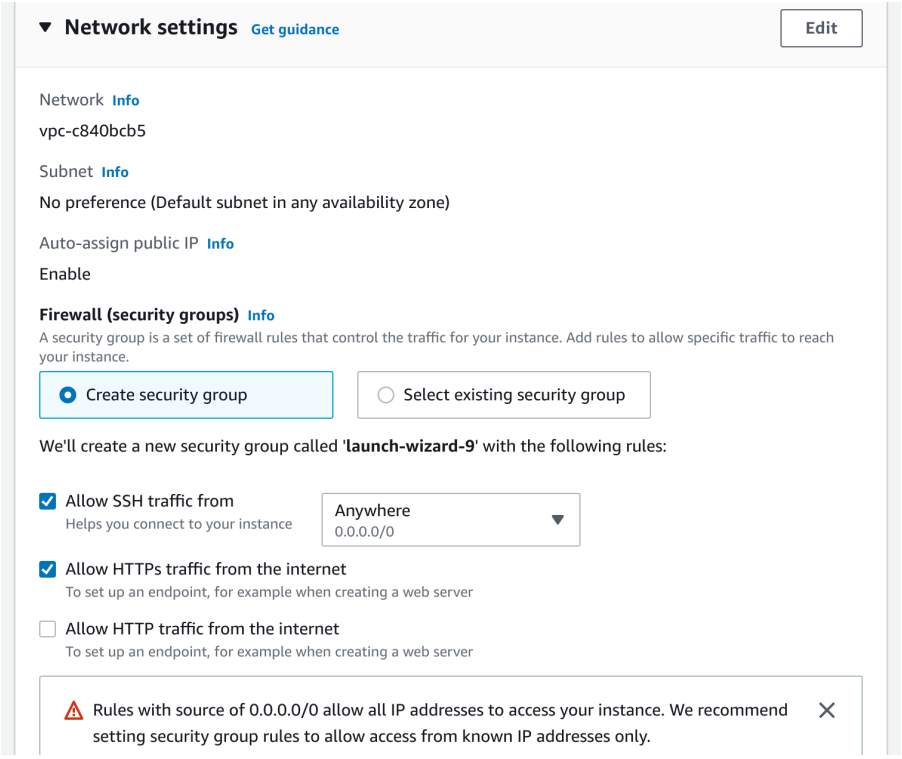
Configure:

1. Name and tags -> clear meaningful name, nothing is too obvious. recommended format: "ec2_purpose_yourname_datetime"
2. Application and OS Images (Amazon Machine Image) -> default amazon linux
3. Instance type -> nano (scroll down)
4. Key pair (login) -> select or make new pair
5. Network settings...(see below)

Network settings:

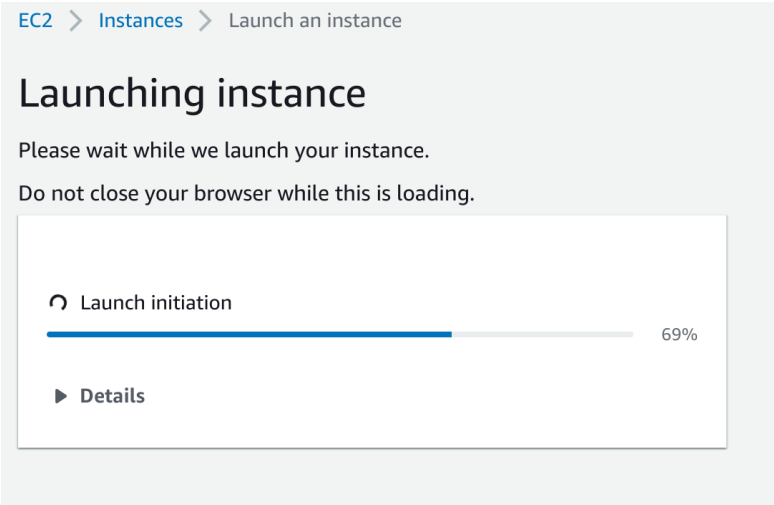
1. firewall security group: create or select
2. "Allow SSH traffic from": must be on to use EC2 connect later (or SSH in yourself)
3. "Allow HTTPs traffic from the internet": If you want this to be public, allow.

4. http: maybe leave insecure http off.

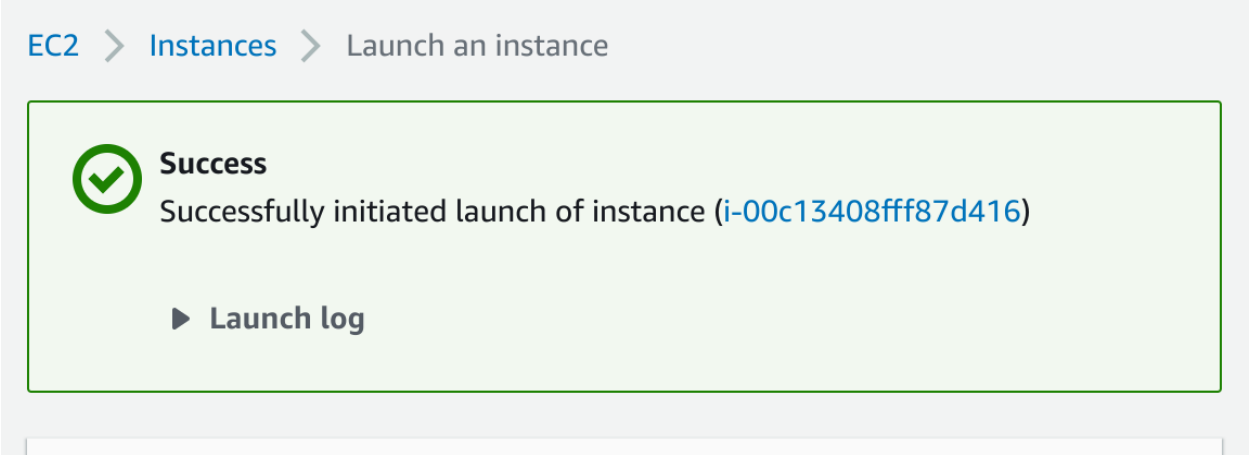


Network settings (continued...)
6. Storage (volumes -> use default
7. Advanced details Info -> ignore
8. Summary -> nothing to do or change here, examine if you want.

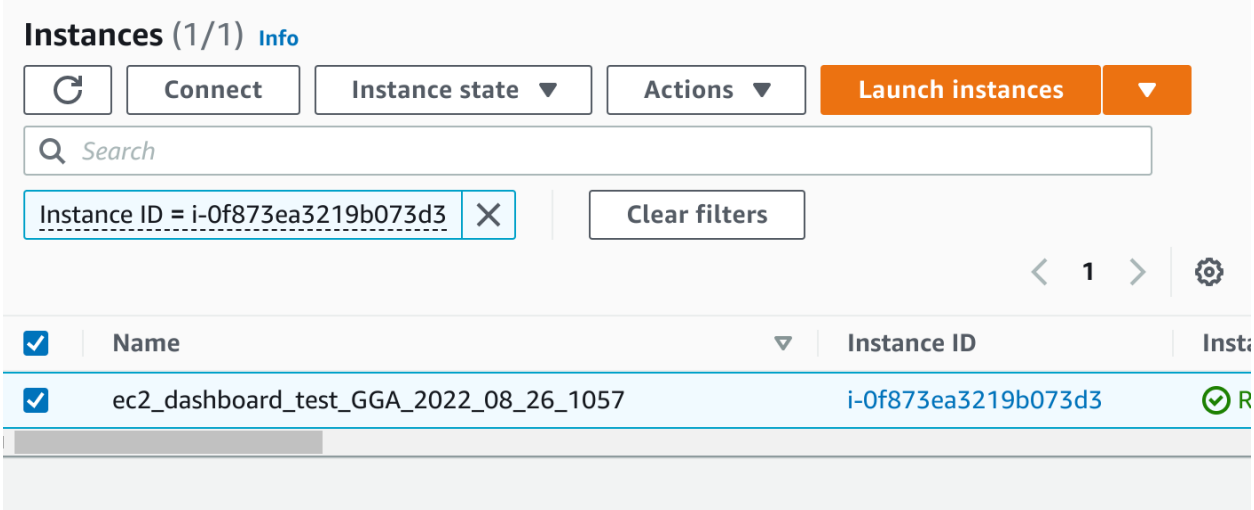
Select: Launch Instance



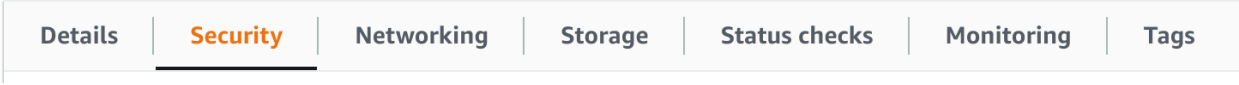
click on blue random string .... (obviously!...?)



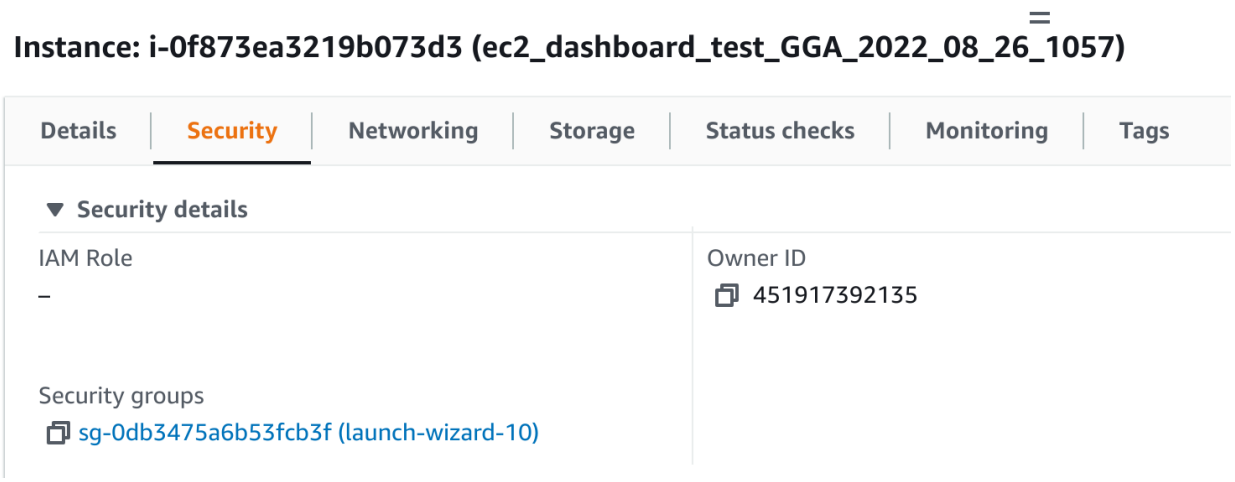
Back at instances window:
your instance should now be highlighted: click on "connect" to connect via web
This is much easier that local-cli ssh (web connect is one of the few actually useful working advances AWS has made.



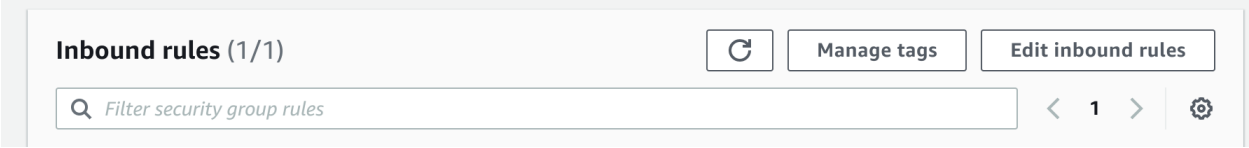
configure in "security"
(Obviously, since you want to do network configuration, and you have the choice of 'networking' you instead need to go to "security." So user friendly.)



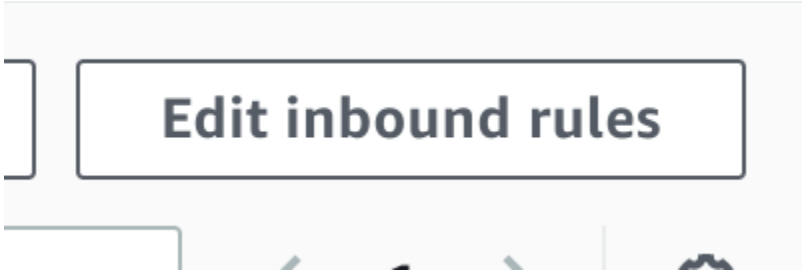
Another random blue-string-click
In the "Security" tab, under "security groups" (plural?) you see a random blue-string-link. click on that (to configure networking...obviously...)



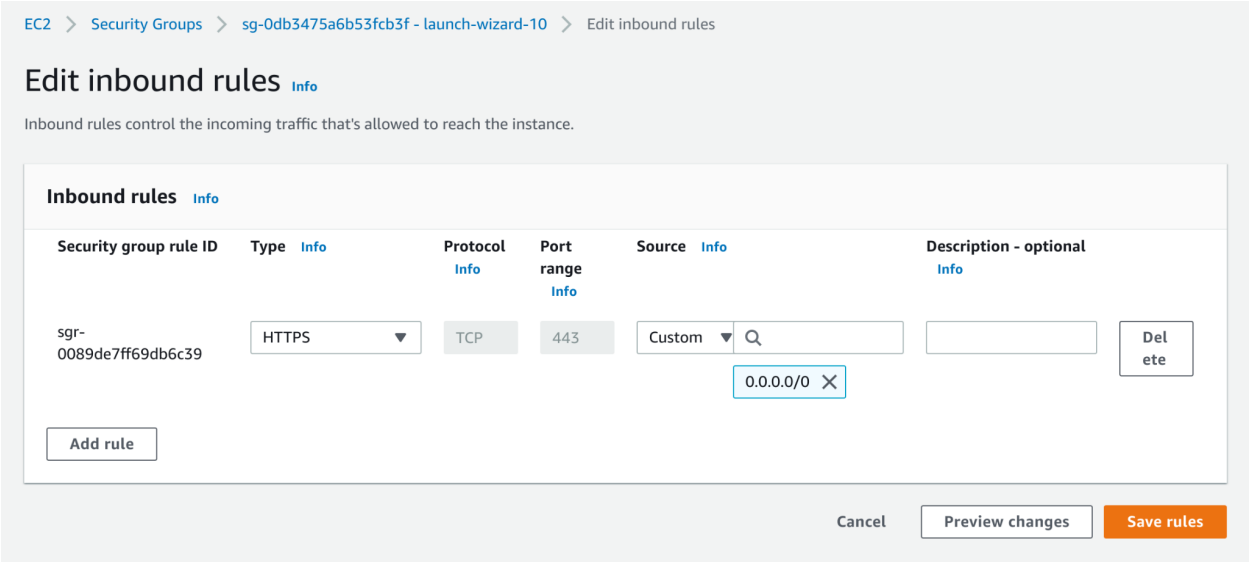
"Inbound rules" You are here!
Finally: This is the basic, rudimentary, necessary, "start here" configuration menu that all this has been leading up to (and should have started with), yet for some obscene reason AWS makes it impossible to even find.



Click "edit inbound rules"

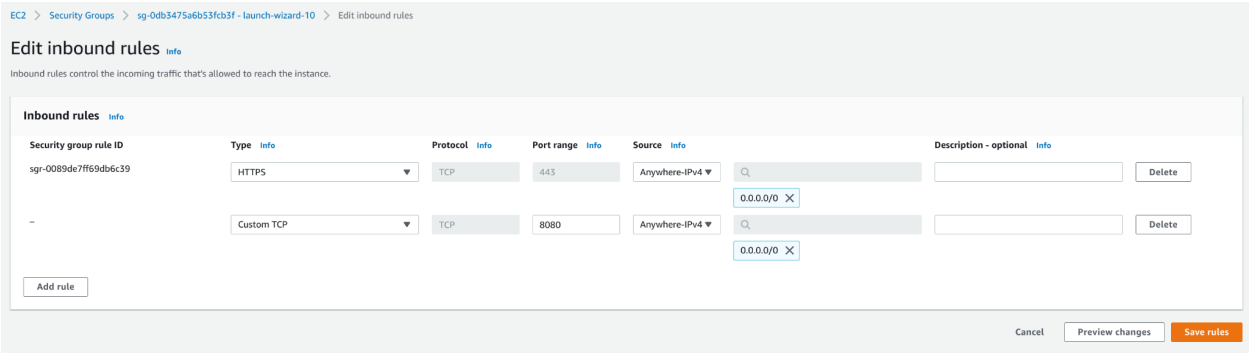


Make and save new rules.
Using the following tool (which you should see now),



Create and save (using the big orange "Save rules" button) the rules in this table.
Existing rules may need to be modified or replaced (e.g. HTTPS may be set to custom, set it to Anywhere IPV4)

```
      Type          (Protocol)  Port Range  Source          (to)
1.    HTTPS TCP     TCP         443         Anywhere IPV4   0.0.0.0/0
2.    Custom TCP    TCP         8080        Anywhere IPV4   0.0.0.0/0
3.    SSH           TCP         22          Custom          0.0.0.0/0
```



Another example rule set:

Set **Type** *HTTP*, **Protocol** *TCP*, **Port range** *80*, and **Source** to "*0.0.0.0/0*".

Set **Type** *HTTP*, **Protocol** *TCP*, **Port range** *80*, and **Source** to "*::/0*".

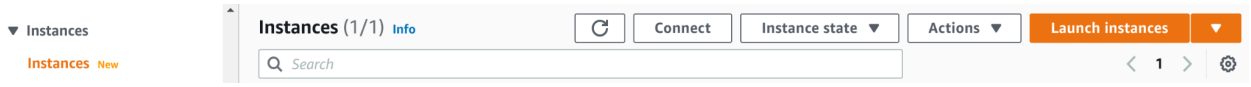
Set **Type** *Custom TCP*, **Protocol** *TCP*, **Port range** *8080*, and **Source** to "*0.0.0.0/0*".

Set **Type** *SSH*, **Protocol** *TCP*, **Port range** *22*, and **Source** to "*0.0.0.0/0*".

Set **Type** *HTTPS*, **Protocol** *TCP*, **Port range** *443*, and **Source** to "*0.0.0.0/0*".

Done.

Go back to the instances tab



Note!

The exact ports you need to select (e.g. 8080 vs. 8050) etc, may depend on what you are doing, and on how your project is configured (flask, dash, fast-api, etc.)

For plotly dash you may need to use 8050 and use this line in your app.run command:

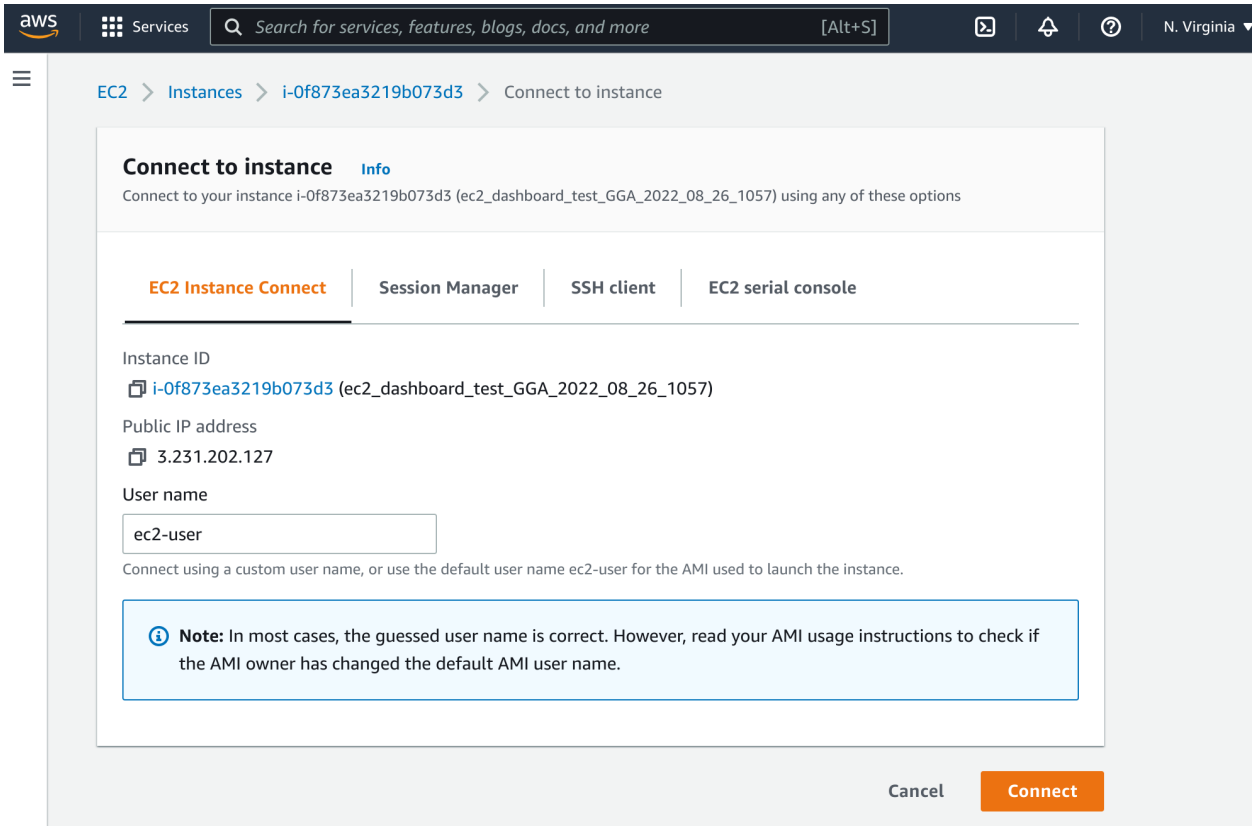
```
if __name__ == '__main__':
    app.run_server(host= '0.0.0.0',port=80)
```

And you may need to add a port suffix after the ipv4URL you get from AWS.
In these working examples, plotly-dash's port 8050 was added to the end of the original url.
```
http://3.94.153.137:8050/
or
http://ec2-3-94-153-137.compute-1.amazonaws.com:8050/
```

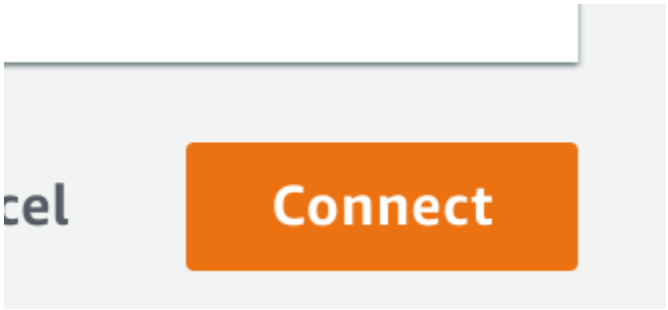
Web Connect

click on "connect"

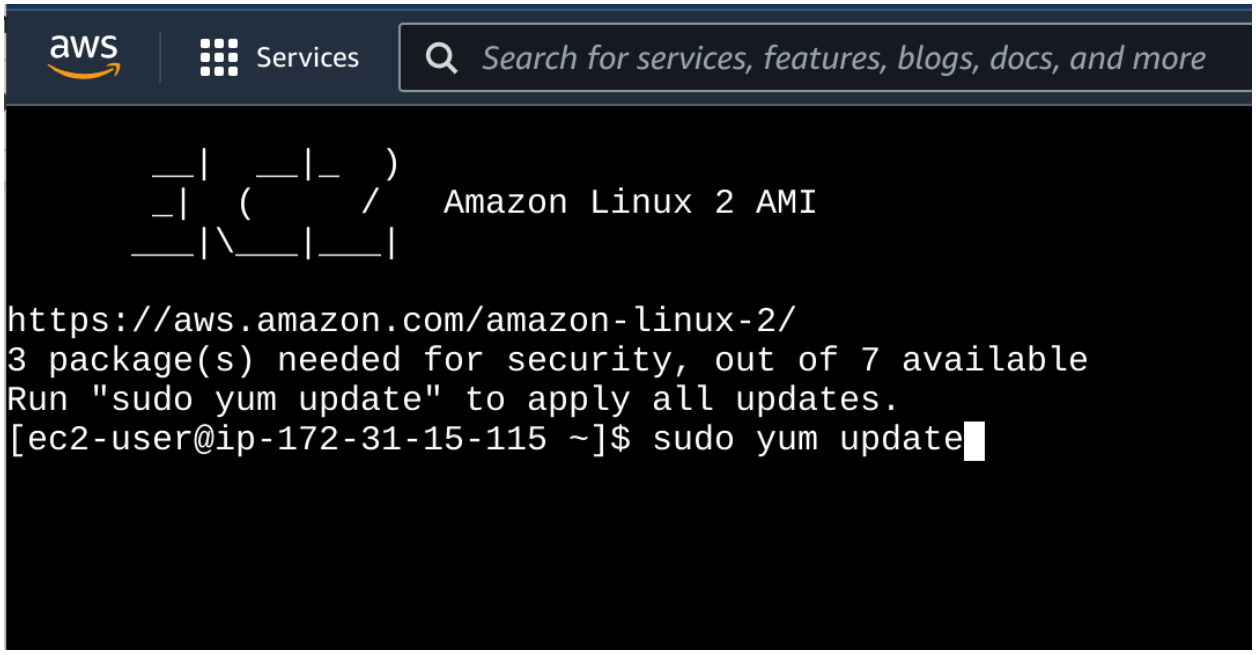
In the 'connect to instance' window
in the "EC2 Instance Connect" tab...



Click on "Connect" (the big orange button)...(dejavu?)



Like SSH but with no convoluted local aws-cli setup nightmare. (This is a good thing.)



Optional steps
You are effectively done, but you may want to run these lines, e.g. if you are going to get files from github

```
$ sudo yum update -y
$ sudo yum install git -y
```

Reminder

You may need to add a port suffix after the ipv4URL you get from AWS.
In these working examples, plotly-dash's port 8050 was added to the end of the original url.
```
http://3.94.153.137:8050/
or
http://ec2-3-94-153-137.compute-1.amazonaws.com:8050/
```