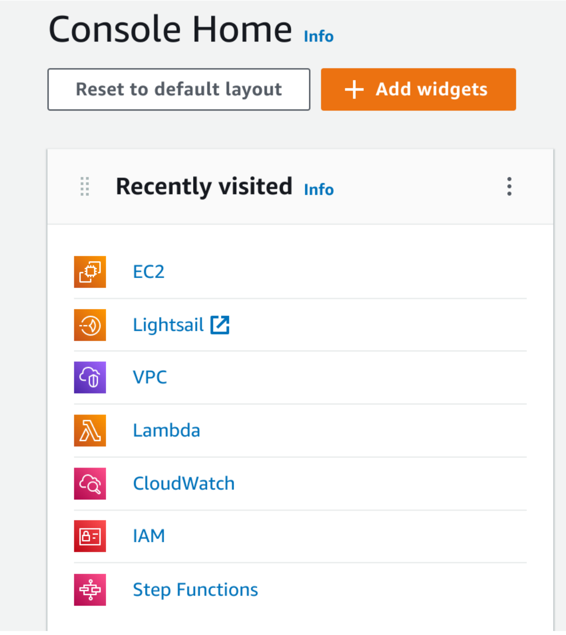
#### aws_public_ec2_setup

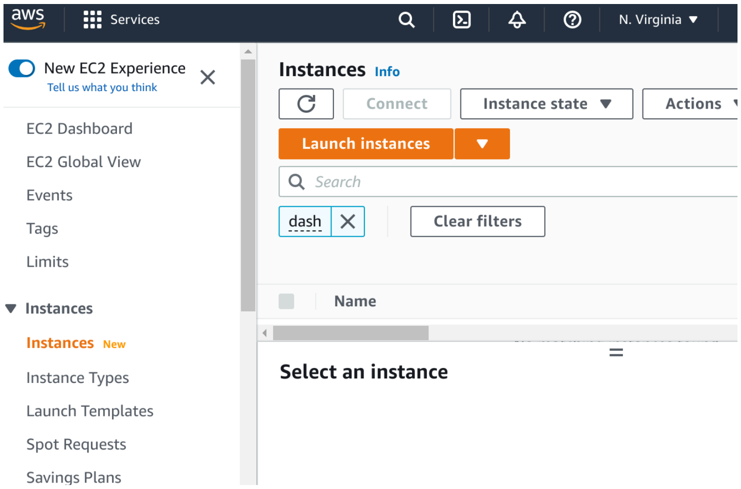
# Instructions to set up public AWS EC2
e.g. to host a flask server, dashboard, REST api endpoint, etc.
(See picture version as pdf in repo, pictures may help!)

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



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

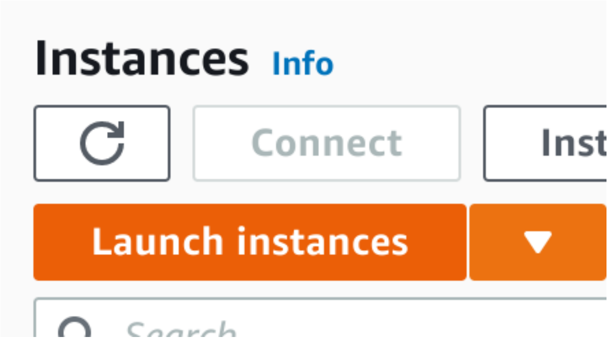
### Go to: instances
the instances tab



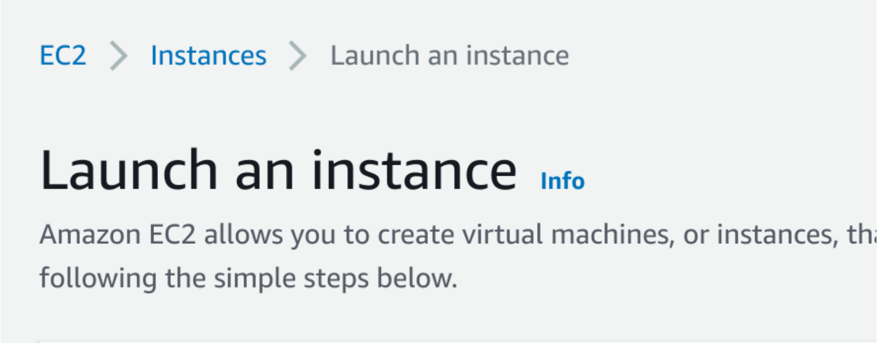
## Launch Instance:

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*



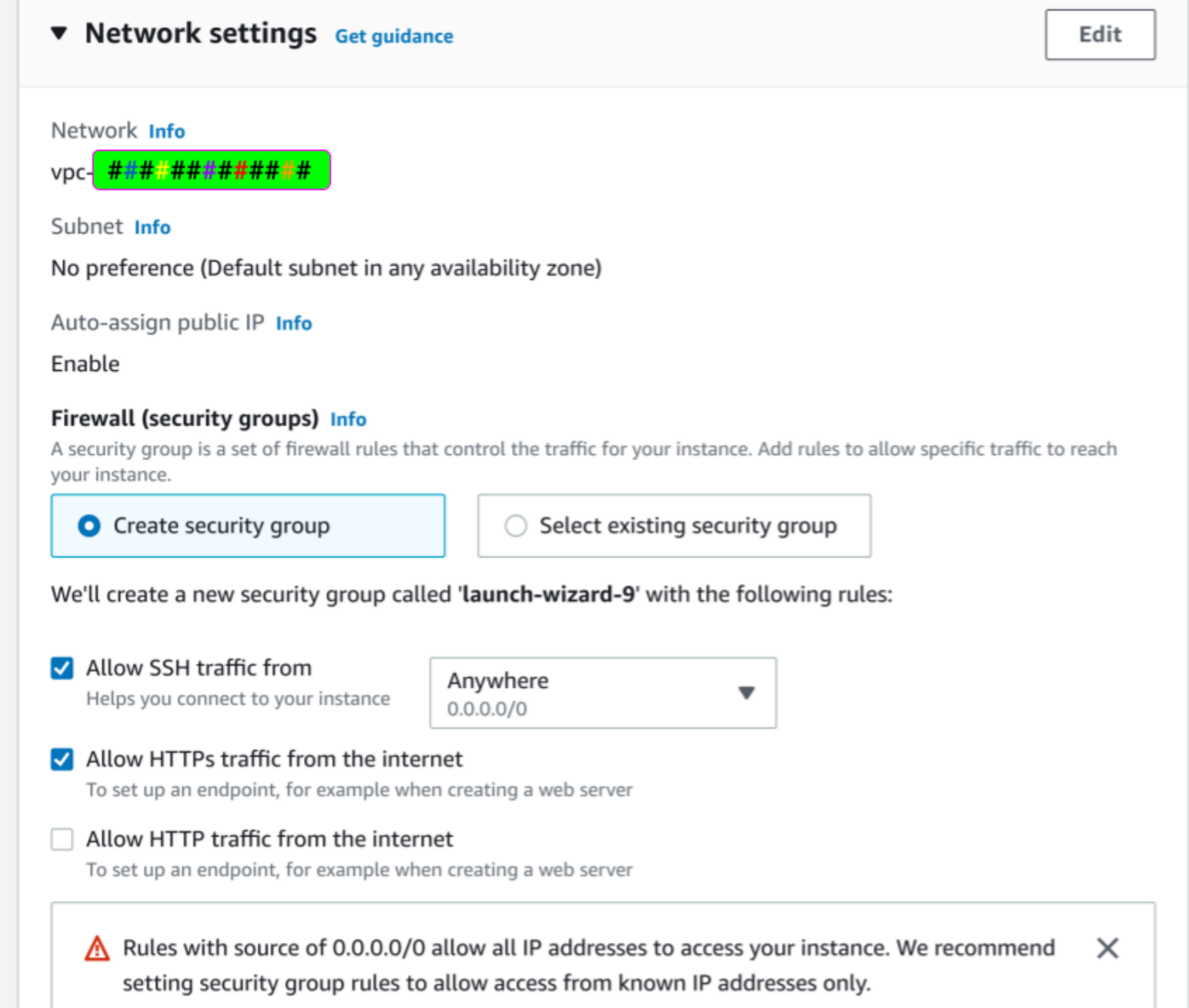
Amazon EC2 allows you to create virtual machines, or instances, th
following the simple steps below.

## 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 may be needed in the mess of aws connection issues, leave it on for now



▼ Network settings Get guidance

Edit

Network Info

vpc-[redacted]

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group   ○ Select existing security group

We'll create a new security group called '**launch-wizard-9**' with the following rules:

☑ Allow SSH traffic from — Anywhere 0.0.0.0/0 ▼
Helps you connect to your instance

☑ Allow HTTPs traffic from the internet
To set up an endpoint, for example when creating a web server

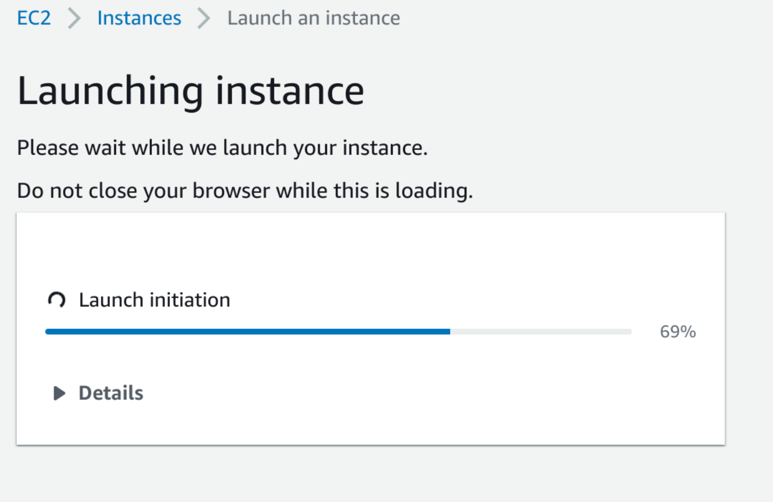
□ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. ✕

### 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



Launching instance

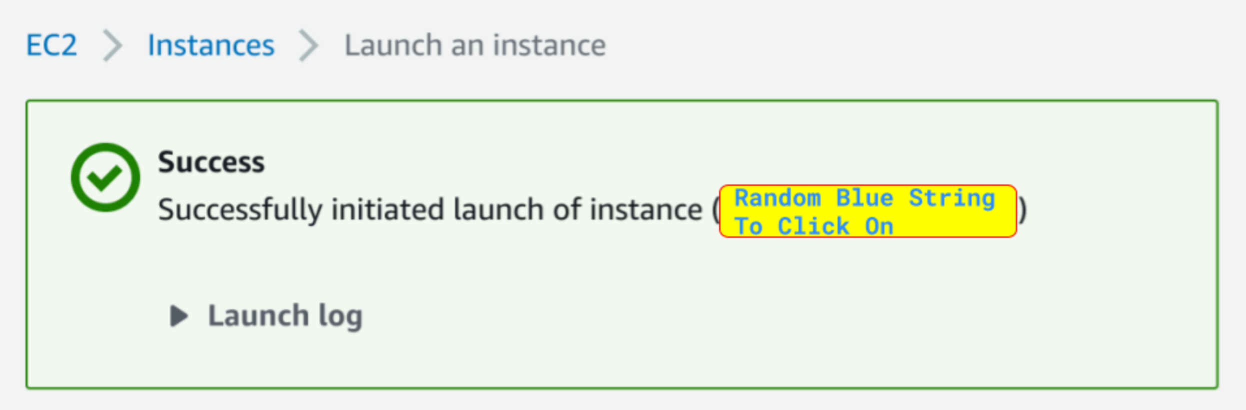
Please wait while we launch your instance.

Do not close your browser while this is loading.

Launch initiation

69%

## Click on the blue random string .... (obviously...?)

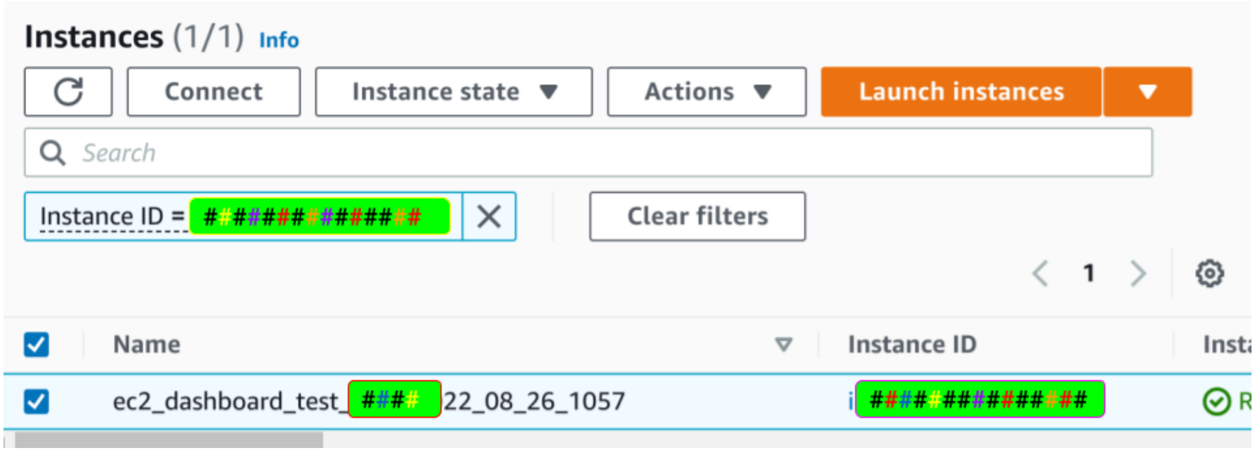


Success

Successfully initiated launch of instance (Random Blue String To Click On)

## 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.

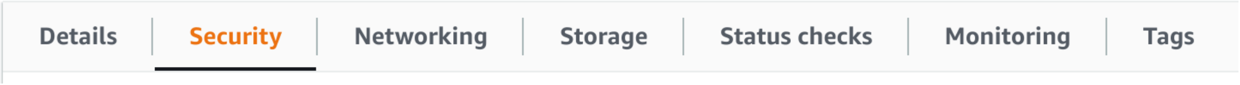


Instance ID = ############### ✕

☑	Name	Instance ID	Insta
☑	ec2_dashboard_test_####22_08_26_1057	i###############	✓ R

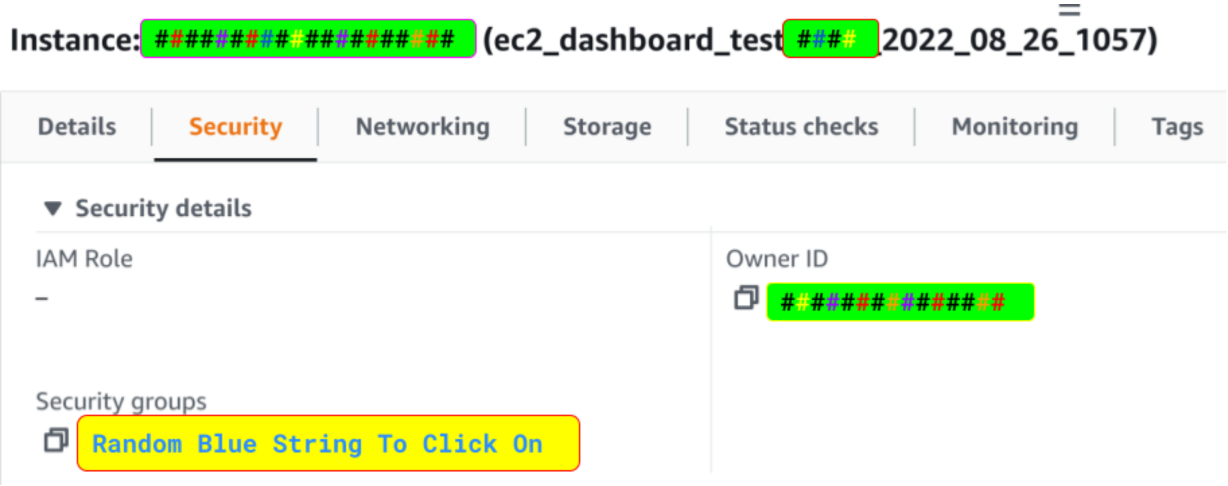
# configure in "security"

(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...)


Instance: #################### (ec2_dashboard_test####2022_08_26_1057)

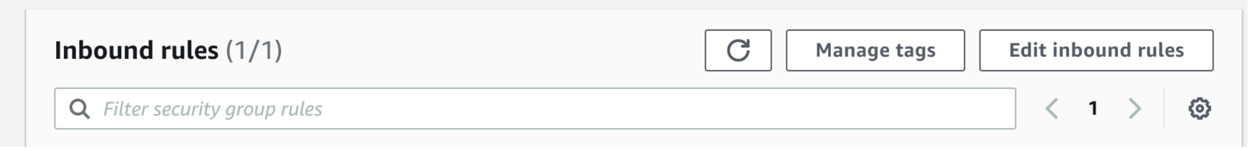
IAM Role
–

Owner ID
###############

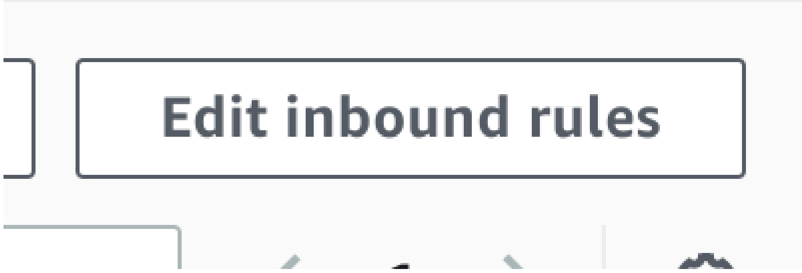
Security groups
Random Blue String To Click On

## "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.

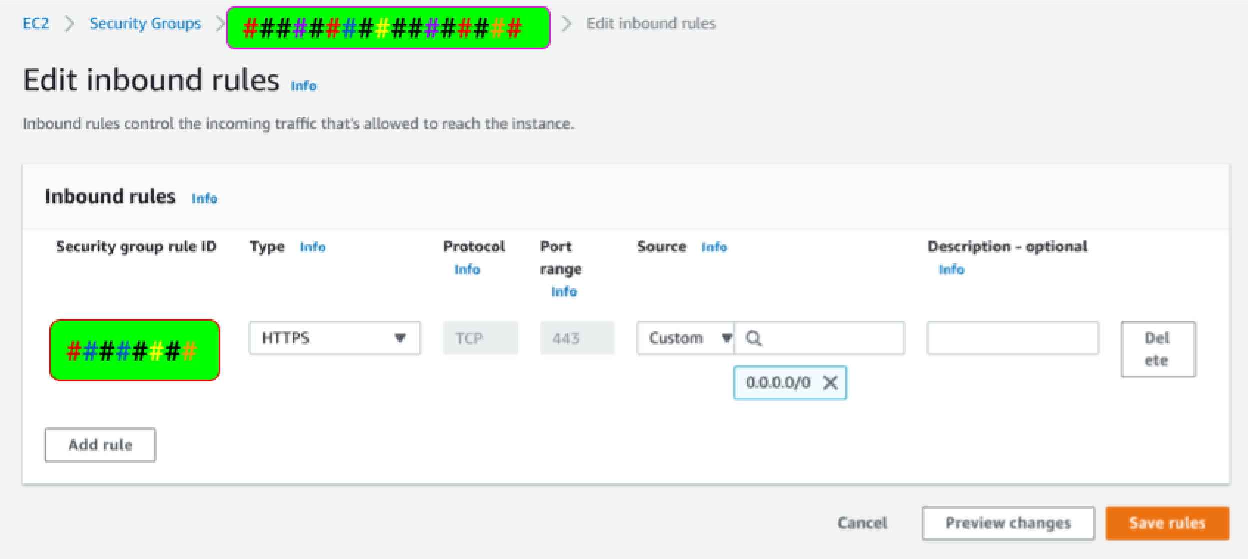


### Click "edit inbound rules"



### Make and save new rules.

Using the following tool (which you should see now),



Edit inbound rules

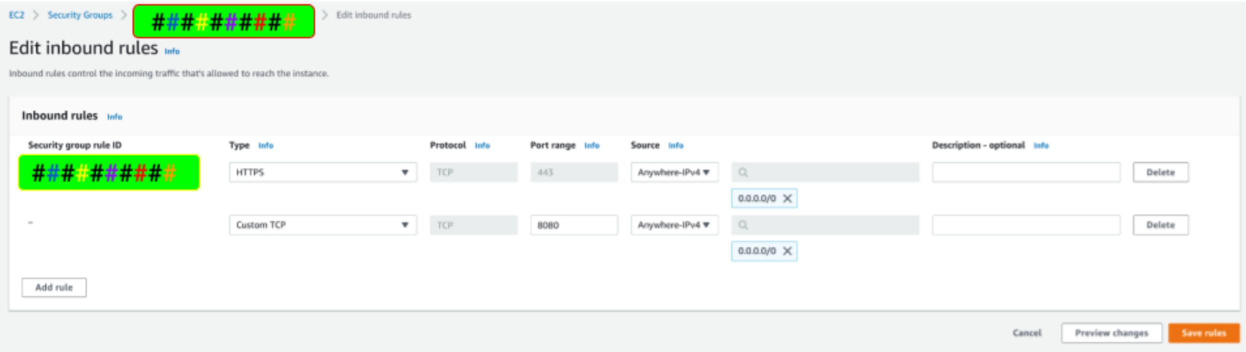
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
[redacted]	HTTPS	TCP	443	Custom 0.0.0.0/0		Delete

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
```



EC2 > Security Groups > [redacted] > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
[redacted]	HTTPS	TCP	443	Anywhere-IPv4	0.0.0.0/0		Delete
-	Custom TCP	TCP	8080	Anywhere-IPv4	0.0.0.0/0		Delete

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*”.

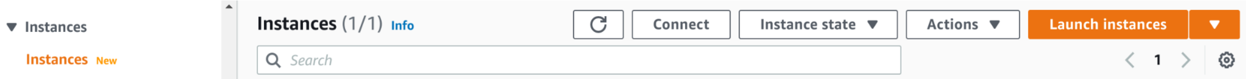
Set **Type** *Custom TCP*, **Protocol** *TCP*, **Port range** *8080*, and **Source** to “*0.0.0.0/0*”.

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.)

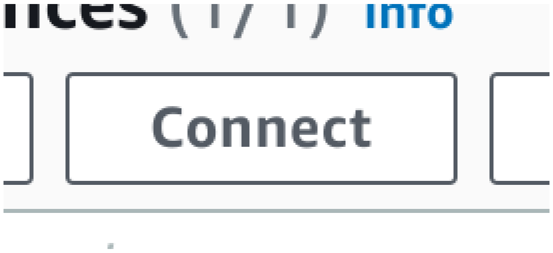
For plotly dash you may need to use 8050 and use this line in your app.run command:

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

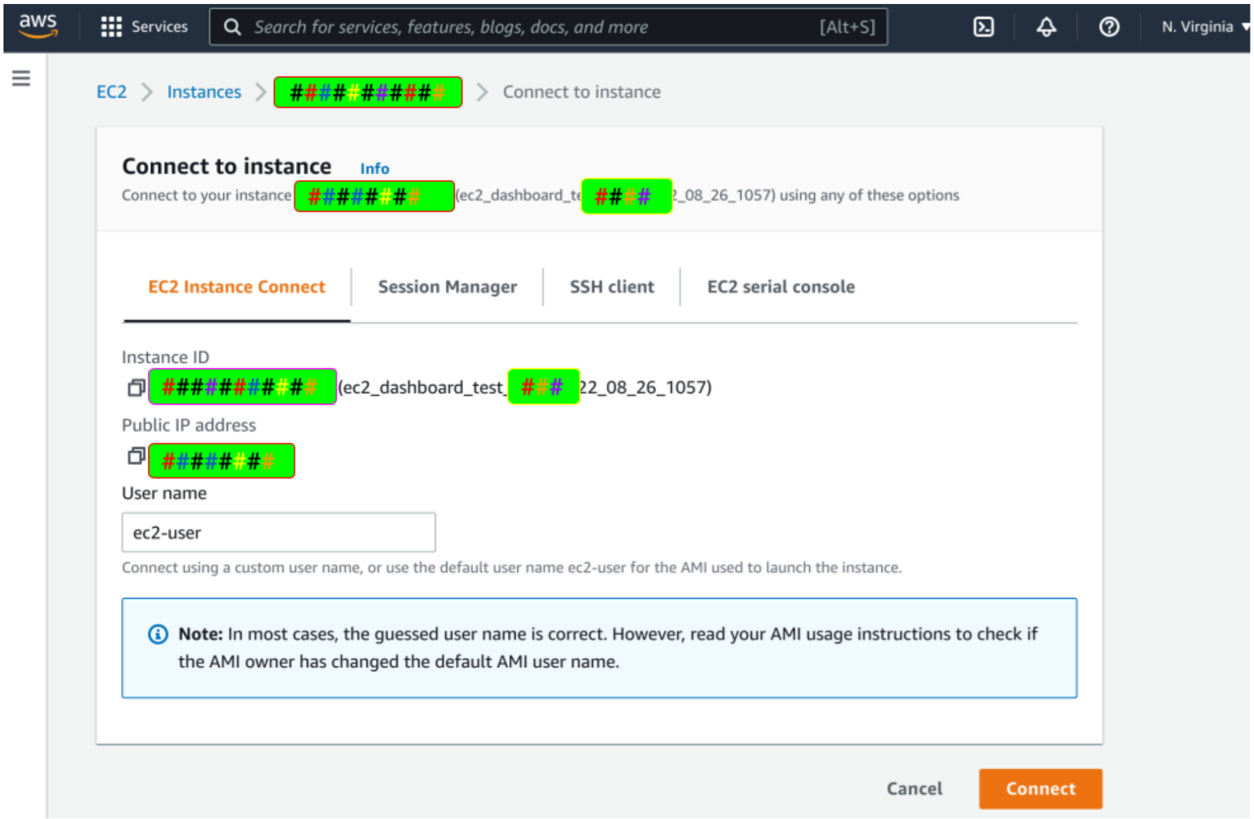
And you may need to add a port suffix after the ipv4URL you get from AWS.
#### In these working examples, plotly-dash's port 8050 was added to the end of the original url.
```
http://3.94.153.137:8050/
or
http://ec2-3-94-153-137.compute-1.amazonaws.com:8050/
```

# Web Connect

## click on "connect"



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



Connect to instance Info

Connect to your instance [redacted] (ec2_dashboard_te [redacted] 2_08_26_1057) using any of these options

Instance ID

[redacted] (ec2_dashboard_test_ [redacted] 22_08_26_1057)

Public IP address

[redacted]

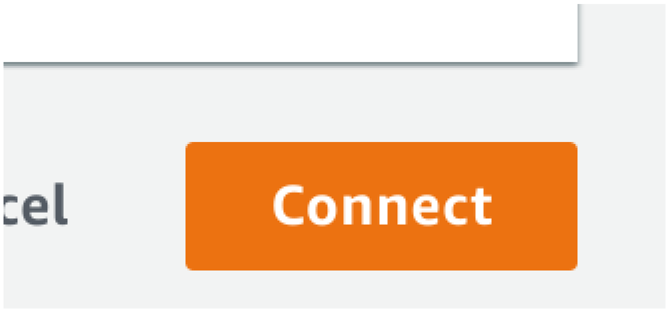
User name

ec2-user

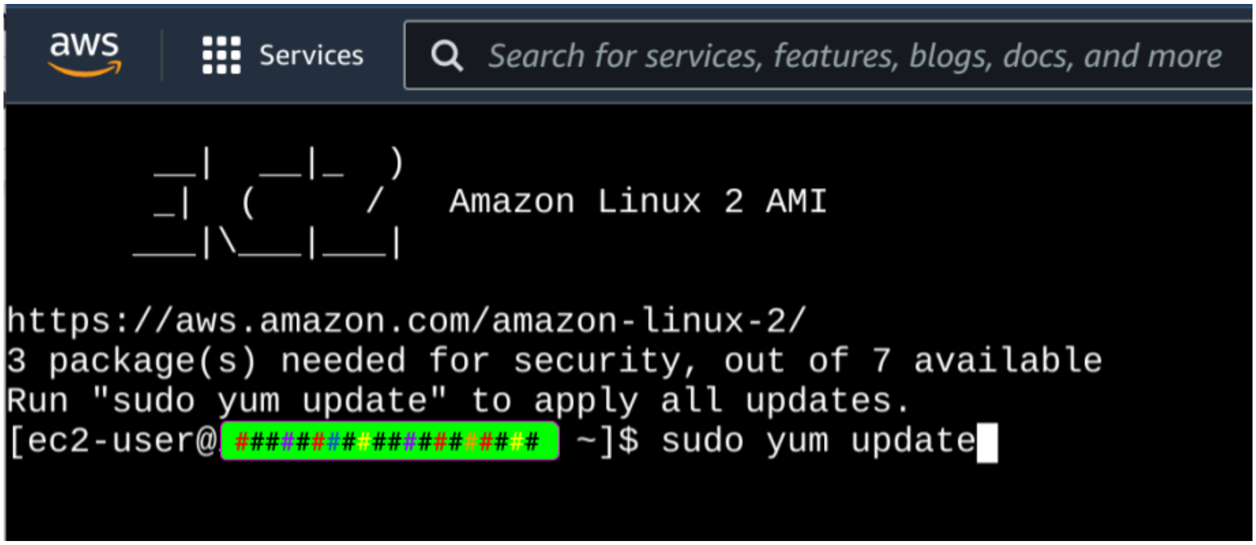
Connect using a custom user name, or use the default user name ec2-user for the AMI used to launch the instance.

Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

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



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



```
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@[redacted] ~]$ sudo yum update
```

### 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

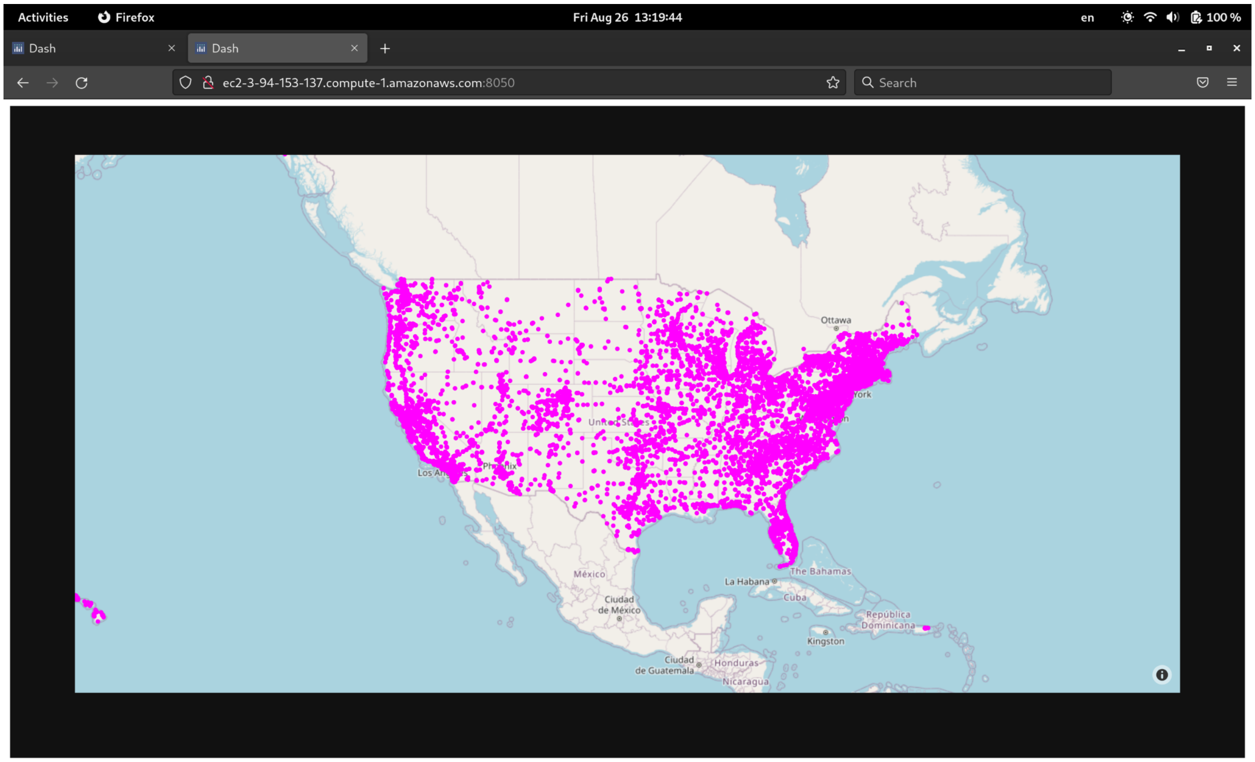
You may need to add a port suffix after the ipv4URL you get from AWS.

#### 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/
```

## Example:

EC2 deployed plotly dash app viewed in browser via public access setup:



# Resources:

- https://stackoverflow.com/questions/67166003/dash-app-not-working-when-deployed-on-amazon-ec2-instance