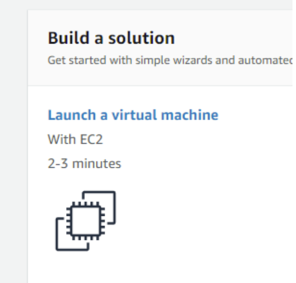
Terraform Labs

Starting and stopping an instance manually

Objective: Deploy an Amazon Linux instance manually. This demonstrates the process is reliable, but rather tedious and not scalable.

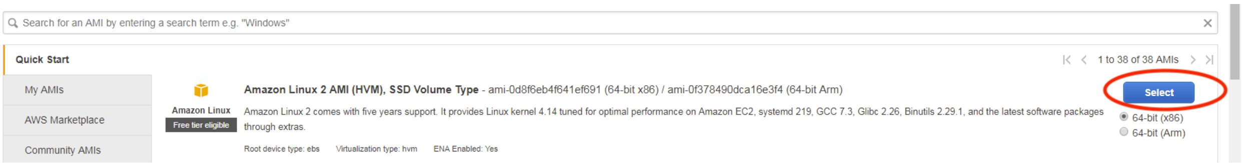
1. Open a Chrome browser window in the WorkSpace
2. Go to “rtt-training.awsapps.com/console”
3. Login with the same userid/password combination used to login to the WorkSpace
4. In the “AWS Management Console” window, click on “Launch a virtual machine”



Build a solution

Get started with simple wizards and automated

5. Select the “Amazon Linux 2” type from the top of the list



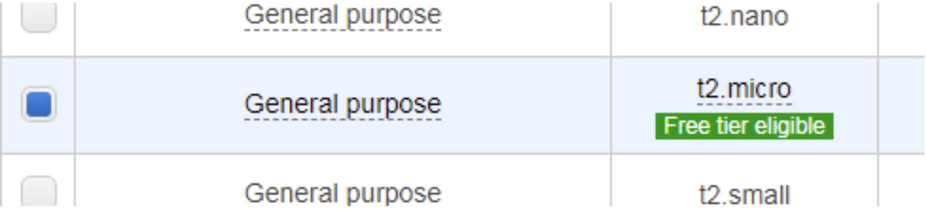
Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0d8f6eb4f641ef691 (64-bit x86) / ami-0f378490dca16e3f4 (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes

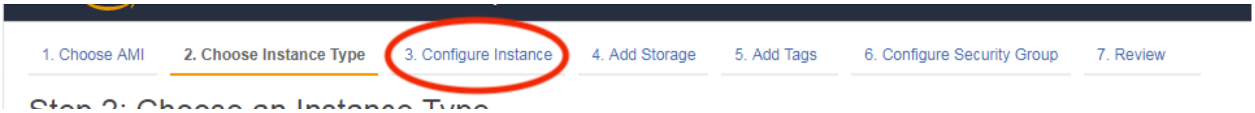
● 64-bit (x86)
○ 64-bit (Arm)

6. Select “t2.micro”

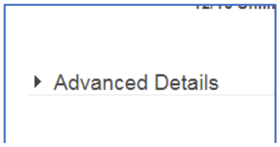


□	General purpose	t2.nano
☑	General purpose	t2.micro Free tier eligible
□	General purpose	t2.small

7. Click on “Configure Instance” link at the top of the page



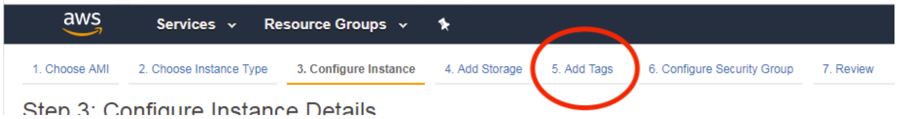
8. Click on “Advanced Details” in the middle of the page



9. Paste the following into the “user data” field.

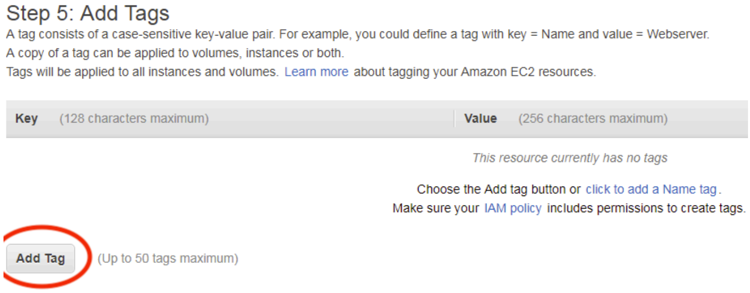
```
#!/bin/bash -xe
yum clean all
yum -y update
yum -y install httpd
apachectl start
echo "<h1>TrainingX: Deployed via Terraform</h1>" | sudo tee /var/www/html/index.html
```

10. Click on the “Add Tags” link at the top of the page



Step 3: Configure Instance Details

11. Click on the “Add Tag” button



Step 5: Add Tags

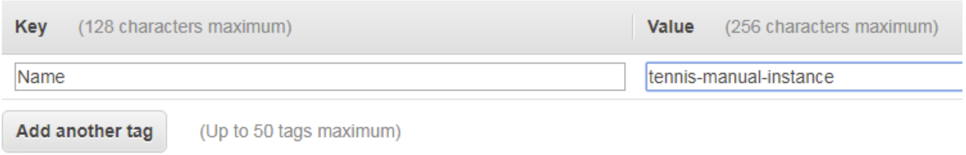
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)

This resource currently has no tags

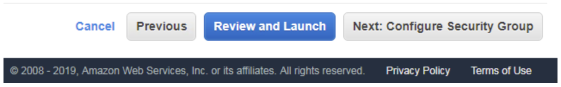
Choose the Add tag button or click to add a Name tag.
Make sure your IAM policy includes permissions to create tags.

12. Set the “Key” field to “Name” and the “Value” field to “<your ID>-manual-instance”



Key (128 characters maximum)	Value (256 characters maximum)
Name	tennis-manual-instance

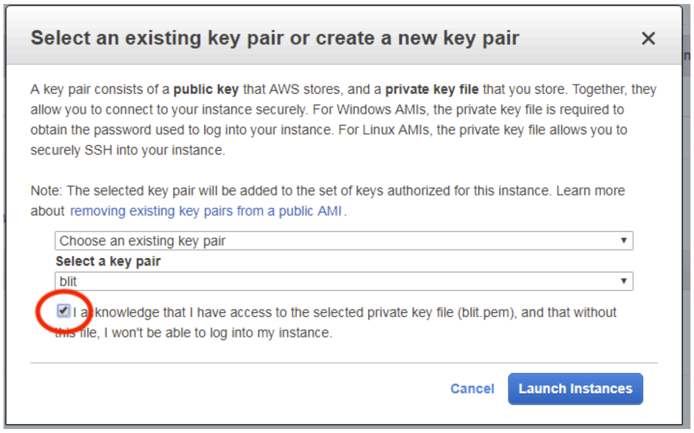
13. Click on “Review and Launch” at the bottom right of the page



© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

14. Click “Launch” on the next page provided.

15. Check the box marked below and click “Launch Instances”



Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

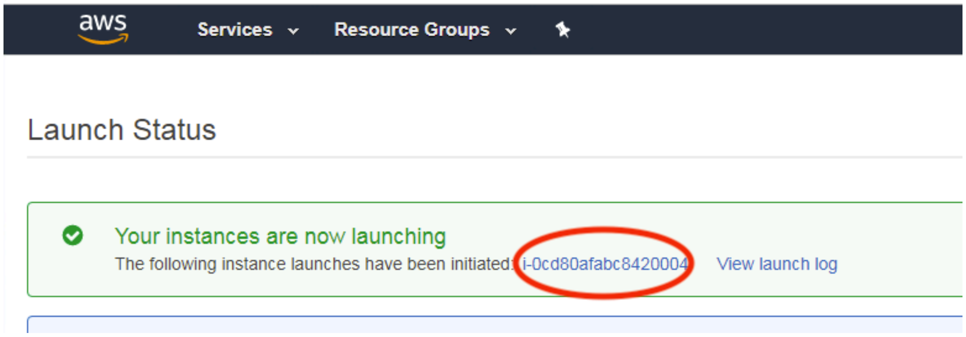
Choose an existing key pair ▾

Select a key pair

blit ▾

☑ I acknowledge that I have access to the selected private key file (blit.pem), and that without this file, I won't be able to log into my instance.

16. On the next page presented, click on the instance number link

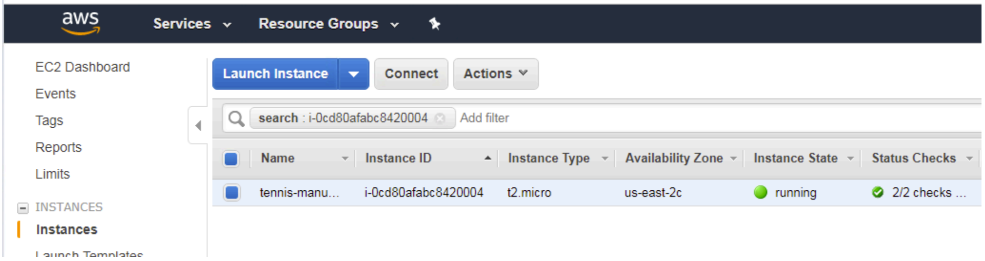


Launch Status

Your instances are now launching

The following instance launches have been initiated: i-0cd80afabc8420004

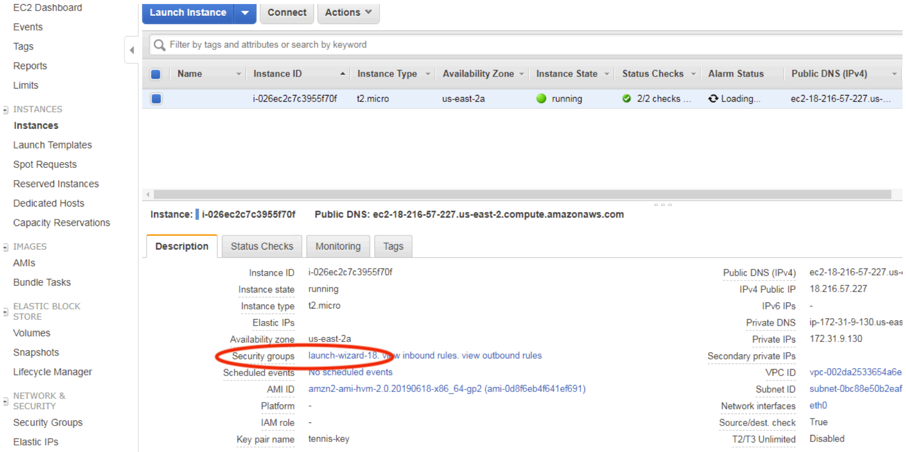
17. This will take you to the ec2 dashboard. You will see your instance running there
18. Wait until the “instance state” is “running” and the “Status Checks” is “2/2 checks”



search : i-0cd80afabc8420004

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
tennis-manu…	i-0cd80afabc8420004	t2.micro	us-east-2c	running	2/2 checks …

19. Look for the “Security groups” label at the bottom of the page



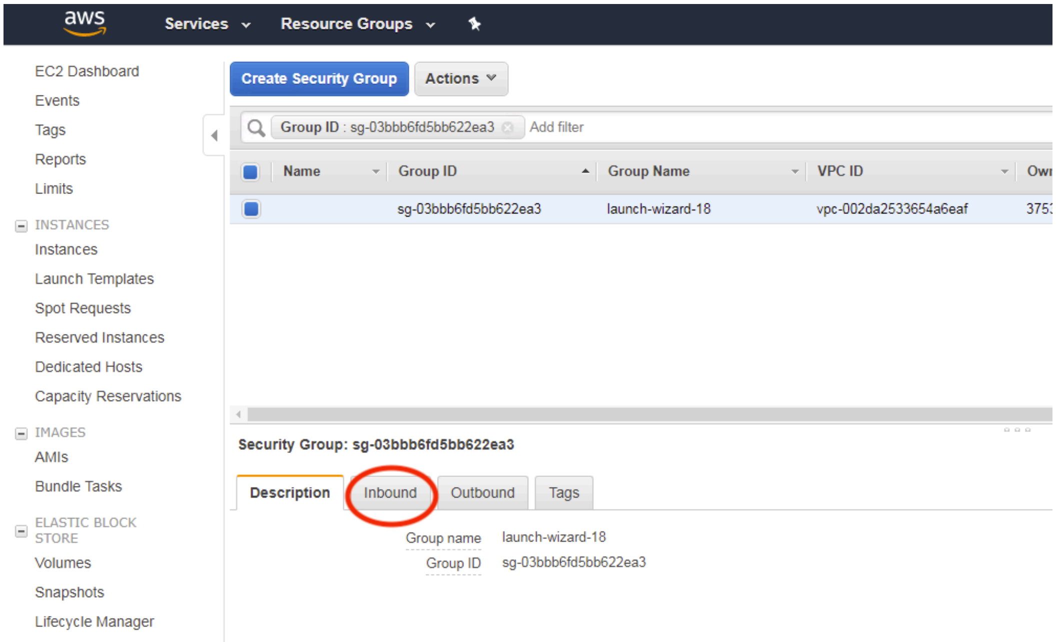
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
	i-026ec2c7c3955f70f	t2.micro	us-east-2a	running	2/2 checks …	Loading...	ec2-18-216-57-227.us-…

Instance: i-026ec2c7c3955f70f Public DNS: ec2-18-216-57-227.us-east-2.compute.amazonaws.com

Instance ID	i-026ec2c7c3955f70f	Public DNS (IPv4)	ec2-18-216-57-227.us-
Instance state	running	IPv4 Public IP	18.216.57.227
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-9-130.us-eas
Availability zone	us-east-2a	Private IPs	172.31.9.130
Security groups	launch-wizard-18. view inbound rules. view outbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-002da2533654a6e
AMI ID	amzn2-ami-hvm-2.0.20190618-x86_64-gp2 (ami-0d8f6eb4f641ef691)	Subnet ID	subnet-0bc88e50b2eaf
Platform	-	Network interfaces	eth0
IAM role	-	Source/dest. check	True
Key pair name	tennis-key	T2/T3 Unlimited	Disabled

20. Click on the link next to “Security groups”. It should be something like “launch-wizard-xx”.

21. You will get a page that looks like this (below). Click on the “Inbound” tab



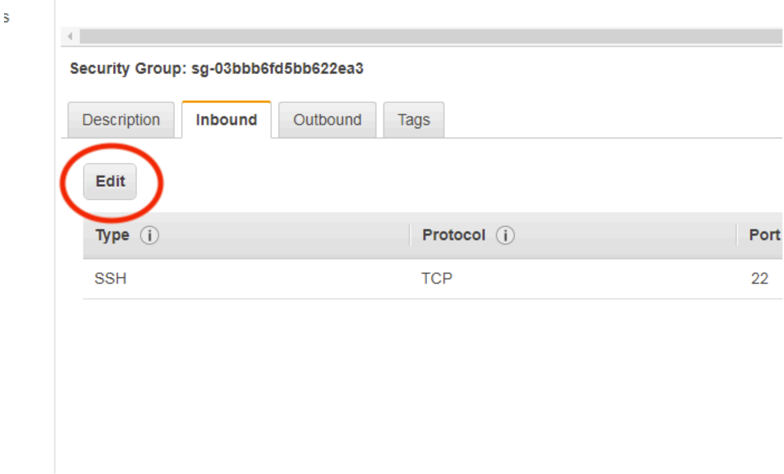
Group ID : sg-03bbb6fd5bb622ea3

Name	Group ID	Group Name	VPC ID	Owr
	sg-03bbb6fd5bb622ea3	launch-wizard-18	vpc-002da2533654a6eaf	3753

Security Group: sg-03bbb6fd5bb622ea3

Group name launch-wizard-18
Group ID sg-03bbb6fd5bb622ea3

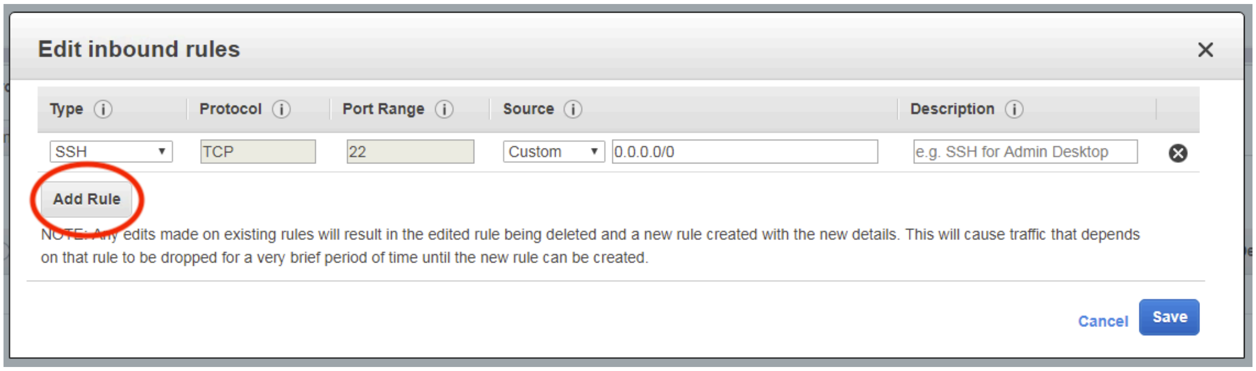
22. Click on the “Edit” button



Security Group: sg-03bbb6fd5bb622ea3

Type	Protocol	Port
SSH	TCP	22

23. Click on the “Add Rule” button in the dialog presented

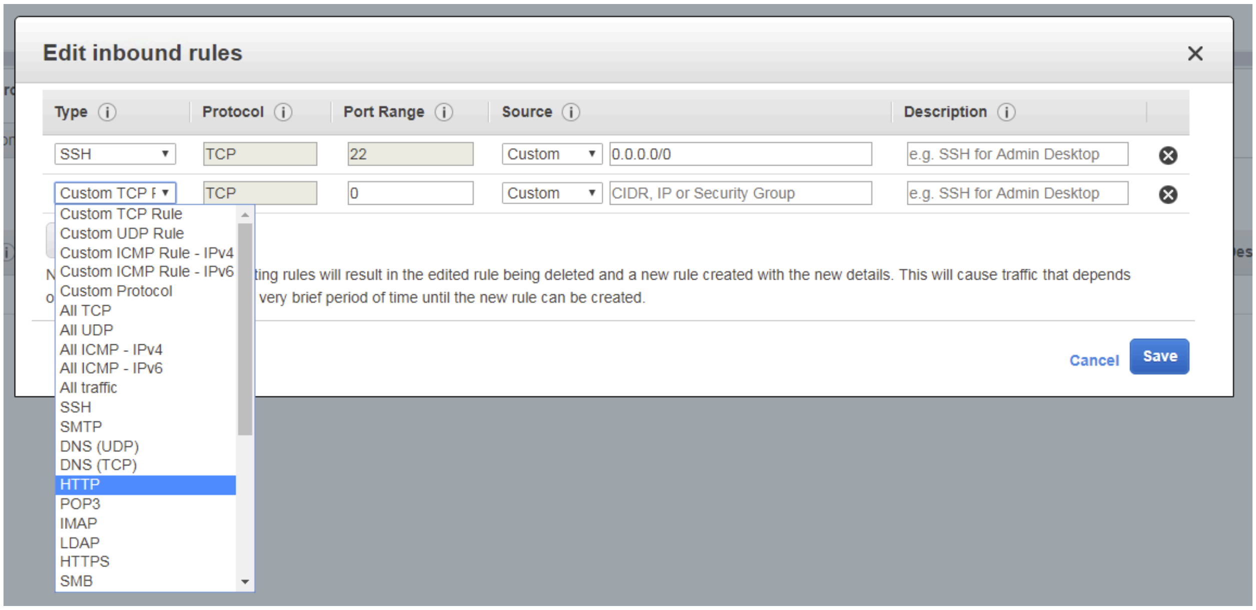


Edit inbound rules

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

24. Select “HTTP” from the fold-down menu of the newly-added field

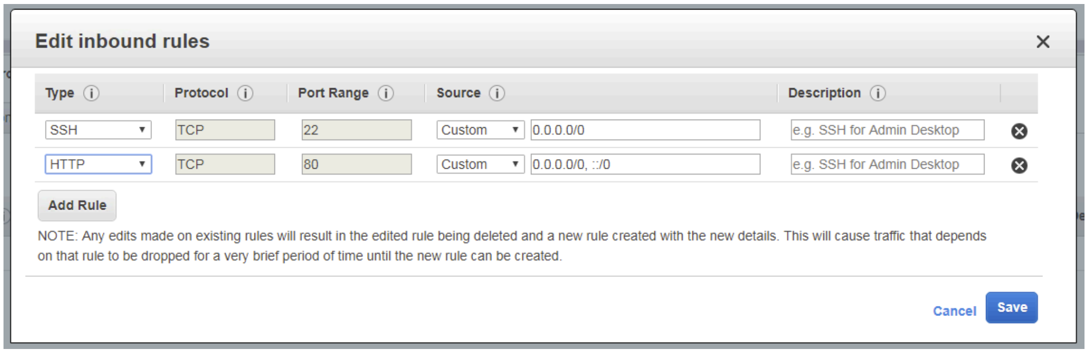


Edit inbound rules

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	0	Custom CIDR, IP or Security Group	e.g. SSH for Admin Desktop

ting rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends very brief period of time until the new rule can be created.

25. Accept the defaults. The dialog box shown should now look like this



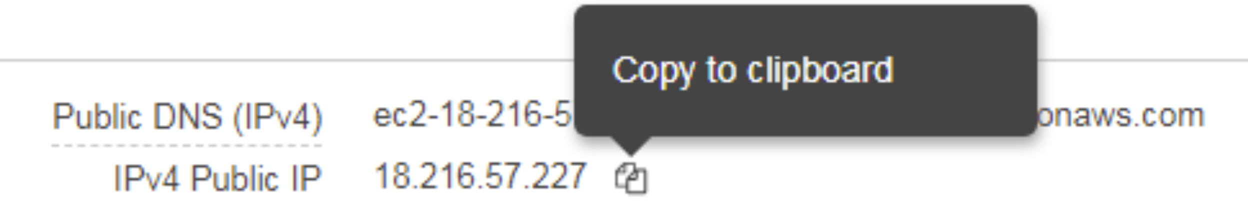
Edit inbound rules

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

26. Click the “Save” button

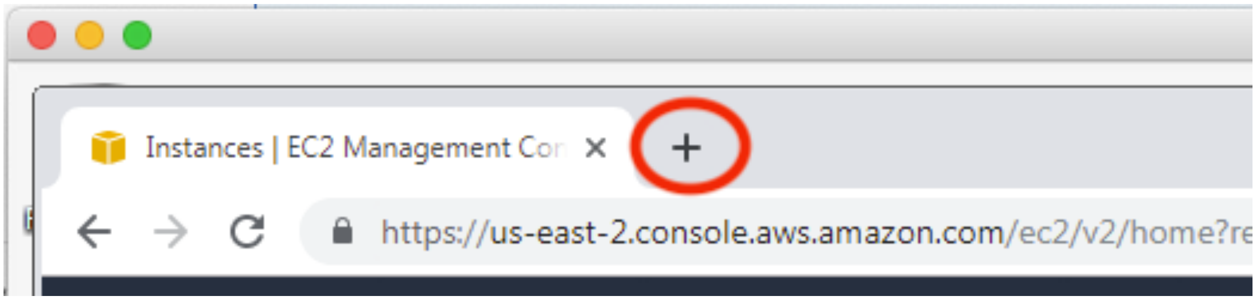
27. Click just to the right of the “IPv4 Public IP” field and you will get a copy of the address



Public DNS (IPv4) ec2-18-216-5 onaws.com

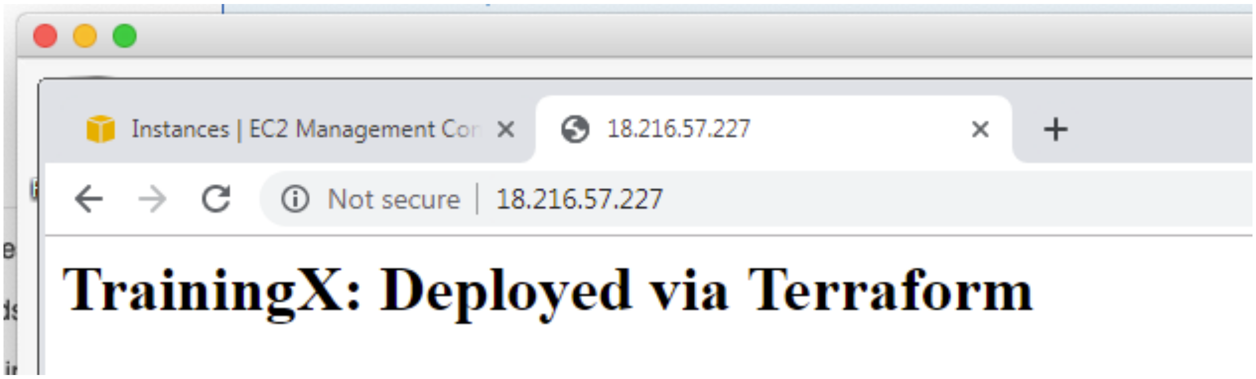
IPv4 Public IP 18.216.57.227

28. Click on the + sign in your browser and open a new tab



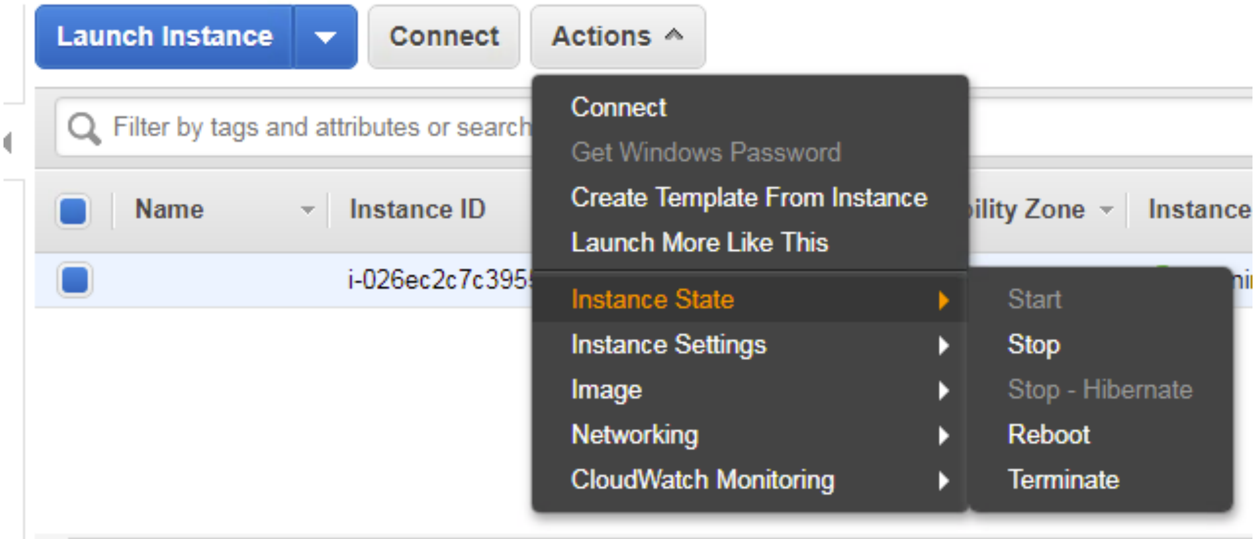
Instances | EC2 Management Co
+
https://us-east-2.console.aws.amazon.com/ec2/v2/home?re

29. In that new tab, paste the address you just copied, and press enter
30. You should see something like this

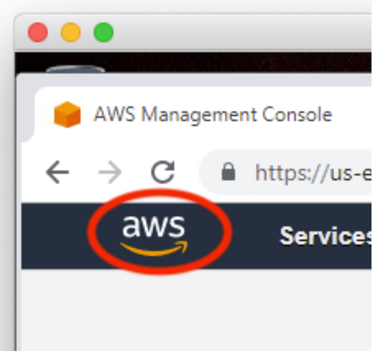


TrainingX: Deployed via Terraform

31. Close the tab
32. Go back to the previous tab and select your running instance
33. Click on the “Actions”->”Instance State”->”Terminate” menu item



34. On your browser, click on the AWS logo in the upper left. This will take you back to the main management console screen



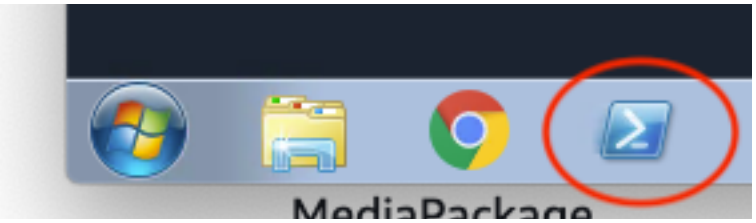
35. Repeat steps 4 through 34. This time change the string “TrainingX” in step 9 to match your Training ID (ex. “Training99”).

Terraform Labs

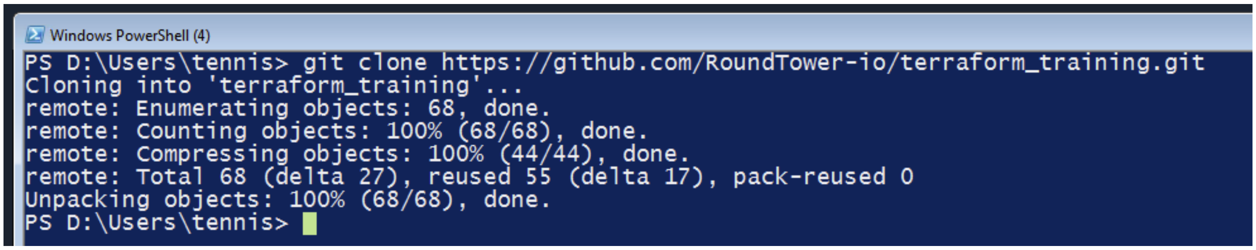
Starting and stopping an instance with Terraform

Objective: Demonstrate how much easier it is to create/destroy instances using Terraform. This is an example of Infrastructure as Code (IaC).

1. Click on the command prompt icon at the bottom left of the WorkSpace screen

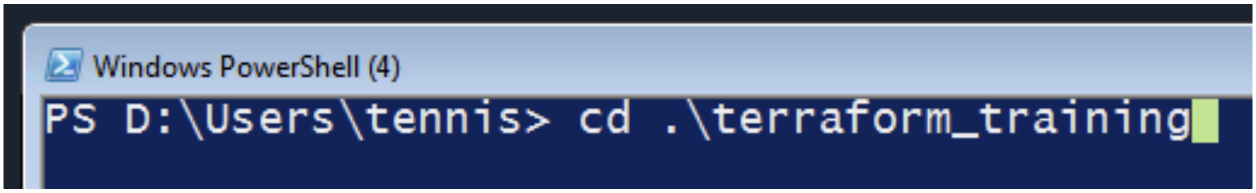


2. Clone a copy of https://github.com/RoundTower-io/terraform_training.git



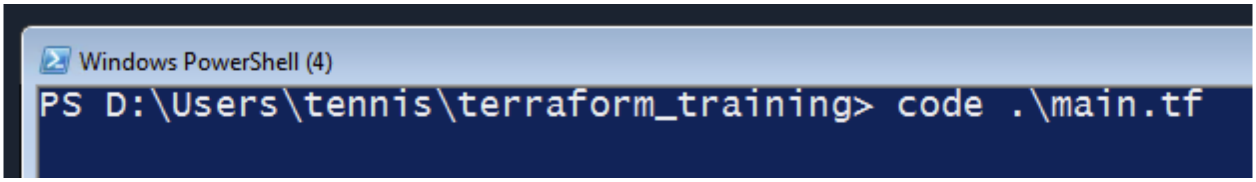
```
PS D:\Users\tennis> git clone https://github.com/RoundTower-io/terraform_training.git
Cloning into 'terraform_training'...
remote: Enumerating objects: 68, done.
remote: Counting objects: 100% (68/68), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 68 (delta 27), reused 55 (delta 17), pack-reused 0
Unpacking objects: 100% (68/68), done.
PS D:\Users\tennis>
```

3. Change directory into “terraform_training”



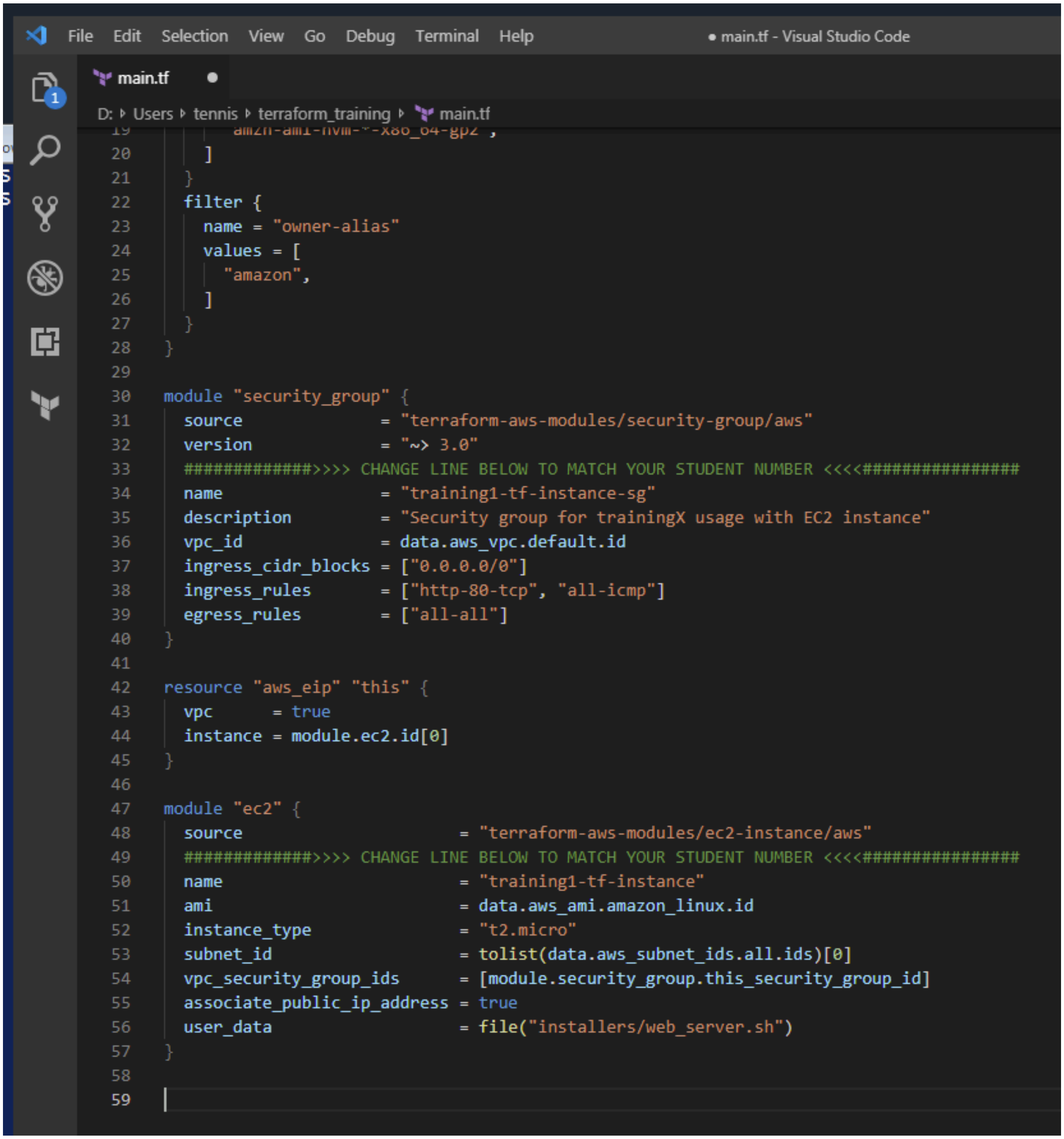
```
PS D:\Users\tennis> cd .\terraform_training
```

4. Edit the file “main.tf”



```
PS D:\Users\tennis\terraform_training> code .\main.tf
```

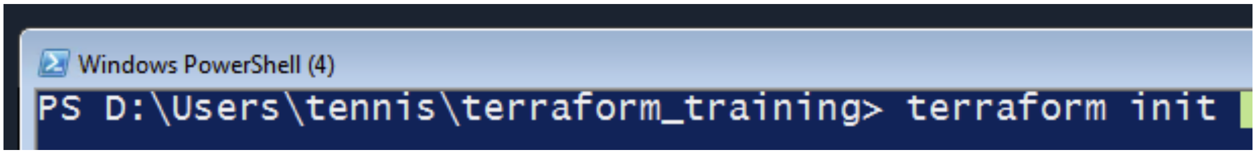
5. Scroll to the bottom of the file and change the values directed by the instructions in the file


File
Edit
Selection
View
Go
Debug
Terminal
Help
● main.tf - Visual Studio Code
main.tf
D: › Users › tennis › terraform_training › main.tf

```
19       "amzn-ami-hvm-*-x86_64-gp2",
20     ]
21   }
22   filter {
23     name = "owner-alias"
24     values = [
25       "amazon",
26     ]
27   }
28 }
29
30 module "security_group" {
31   source              = "terraform-aws-modules/security-group/aws"
32   version             = "~> 3.0"
33   #############>>>> CHANGE LINE BELOW TO MATCH YOUR STUDENT NUMBER <<<<################
34   name                = "training1-tf-instance-sg"
35   description         = "Security group for trainingX usage with EC2 instance"
36   vpc_id              = data.aws_vpc.default.id
37   ingress_cidr_blocks = ["0.0.0.0/0"]
38   ingress_rules       = ["http-80-tcp", "all-icmp"]
39   egress_rules        = ["all-all"]
40 }
41
42 resource "aws_eip" "this" {
43   vpc      = true
44   instance = module.ec2.id[0]
45 }
46
47 module "ec2" {
48   source                      = "terraform-aws-modules/ec2-instance/aws"
49   #############>>>> CHANGE LINE BELOW TO MATCH YOUR STUDENT NUMBER <<<<################
50   name                        = "training1-tf-instance"
51   ami                         = data.aws_ami.amazon_linux.id
52   instance_type               = "t2.micro"
53   subnet_id                   = tolist(data.aws_subnet_ids.all.ids)[0]
54   vpc_security_group_ids      = [module.security_group.this_security_group_id]
55   associate_public_ip_address = true
56   user_data                   = file("installers/web_server.sh")
57 }
58
59
```

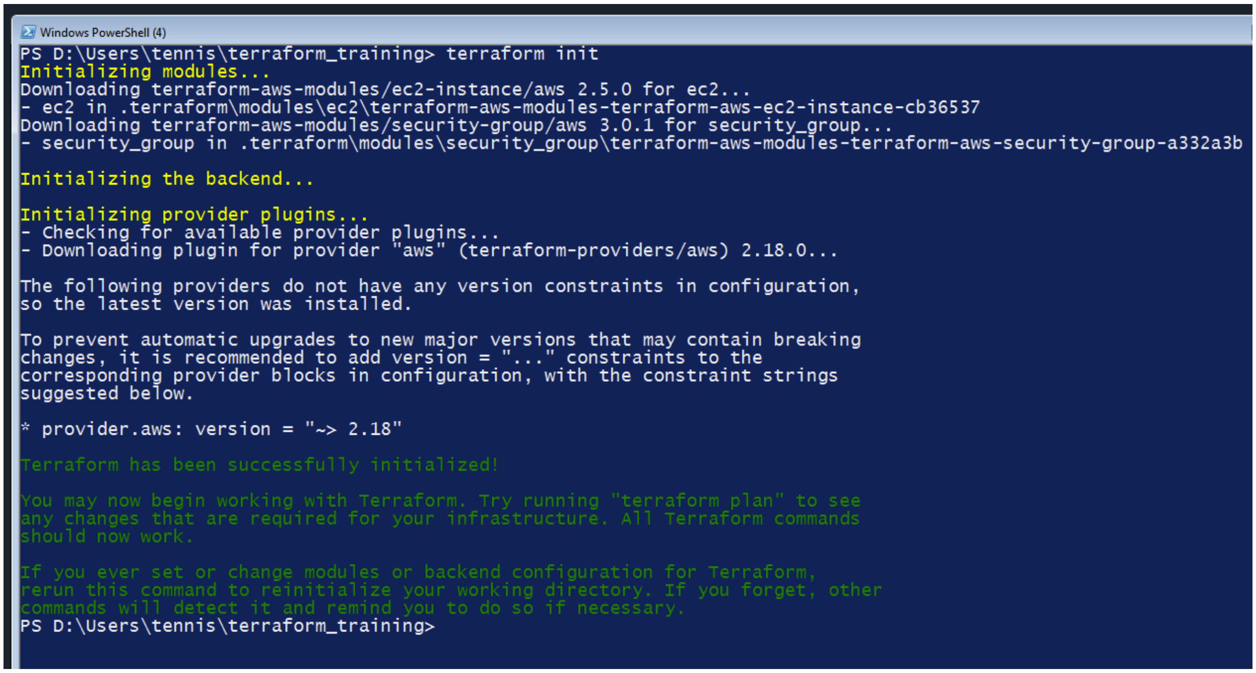
6. Take note that value on line 50 will be the name of your ec2 instance. For example, if your training ID is “3”, your ec2 instance will be named “training3-tf-instance”
7. Make sure you set the training ID’s correctly.
8. Save and exit the file

9. Initialize terraform by typing “terraform init”



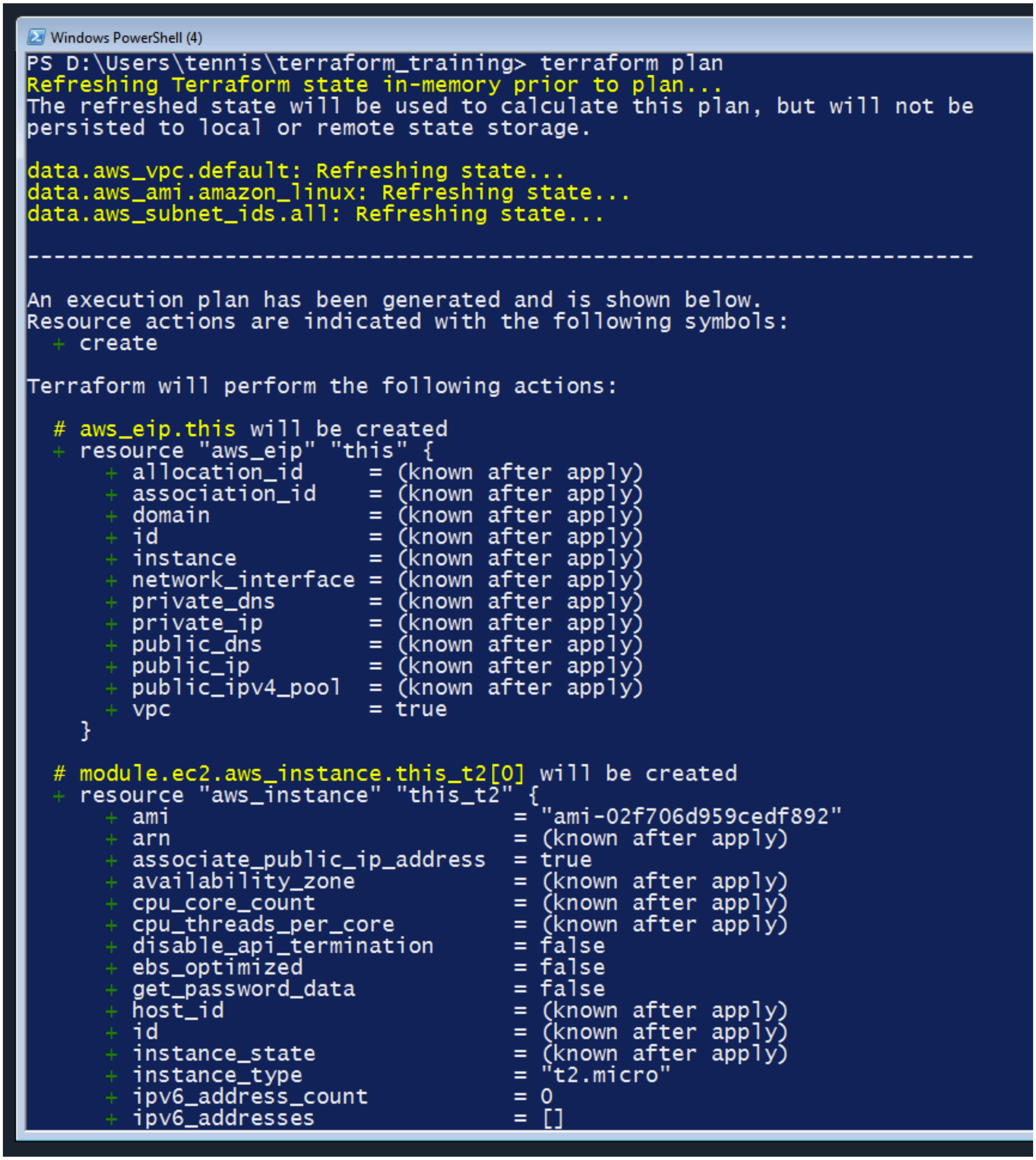
```
PS D:\Users\tennis\terraform_training> terraform init
```

10. The output should look something like this



```
PS D:\Users\tennis\terraform_training> terraform init
Initializing modules...
Downloading terraform-aws-modules/ec2-instance/aws 2.5.0 for ec2...
- ec2 in .terraform\modules\ec2\terraform-aws-modules-terraform-aws-ec2-instance-cb36537
Downloading terraform-aws-modules/security-group/aws 3.0.1 for security_group...
- security_group in .terraform\modules\security_group\terraform-aws-modules-terraform-aws-security-group-a332a3b

Initializing the backend...

Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "aws" (terraform-providers/aws) 2.18.0...

The following providers do not have any version constraints in configuration,
so the latest version was installed.

To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.

* provider.aws: version = "~> 2.18"

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS D:\Users\tennis\terraform_training>
```

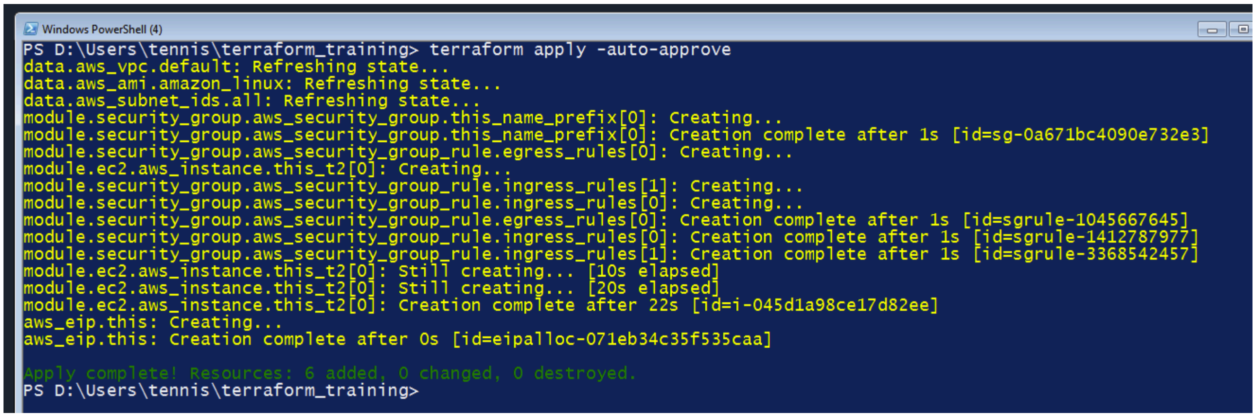
11. Now, we will test that our terraform script will run correctly. Type in “terraform plan”



```
PS D:\Users\tennis\terraform_training> terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.

data.aws_vpc.default: Refreshing state...
data.aws_ami.amazon_linux: Refreshing state...
data.aws_subnet_ids.all: Refreshing state...

------------------------------------------------------------------------

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_eip.this will be created
  + resource "aws_eip" "this" {
      + allocation_id     = (known after apply)
      + association_id    = (known after apply)
      + domain            = (known after apply)
      + id                = (known after apply)
      + instance          = (known after apply)
      + network_interface = (known after apply)
      + private_dns       = (known after apply)
      + private_ip        = (known after apply)
      + public_dns        = (known after apply)
      + public_ip         = (known after apply)
      + public_ipv4_pool  = (known after apply)
      + vpc               = true
    }

  # module.ec2.aws_instance.this_t2[0] will be created
  + resource "aws_instance" "this_t2" {
      + ami                          = "ami-02f706d959cedf892"
      + arn                          = (known after apply)
      + associate_public_ip_address  = true
      + availability_zone            = (known after apply)
      + cpu_core_count               = (known after apply)
      + cpu_threads_per_core         = (known after apply)
      + disable_api_termination      = false
      + ebs_optimized                = false
      + get_password_data            = false
      + host_id                      = (known after apply)
      + id                           = (known after apply)
      + instance_state               = (known after apply)
      + instance_type                = "t2.micro"
      + ipv6_address_count           = 0
      + ipv6_addresses               = []
```

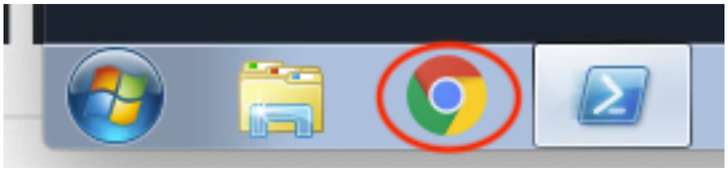
12. If we don’t get any errors, we will now try to run the script

13. Type in “terraform apply -auto-approve” and hit enter. The output should look something like this

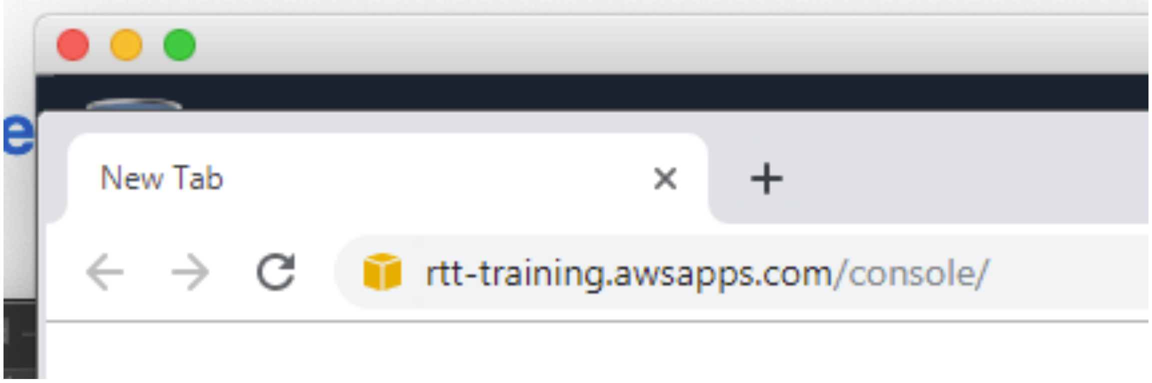


```
PS D:\Users\tennis\terraform_training> terraform apply -auto-approve
data.aws_vpc.default: Refreshing state...
data.aws_ami.amazon_linux: Refreshing state...
data.aws_subnet_ids.all: Refreshing state...
module.security_group.aws_security_group.this_name_prefix[0]: Creating...
module.security_group.aws_security_group.this_name_prefix[0]: Creation complete after 1s [id=sg-0a671bc4090e732e3]
module.security_group.aws_security_group_rule.egress_rules[0]: Creating...
module.ec2.aws_instance.this_t2[0]: Creating...
module.security_group.aws_security_group_rule.ingress_rules[1]: Creating...
module.security_group.aws_security_group_rule.ingress_rules[0]: Creating...
module.security_group.aws_security_group_rule.egress_rules[0]: Creation complete after 1s [id=sgrule-1045667645]
module.security_group.aws_security_group_rule.ingress_rules[0]: Creation complete after 1s [id=sgrule-1412787977]
module.security_group.aws_security_group_rule.ingress_rules[1]: Creation complete after 1s [id=sgrule-3368542457]
module.ec2.aws_instance.this_t2[0]: Still creating... [10s elapsed]
module.ec2.aws_instance.this_t2[0]: Still creating... [20s elapsed]
module.ec2.aws_instance.this_t2[0]: Creation complete after 22s [id=i-045d1a98ce17d82ee]
aws_eip.this: Creating...
aws_eip.this: Creation complete after 0s [id=eipalloc-071eb34c35f535caa]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.
PS D:\Users\tennis\terraform_training>
```

14. Open a browser in the WorkSpace

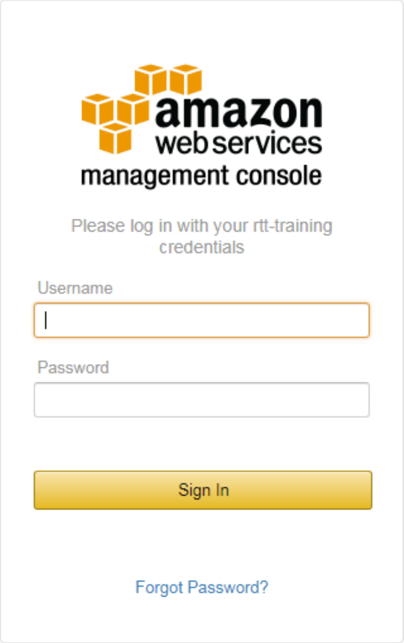


15. Go to this url: http://rtt-training.awsapps.com/console/



rtt-training.awsapps.com/console/

16. Login using the same user ID/password combination you used to login to the WorkSpace

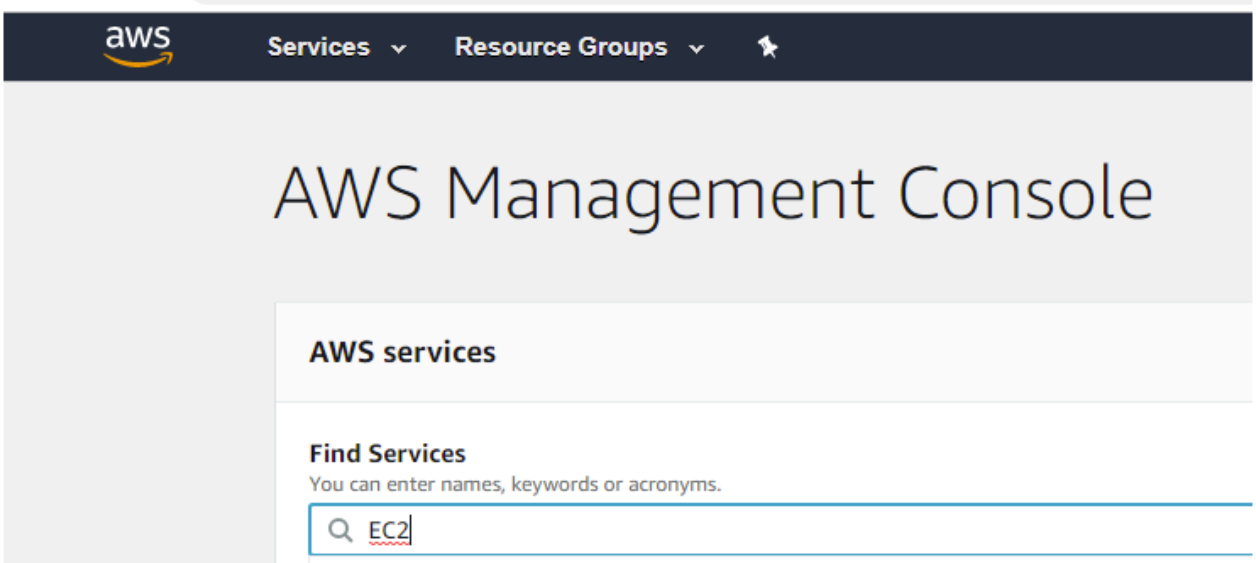


Please log in with your rtt-training credentials

Username:

Password:

17. This will take you to the main "AWS Management Console". Type in "EC2" in the "Find Services" search box and hit enter



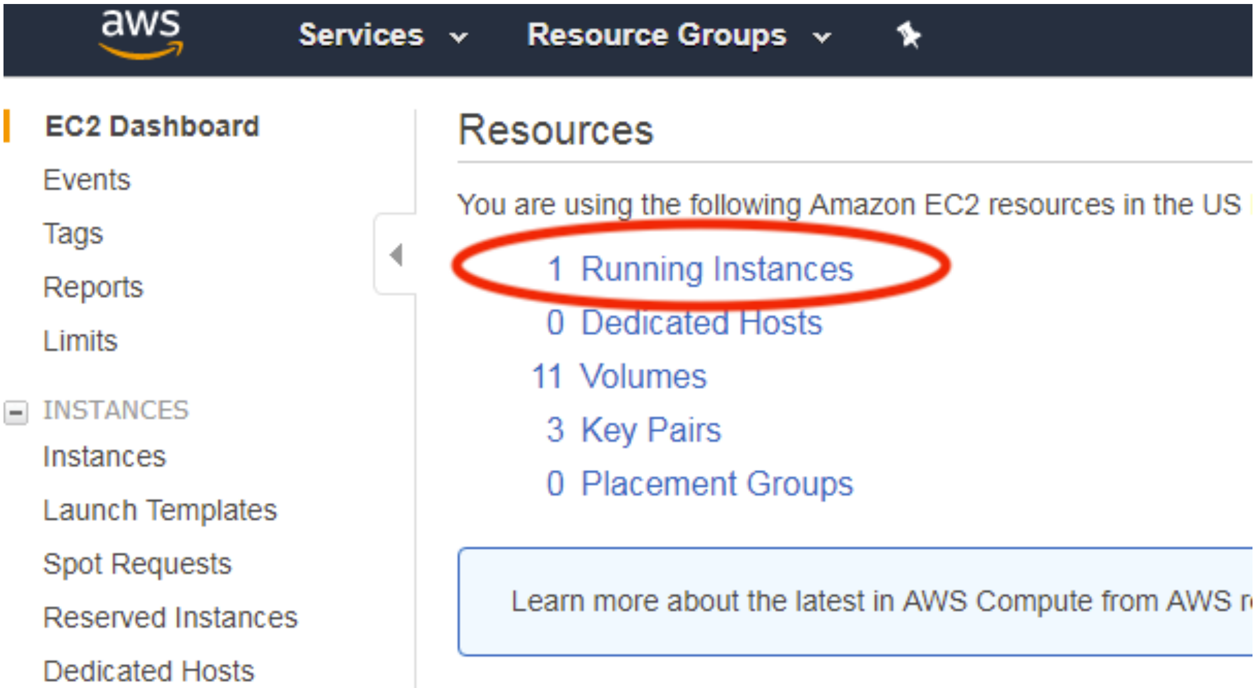
Find Services

You can enter names, keywords or acronyms.

EC2

18. This will take you to the main EC2 console.

19. Click on the “Running Instances” link



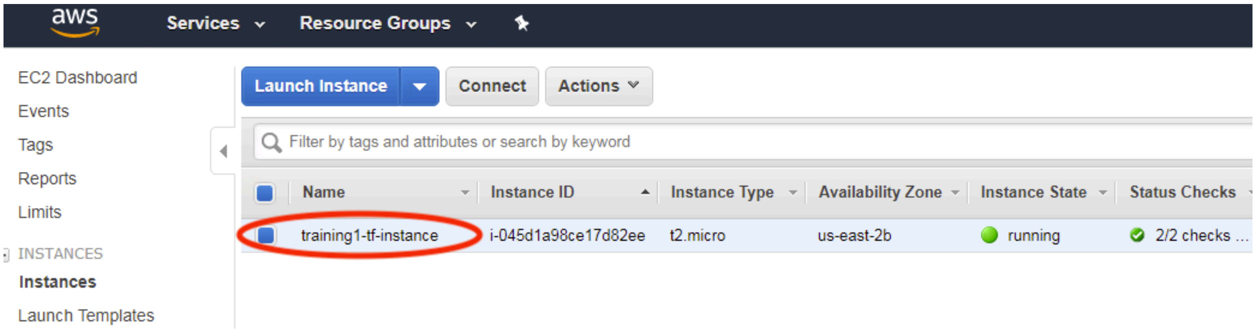
Resources

You are using the following Amazon EC2 resources in the US

1 Running Instances
0 Dedicated Hosts
11 Volumes
3 Key Pairs
0 Placement Groups

Learn more about the latest in AWS Compute from AWS r

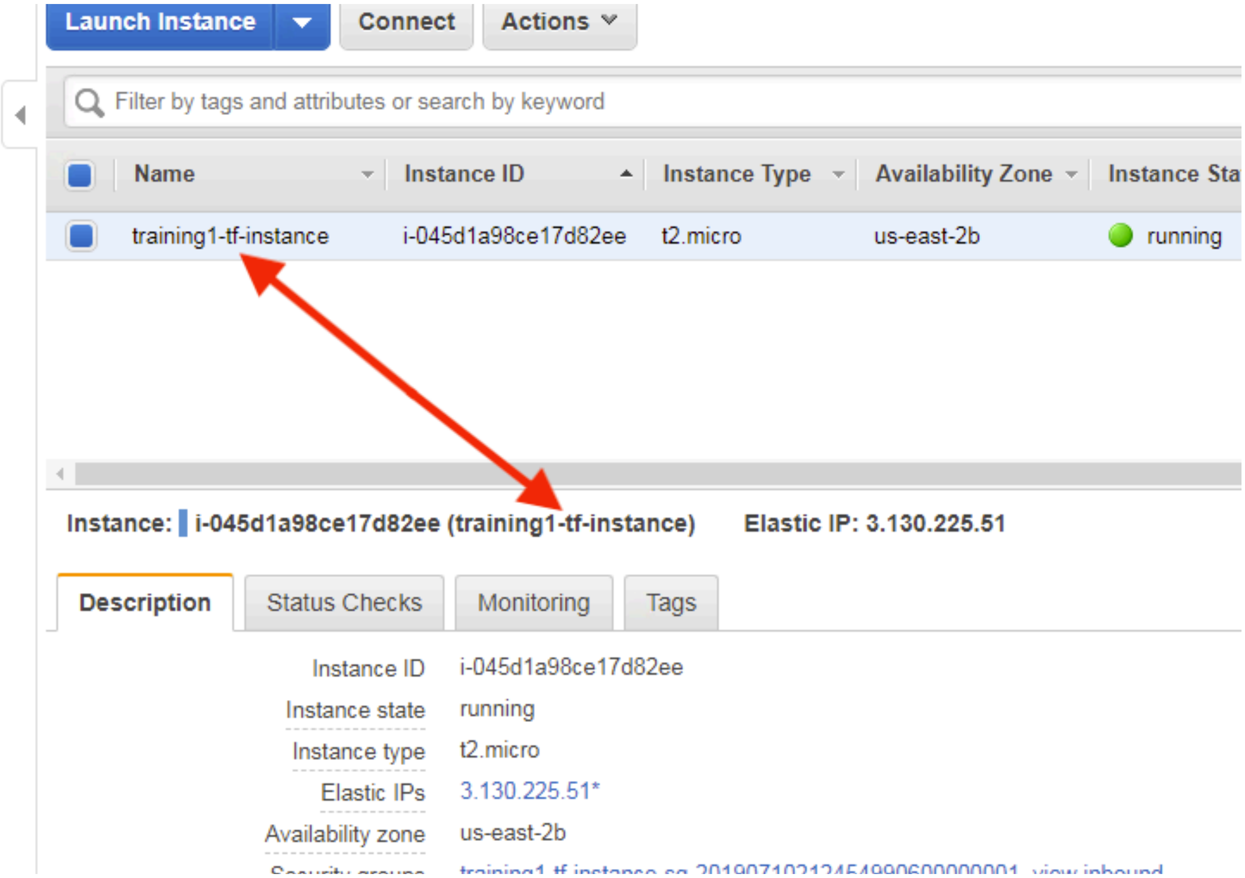
20. In the list of running instances, you should see your own



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
training1-tf-instance	i-045d1a98ce17d82ee	t2.micro	us-east-2b	● running	✓ 2/2 checks ...

21. Select your instance by clicking on the radio button next to the name (above)
22. Wait until the “instance state” is “running” and the “Status Checks” is “2/2 checks”

23. Below the selected instance, you should now see a pane with lots of details about your instance

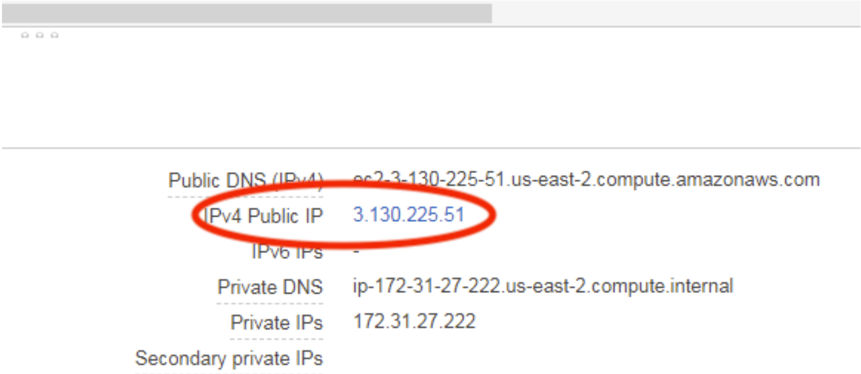


Name	Instance ID	Instance Type	Availability Zone	Instance Sta
training1-tf-instance	i-045d1a98ce17d82ee	t2.micro	us-east-2b	● running

Instance: i-045d1a98ce17d82ee (training1-tf-instance)   Elastic IP: 3.130.225.51

Instance ID i-045d1a98ce17d82ee
Instance state running
Instance type t2.micro
Elastic IPs 3.130.225.51*
Availability zone us-east-2b
Security groups training1-tf-instance-sg-20190710212454990600000001. view inbound

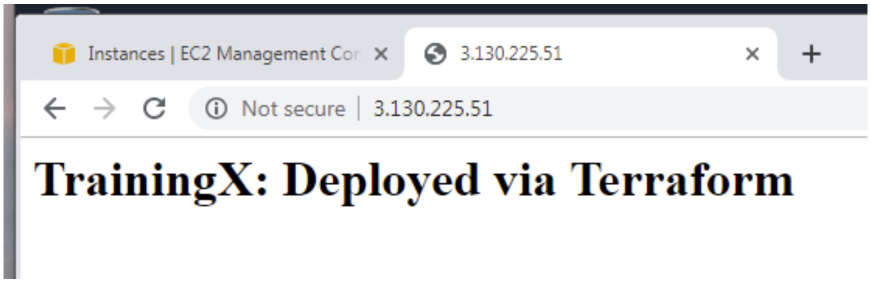
24. Look for the "IPv4 Public IP" on the right and make a copy of it



Public DNS (IPv4) ec2-3-130-225-51.us-east-2.compute.amazonaws.com
IPv4 Public IP 3.130.225.51
IPv6 IPs -
Private DNS ip-172-31-27-222.us-east-2.compute.internal
Private IPs 172.31.27.222
Secondary private IPs

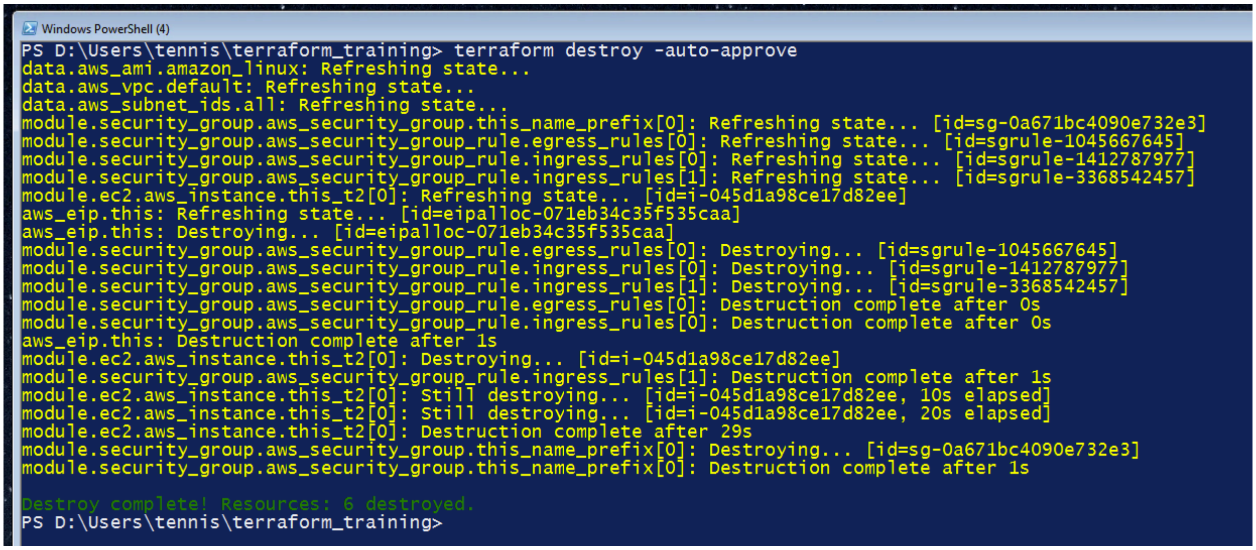
25. Open a new tab in your browser
26. Paste the ip address on the address line of the browser

27. You should see something like this



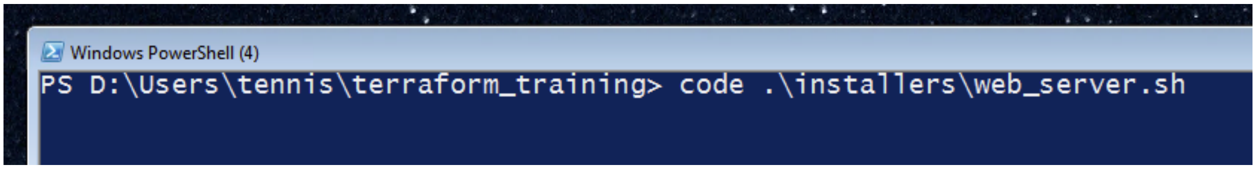
TrainingX: Deployed via Terraform

28. The website works, but the training number is wrong. Lets fix that.
29. Go back to your command line window and type “terraform destroy -auto-approve”



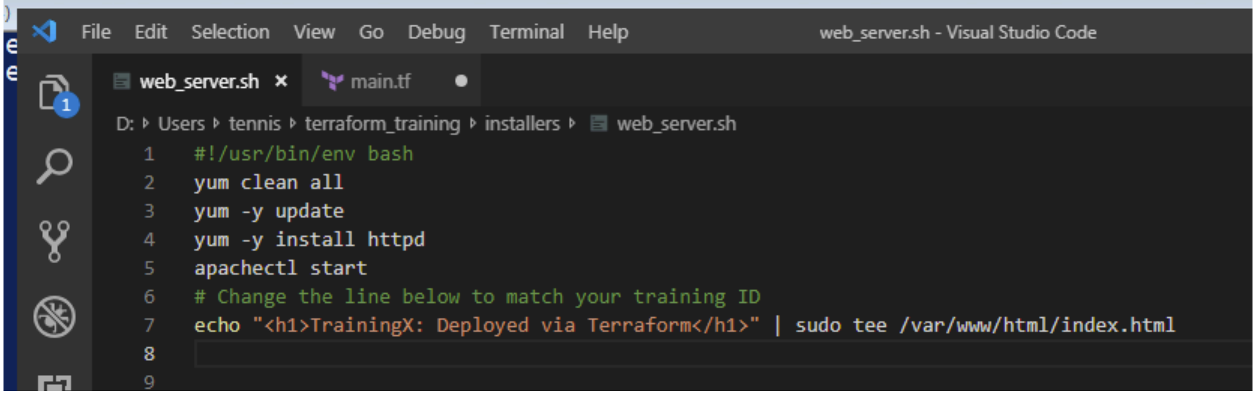
```
PS D:\Users\tennis\terraform_training> terraform destroy -auto-approve
data.aws_ami.amazon_linux: Refreshing state...
data.aws_vpc.default: Refreshing state...
data.aws_subnet_ids.all: Refreshing state...
module.security_group.aws_security_group.this_name_prefix[0]: Refreshing state... [id=sg-0a671bc4090e732e3]
module.security_group.aws_security_group_rule.egress_rules[0]: Refreshing state... [id=sgrule-1045667645]
module.security_group.aws_security_group_rule.ingress_rules[0]: Refreshing state... [id=sgrule-1412787977]
module.security_group.aws_security_group_rule.ingress_rules[1]: Refreshing state... [id=sgrule-3368542457]
module.ec2.aws_instance.this_t2[0]: Refreshing state... [id=i-045d1a98ce17d82ee]
aws_eip.this: Refreshing state... [id=eipalloc-071eb34c35f535caa]
aws_eip.this: Destroying... [id=eipalloc-071eb34c35f535caa]
module.security_group.aws_security_group_rule.egress_rules[0]: Destroying... [id=sgrule-1045667645]
module.security_group.aws_security_group_rule.ingress_rules[0]: Destroying... [id=sgrule-1412787977]
module.security_group.aws_security_group_rule.ingress_rules[1]: Destroying... [id=sgrule-3368542457]
module.security_group.aws_security_group_rule.egress_rules[0]: Destruction complete after 0s
module.security_group.aws_security_group_rule.ingress_rules[0]: Destruction complete after 0s
aws_eip.this: Destruction complete after 1s
module.ec2.aws_instance.this_t2[0]: Destroying... [id=i-045d1a98ce17d82ee]
module.security_group.aws_security_group_rule.ingress_rules[1]: Destruction complete after 1s
module.ec2.aws_instance.this_t2[0]: Still destroying... [id=i-045d1a98ce17d82ee, 10s elapsed]
module.ec2.aws_instance.this_t2[0]: Still destroying... [id=i-045d1a98ce17d82ee, 20s elapsed]
module.ec2.aws_instance.this_t2[0]: Destruction complete after 29s
module.security_group.aws_security_group.this_name_prefix[0]: Destroying... [id=sg-0a671bc4090e732e3]
module.security_group.aws_security_group.this_name_prefix[0]: Destruction complete after 1s

Destroy complete! Resources: 6 destroyed.
PS D:\Users\tennis\terraform_training>
```

30. Now, edit the file “installers/web_server.sh”



```
PS D:\Users\tennis\terraform_training> code .\installers\web_server.sh
```

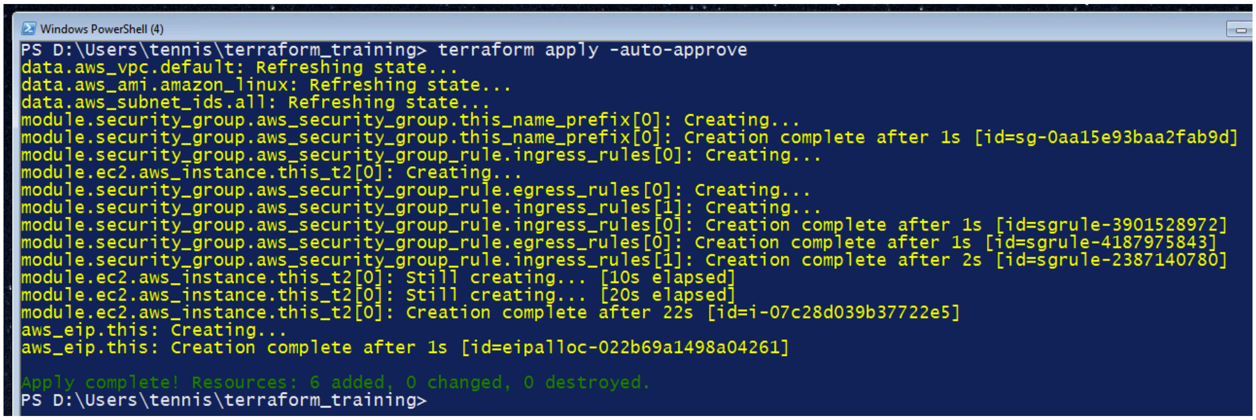
31. Change the training ID value on line 7 to your training ID number


File
Edit
Selection
View
Go
Debug
Terminal
Help
web_server.sh - Visual Studio Code
web_server.sh
main.tf
D: › Users › tennis › terraform_training › installers › web_server.sh

```
1  #!/usr/bin/env bash
2  yum clean all
3  yum -y update
4  yum -y install httpd
5  apachectl start
6  # Change the line below to match your training ID
7  echo "<h1>TrainingX: Deployed via Terraform</h1>" | sudo tee /var/www/html/index.html
8
9
```

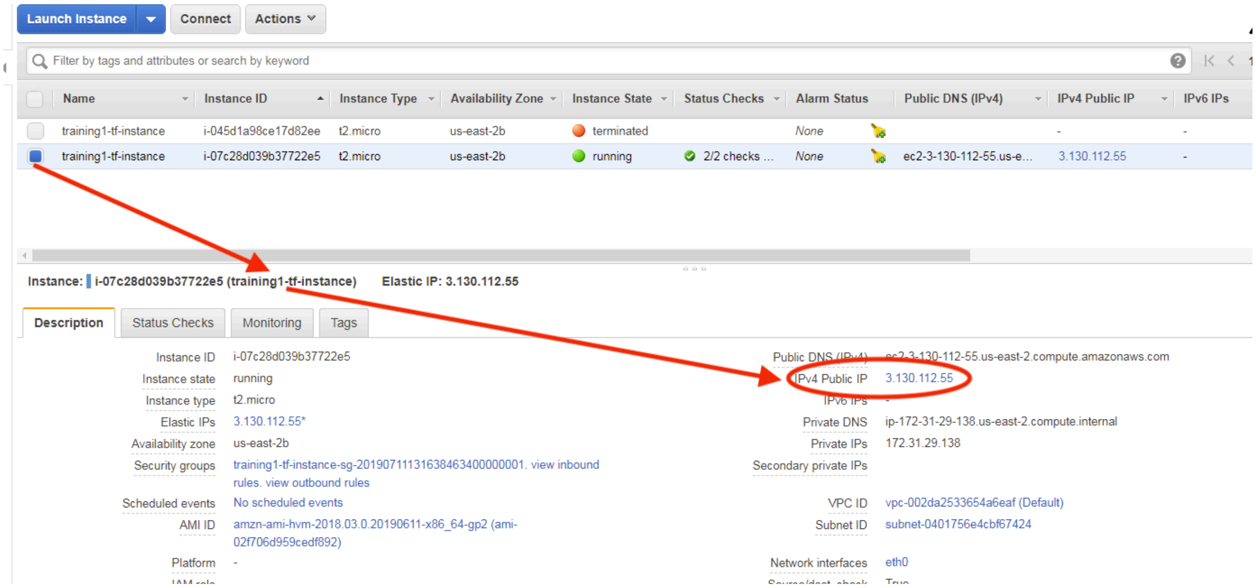
32. Save and close the file

33. Create a new ec2 instance with “terraform apply”



```
PS D:\Users\tennis\terraform_training> terraform apply -auto-approve
data.aws_vpc.default: Refreshing state...
data.aws_ami.amazon_linux: Refreshing state...
data.aws_subnet_ids.all: Refreshing state...
module.security_group.aws_security_group.this_name_prefix[0]: Creating...
module.security_group.aws_security_group.this_name_prefix[0]: Creation complete after 1s [id=sg-0aa15e93baa2fab9d]
module.security_group.aws_security_group_rule.ingress_rules[0]: Creating...
module.ec2.aws_instance.this_t2[0]: Creating...
module.security_group.aws_security_group_rule.egress_rules[0]: Creating...
module.security_group.aws_security_group_rule.ingress_rules[1]: Creating...
module.security_group.aws_security_group_rule.ingress_rules[0]: Creation complete after 1s [id=sgrule-3901528972]
module.security_group.aws_security_group_rule.egress_rules[0]: Creation complete after 1s [id=sgrule-4187975843]
module.security_group.aws_security_group_rule.ingress_rules[1]: Creation complete after 2s [id=sgrule-2387140780]
module.ec2.aws_instance.this_t2[0]: Still creating... [10s elapsed]
module.ec2.aws_instance.this_t2[0]: Still creating... [20s elapsed]
module.ec2.aws_instance.this_t2[0]: Creation complete after 22s [id=i-07c28d039b37722e5]
aws_eip.this: Creating...
aws_eip.this: Creation complete after 1s [id=eipalloc-022b69a1498a04261]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.
PS D:\Users\tennis\terraform_training>
```

34. Again, switch over to your browser and look for the new ec2 instance
35. Locate the new instance’s ip address

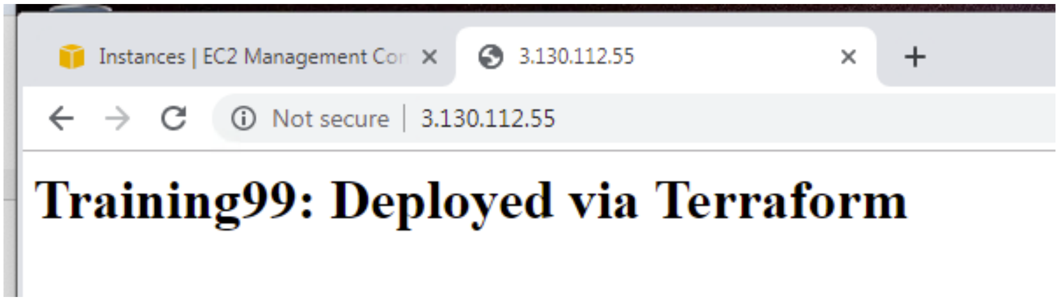


Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
training1-tf-instance	i-045d1a98ce17d82ee	t2.micro	us-east-2b	terminated		None		-	-
training1-tf-instance	i-07c28d039b37722e5	t2.micro	us-east-2b	running	2/2 checks ...	None	ec2-3-130-112-55.us-e...	3.130.112.55	-

Instance: i-07c28d039b37722e5 (training1-tf-instance)   Elastic IP: 3.130.112.55

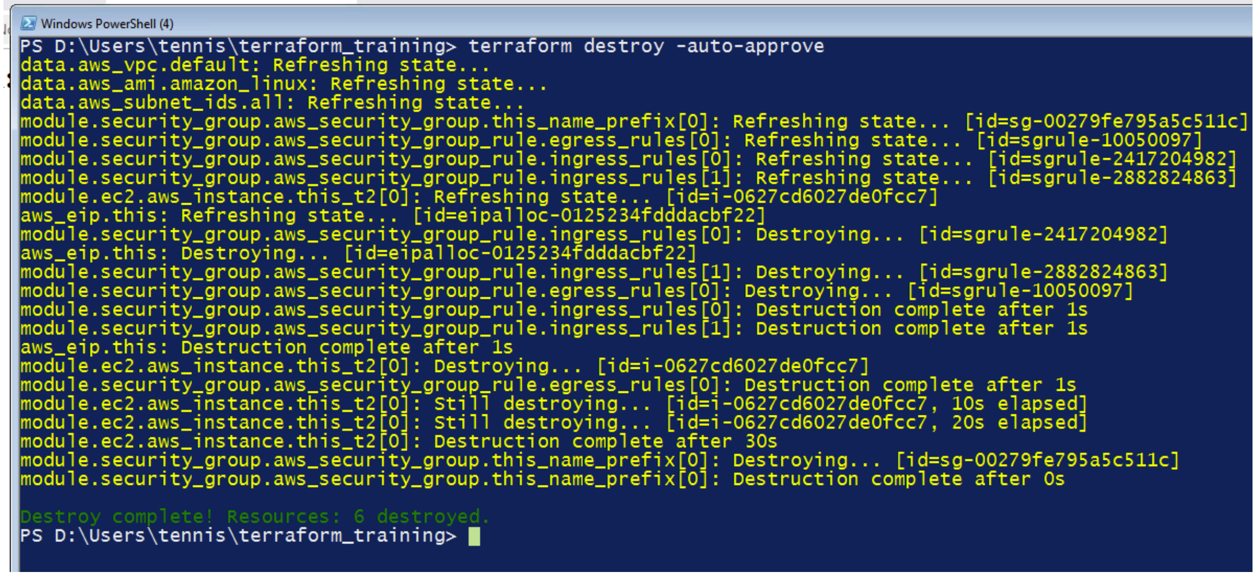
Instance ID	i-07c28d039b37722e5	Public DNS (IPv4)	ec2-3-130-112-55.us-east-2.compute.amazonaws.com
Instance state	running	IPv4 Public IP	3.130.112.55
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs	3.130.112.55*	Private DNS	ip-172-31-29-138.us-east-2.compute.internal
Availability zone	us-east-2b	Private IPs	172.31.29.138
Security groups	training1-tf-instance-sg-20190711131638463400000001. view inbound rules. view outbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-002da2533654a6eaf (Default)
AMI ID	amzn-ami-hvm-2018.03.0.20190611-x86_64-gp2 (ami-02f706d959cedf892)	Subnet ID	subnet-0401756e4cbf67424
Platform	-	Network interfaces	eth0

36. Open a new tab in your WorkSpace browser, plug in that address, and hit enter



Training99: Deployed via Terraform

37. Switch back to your command line and type “terraform destroy -auto-approve”



```
PS D:\Users\tennis\terraform_training> terraform destroy -auto-approve
data.aws_vpc.default: Refreshing state...
data.aws_ami.amazon_linux: Refreshing state...
data.aws_subnet_ids.all: Refreshing state...
module.security_group.aws_security_group.this_name_prefix[0]: Refreshing state... [id=sg-00279fe795a5c511c]
module.security_group.aws_security_group_rule.egress_rules[0]: Refreshing state... [id=sgrule-10050097]
module.security_group.aws_security_group_rule.ingress_rules[0]: Refreshing state... [id=sgrule-2417204982]
module.security_group.aws_security_group_rule.ingress_rules[1]: Refreshing state... [id=sgrule-2882824863]
module.ec2.aws_instance.this_t2[0]: Refreshing state... [id=i-0627cd6027de0fcc7]
aws_eip.this: Refreshing state... [id=eipalloc-0125234fdddacbf22]
module.security_group.aws_security_group_rule.ingress_rules[0]: Destroying... [id=sgrule-2417204982]
aws_eip.this: Destroying... [id=eipalloc-0125234fdddacbf22]
module.security_group.aws_security_group_rule.ingress_rules[1]: Destroying... [id=sgrule-2882824863]
module.security_group.aws_security_group_rule.egress_rules[0]: Destroying... [id=sgrule-10050097]
module.security_group.aws_security_group_rule.ingress_rules[0]: Destruction complete after 1s
module.security_group.aws_security_group_rule.ingress_rules[1]: Destruction complete after 1s
aws_eip.this: Destruction complete after 1s
module.ec2.aws_instance.this_t2[0]: Destroying... [id=i-0627cd6027de0fcc7]
module.security_group.aws_security_group_rule.egress_rules[0]: Destruction complete after 1s
module.ec2.aws_instance.this_t2[0]: Still destroying... [id=i-0627cd6027de0fcc7, 10s elapsed]
module.ec2.aws_instance.this_t2[0]: Still destroying... [id=i-0627cd6027de0fcc7, 20s elapsed]
module.ec2.aws_instance.this_t2[0]: Destruction complete after 30s
module.security_group.aws_security_group.this_name_prefix[0]: Destroying... [id=sg-00279fe795a5c511c]
module.security_group.aws_security_group.this_name_prefix[0]: Destruction complete after 0s

Destroy complete! Resources: 6 destroyed.
PS D:\Users\tennis\terraform_training>
```