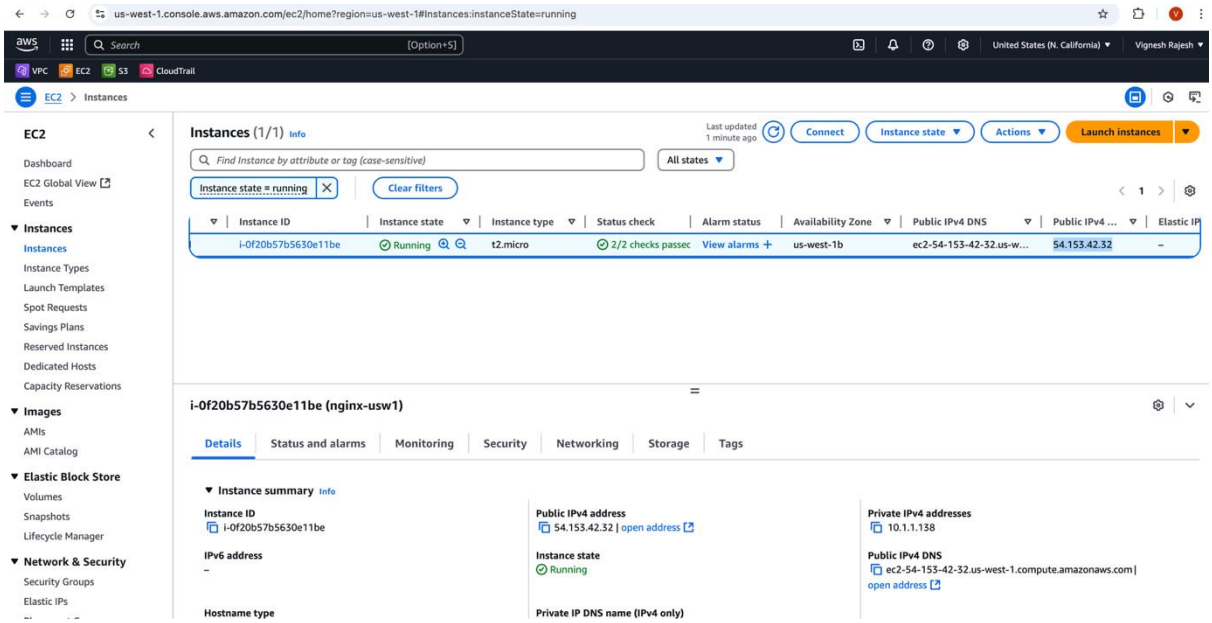
```
vignesh@Vignesh multi-region-ec2 % aws configure
AWS Access Key ID [****************QKWD]: AKIAVIOZFUX6O5TCQKWD
AWS Secret Access Key [****************uVHR]: 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
Default region name [us-east-1]:
Default output format [None]:
vignesh@Vignesh multi-region-ec2 % terraform -v
Terraform v1.11.4
on darwin_arm64
+ provider registry.terraform.io/hashicorp/aws v5.95.0
vignesh@Vignesh multi-region-ec2 % ls
main.tf                         multi-region-ec2.code-workspace terraform.tfstate               terraform.tfstate.backup
vignesh@Vignesh multi-region-ec2 % rm main.tf
vignesh@Vignesh multi-region-ec2 % rm multi-region-ec2.code-workspace
vignesh@Vignesh multi-region-ec2 % rm terraform.tfstate
vignesh@Vignesh multi-region-ec2 % rm terraform.tfstate.backup
vignesh@Vignesh multi-region-ec2 % nano main.tf
vignesh@Vignesh multi-region-ec2 % terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.95.0

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.
vignesh@Vignesh multi-region-ec2 %
```

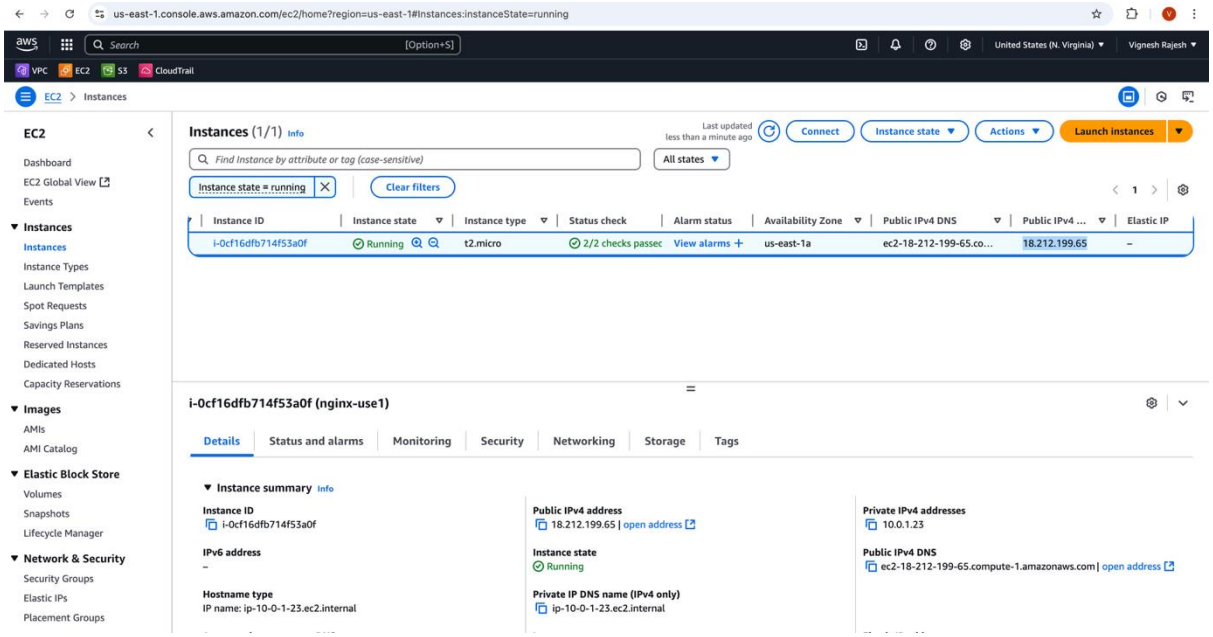
```
      + tags_all                                   = (known after apply)
      + vpc_id                                     = (known after apply)
    }

  # aws_subnet.subnet_usw1 will be created
  + resource "aws_subnet" "subnet_usw1" {
      + arn                                        = (known after apply)
      + assign_ipv6_address_on_creation            = false
      + availability_zone                          = "us-west-1b"
      + availability_zone_id                       = (known after apply)
      + cidr_block                                 = "10.1.1.0/24"
      + enable_dns64                               = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                         = (known after apply)
      + ipv6_cidr_block_association_id             = (known after apply)
      + ipv6_native                                = false
      + map_public_ip_on_launch                    = false
      + owner_id                                   = (known after apply)
      + private_dns_hostname_type_on_launch        = (known after apply)
      + tags_all                                   = (known after apply)
      + vpc_id                                     = (known after apply)
    }

  # aws_vpc.vpc_use1 will be created
  + resource "aws_vpc" "vpc_use1" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = true
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags_all                             = (known after apply)
    }

  # aws_vpc.vpc_usw1 will be created
  + resource "aws_vpc" "vpc_usw1" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.1.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = true
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags_all                             = (known after apply)
    }

Plan: 14 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + us_east_instance_public_ip = (known after apply)
  + us_west_instance_public_ip = (known after apply)
```

```
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
vignesh@Vignesh multi-region-ec2 %
```

```
      + private_dns_hostname_type_on_launch            = (known after apply)
      + tags_all                                       = (known after apply)
      + vpc_id                                         = (known after apply)
    }

  # aws_vpc.vpc_use1 will be created
  + resource "aws_vpc" "vpc_use1" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags_all                             = (known after apply)
    }

  # aws_vpc.vpc_usw1 will be created
  + resource "aws_vpc" "vpc_usw1" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.1.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + tags_all                             = (known after apply)
    }

Plan: 8 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_vpc.vpc_use1: Creating...
aws_vpc.vpc_usw1: Creating...
aws_vpc.vpc_usw1: Creation complete after 5s [id=vpc-072afb1111dc26324]
aws_subnet.subnet_usw1: Creating...
aws_security_group.sg_usw1: Creating...
aws_vpc.vpc_use1: Creation complete after 5s [id=vpc-022b893db8298dc37]
aws_subnet.subnet_use1: Creating...
aws_security_group.sg_use1: Creating...
aws_subnet.subnet_usw1: Creation complete after 2s [id=subnet-07a55f6890368e497]
aws_subnet.subnet_use1: Creation complete after 2s [id=subnet-021f7735014611ac9]
aws_security_group.sg_usw1: Creation complete after 6s [id=sg-0dd505e2a76d94cf8]
aws_instance.ec2_usw1: Creating...
aws_security_group.sg_use1: Creation complete after 6s [id=sg-01d64d58b7f288d3c]
aws_instance.ec2_use1: Creating...
aws_instance.ec2_usw1: Still creating... [10s elapsed]
aws_instance.ec2_use1: Still creating... [10s elapsed]
aws_instance.ec2_usw1: Creation complete after 16s [id=i-08e6a1e54dfc8ac21]
aws_instance.ec2_use1: Creation complete after 16s [id=i-0e98dd913f6c8f515]

Apply complete! Resources: 8 added, 0 changed, 0 destroyed.
vignesh@Vignesh multi-region-ec2 %
```


Not Secure 54.153.42.32
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.


us-west-1.console.aws.amazon.com/ec2/home?region=us-west-1#Instances:instanceState=running
Search [Option+S]
United States (N. California)
Vignesh Rajesh
VPC EC2 S3 CloudTrail
EC2 > Instances
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Instances (1/1) Info
Last updated 1 minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Instance state = running
Clear filters
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Elastic IP
i-0f20b57b5630e11be
Running
t2.micro
2/2 checks passed
View alarms
us-west-1b
ec2-54-153-42-32.us-w...
54.153.42.32
i-0f20b57b5630e11be (nginx-usw1)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-0f20b57b5630e11be
IPv6 address
Hostname type
Public IPv4 address
54.153.42.32 | open address
Instance state
Running
Private IP DNS name (IPv4 only)
Private IPv4 addresses
10.1.1.138
Public IPv4 DNS
ec2-54-153-42-32.us-west-1.compute.amazonaws.com | open address


us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:instanceState=running
Search
[Option+S]
United States (N. Virginia)
Vignesh Rajesh
VPC
EC2
S3
CloudTrail
EC2 > Instances
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Instances (1/1) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Instance state = running
Clear filters
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Elastic IP
i-0cf16dfb714f53a0f
Running
t2.micro
2/2 checks passed
View alarms
us-east-1a
ec2-18-212-199-65.co...
18.212.199.65
–
i-0cf16dfb714f53a0f (nginx-use1)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-0cf16dfb714f53a0f
IPv6 address
–
Hostname type
IP name: ip-10-0-1-23.ec2.internal
Public IPv4 address
18.212.199.65 | open address
Instance state
Running
Private IP DNS name (IPv4 only)
ip-10-0-1-23.ec2.internal
Private IPv4 addresses
10.0.1.23
Public IPv4 DNS
ec2-18-212-199-65.compute-1.amazonaws.com | open address


Not Secure
18.212.199.65
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.