

## Lunching EC2 and hosting an node application from Terraform.

We initialize the terraform and it is initilized successfully.

```
root@samundra:~/terraformdemo/rds-terraform# terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.54.1

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.
root@samundra:~/terraformdemo/rds-terraform#
```

We now will execute the plan command and the plan command will create the resources.

```
root@samundra:~/terraformdemo/rds-terraform# terraform plan

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

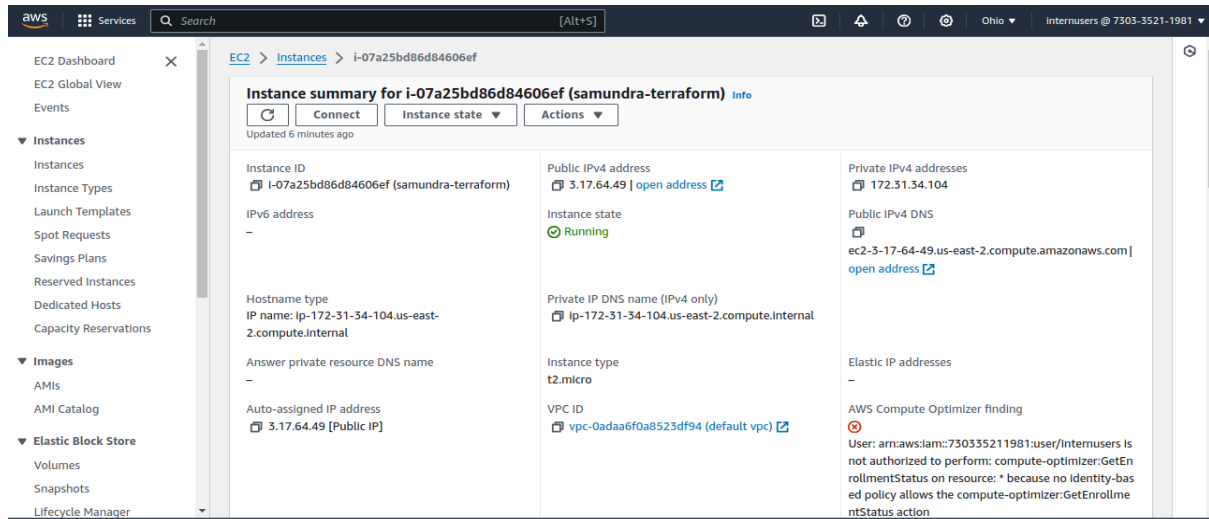
Terraform will perform the following actions:

# aws_db_instance.db_instance will be created
+ resource "aws_db_instance" "db_instance" {
+   address                = (known after apply)
+   allocated_storage      = 20
+   apply_immediately      = false
+   arn                    = (known after apply)
+   auto_minor_version_upgrade = true
+   availability_zone      = "us-east-2a"
+   backup_retention_period = (known after apply)
+   backup_target           = (known after apply)
}
```

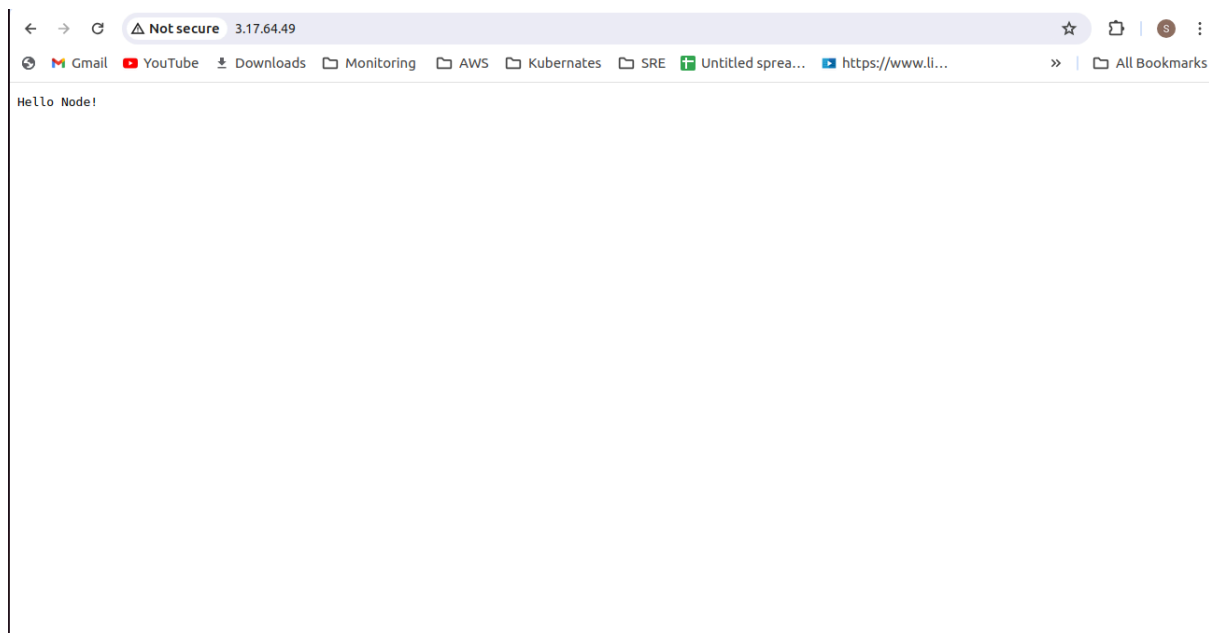
We can verify the ec2 instance as per the image.

Instances (1/2) Info							
Find Instance by attribute or tag (case-sensitive)							
All states							
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	amrit	I-0a212f09cd1df4633	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-2b
<input checked="" type="checkbox"/>	samundra-terr...	I-003196db7b7c42120	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-2b

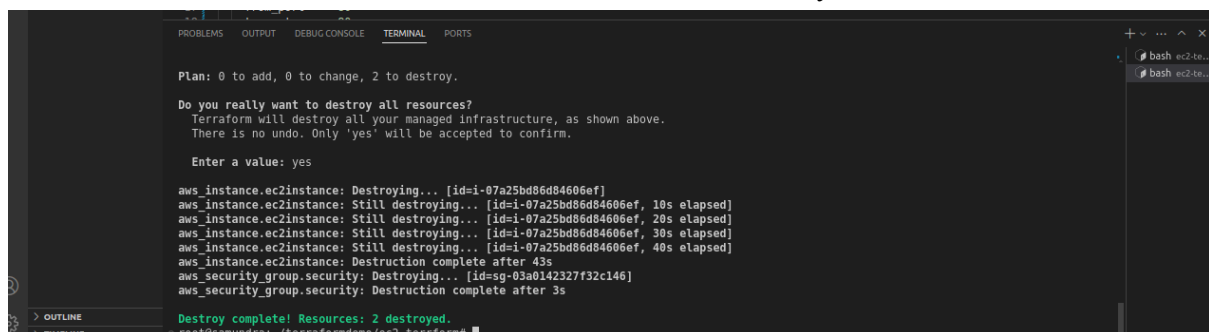
We can further verify the instance from the instance.



As, per our requirements we have install dependencies and hosted the application in apache and then we access the application here within.

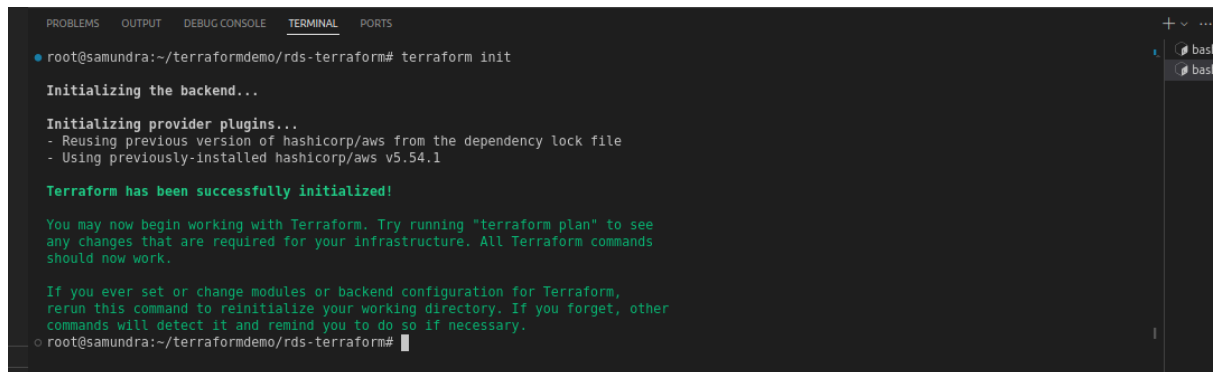


Furthermore, we have deleted our ec2 machine from terraform destroy.



## Lunching RDS and accessing the database from Terraform.

We initialize the terraform and it is initilized successfully.

A terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing the output of the 'terraform init' command. The output includes messages about initializing the backend and provider plugins, and a confirmation that Terraform has been successfully initialized. The prompt is root@samundra:~/terraformdemo/rds-terraform#.

```

root@samundra:~/terraformdemo/rds-terraform# terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.54.1

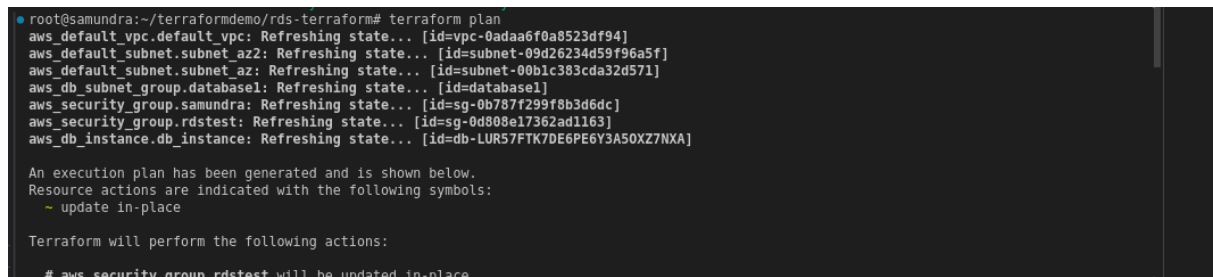
Terraform has been successfully initialized!

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.
root@samundra:~/terraformdemo/rds-terraform#

```

We now will execute the plan command and the plan command will create the resources.

A terminal window showing the output of the 'terraform plan' command. It lists several AWS resources being refreshed, including vpc, subnets, database, security groups, and an RDS instance. It also shows the execution plan and the actions that will be performed. The prompt is root@samundra:~/terraformdemo/rds-terraform#.

```

root@samundra:~/terraformdemo/rds-terraform# terraform plan
aws_default_vpc.default_vpc: Refreshing state... [id=vpc-0adaa6f0a8523df94]
aws_default_subnet.subnet_az2: Refreshing state... [id=subnet-09d26234d59f96a5f]
aws_default_subnet.subnet_az: Refreshing state... [id=subnet-00b1c383cda32d571]
aws_db_subnet_group.database1: Refreshing state... [id=database1]
aws_security_group.samundra: Refreshing state... [id=sg-0b787f299f8b3d6dc]
aws_security_group.rdstest: Refreshing state... [id=sg-0d808e17362ad1163]
aws_db_instance.db_instance: Refreshing state... [id=db-LUR57FTK7DE6PE6Y3A50XZ7NXA]

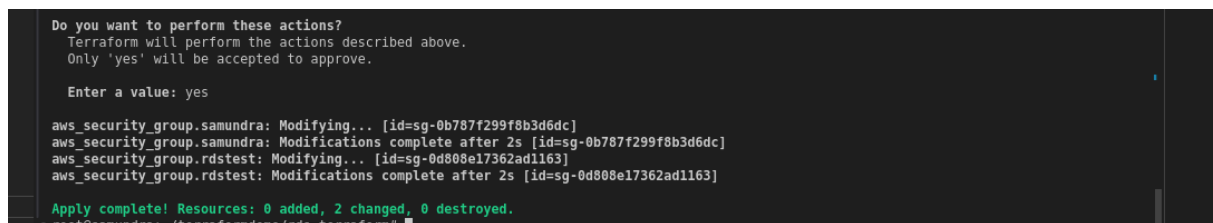
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
~ update in-place

Terraform will perform the following actions:

# aws_security_group.rdstest will be updated in-place

```

We wil then apply and the created resources will be reflected in aws dashboard in the specific resource.

A terminal window showing the output of the 'terraform apply' command. It prompts for confirmation to perform the actions, and then shows the progress of applying the resources. The output indicates that the resources have been successfully applied. The prompt is root@samundra:~/terraformdemo/rds-terraform#.

```

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_security_group.samundra: Modifying... [id=sg-0b787f299f8b3d6dc]
aws_security_group.samundra: Modifications complete after 2s [id=sg-0b787f299f8b3d6dc]
aws_security_group.rdstest: Modifying... [id=sg-0d808e17362ad1163]
aws_security_group.rdstest: Modifications complete after 2s [id=sg-0d808e17362ad1163]

Apply complete! Resources: 0 added, 2 changed, 0 destroyed.
root@samundra:~/terraformdemo/rds-terraform#

```

As, mention earlier we now verify the created database from the dashboard.

○

samundradatabase

Available

Instance

MySQL Community

us-east-2a

db.t3.micro

Summary

DB Identifier

samundradatabase

CPU

2.75%

Status

Available

Class

db.t3.micro

Role

Instance

Current activity

0

Connections

Engine

MySQL Community

Region & AZ

us-east-2a

Recommendations

<

Connectivity & security

Monitoring

Logs & events

Configuration

Zero-ETL integrations

Maintenance & backups

>

Connectivity & security

Endpoint & port

Endpoint

📄

samundradatabase.c780ygmcy1iz.us-east-2.rds.amazonaws.com

Port

3306

Networking

Availability Zone

us-east-2a

VPC

default vpc (vpc-0adaa6f0a8523df94)

Security

VPC security groups

rdstest (sg-0d808e17362ad1163)

Active

Publicly accessible

Yes

Earlier we didnot allow to access our database so, we have all allow port 3306 from any ips to access it.

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

Details

Security group name

rdstest

Security group ID

sg-0d808e17362ad1163

Description

enable mysql/aurora access on port 3306

VPC ID

vpc-0adaa6f0a8523df94

Owner

730335211981

Inbound rules count

2 Permission entries

Outbound rules count

1 Permission entry

Inbound rules

Outbound rules

Tags

Inbound rules (2)

🔄

Manage tags

Edit inbound rules

🔍 Search

< 1 > ⌕

Security group rule...	IP version	Type	Protocol	Port range	Source
sg-r01fd656e60b401fd4	-	MySQL/Aurora	TCP	3306	sg-0b787f29
sg-r0b390bc8e29f29033	IPv4	MySQL/Aurora	TCP	3306	0.0.0.0/0

The configuration we have initilize in our terraform configuration is displayed. It provides the information about our database, database user name and password.

Instance			
Configuration	Instance class	Storage	Performance Insights
DB instance ID samundradatabase	Instance class db.t3.micro	Encryption Not enabled	Performance Insights enabled Turned off
Engine version 5.7.44	vCPU 2	Storage type General Purpose SSD (gp2)	
RDS Extended Support Enabled	RAM 1 GB	Storage 20 GiB	
DB name samundradb	Availability	Provisioned IOPS -	
License model General Public License	Master username samundra	Storage throughput -	
Option groups default:mysql-5-7 <span>In sync</span>	Master password *****	Storage autoscaling Disabled	
Amazon Resource Name (ARN)  arn:aws:rds:us-east-2:73033521198:1:db:samundradatabase	IAM DB authentication Not enabled	Storage file system configuration Current	
Resource ID db-LURS7FTK7DE6PE6Y3A5OXZ7NXA	Multi-AZ No		
Created time June 21, 2024, 14:09 (UTC+05:45)	Secondary Zone -		

We now can access the database with the hepl of endpoint and the user name and password.

```
root@samundra:/home/samundra/Downloads# mysql -h samundradatabase.c780ygmcy1iz.us-east-2.rds.amazonaws.com -u samundra -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 5.7.44 Please upgrade to 8.0 or opt-in to the paid RDS Extended Support service before 5.7 reaches end of standard support on
29 February, 2024: https://a.co/hqqlin0

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| innodb |
| mysql |
| performance_schema |
| samundradb |
| sys |
+-----+
6 rows in set (0.26 sec)
```

Then we wil destroy the database as it is for the test purpose with terraform destroy command.

## VPC creation with terraform configuration file

We initialize the terraform and it is initialized successfully.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
root@samundra:~/terraformdemo/vpc-terraform# terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.55.0

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.
root@samundra:~/terraformdemo/vpc-terraform#
```

We now will execute the plan command and the plan command will create the resources.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

+ ipv6_cidr_block_network_border_group = (known after apply)
+ main_route_table_id                  = (known after apply)
+ owner_id                             = (known after apply)
+ tags                                  = {
+   "Name" = "samundravpc"
}
+ tags_all                             = {
+   "Name" = "samundravpc"
}
}

Plan: 6 to add, 0 to change, 0 to destroy.

-----

Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.
root@samundra:~/terraformdemo/vpc-terraform#
```

We will then apply and the created resources will be reflected in aws dashboard in the specific resource

```
    }
  }
}

Plan: 1 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_instance.samundraec2: Creating...
aws_instance.samundraec2: Still creating... [10s elapsed]
aws_instance.samundraec2: Still creating... [20s elapsed]
aws_instance.samundraec2: Still creating... [30s elapsed]
aws_instance.samundraec2: Still creating... [40s elapsed]
aws_instance.samundraec2: Creation complete after 46s [id=i-0b3abeade8a6776bb]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
root@samundra:~/terraformdemo/vpc-terraform#
```

The created VPC is reflected in the dashboard with the name we mention in the configuration file.

VPC > Your VPCs > vpc-0b2b5746239c5b5e0

### vpc-0b2b5746239c5b5e0 / samundra-vpc

Details info

VPC ID vpc-0b2b5746239c5b5e0	State <span>Available</span>	DNS hostnames Enabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-00dc679984cc8eae7	Main route table rtb-0f37fca8b0e5f4613	Main network ACL acl-0767a382e7fc3799d
Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -	IPv6 CIDR -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups <span>Failed to load rule groups</span>	Owner ID 730335211981	

We further can verify the subnet within the vpc as mention below.

VPC > Subnets > subnet-0cfd710740c1afde

### subnet-0cfd710740c1afde / samundra-public-subnet

Details

Subnet ID subnet-0cfd710740c1afde	Subnet ARN arn:aws:ec2:us-east-2:730335211981:subnet/subnet-0cfd710740c1afde	State <span>Available</span>	IPv4 CIDR 10.0.1.0/24
Available IPv4 addresses 250	IPv6 CIDR -	Availability Zone us-east-2a	Availability Zone ID use2-az1
VPC vpc-0b2b5746239c5b5e0   samundra-vpc	Route table rtb-0d05595093cb85894   samundra-public-rt	Network ACL acl-0767a382e7fc3799d	Default subnet No
Auto-assign public IPv4 address Yes	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No	Customer-owned IPv4 pool -
Outpost ID -	IPv4 CIDR reservations -	IPv6 CIDR reservations -	IPv6-only No
Hostname type IP name	Resource name DNS AAAA record Disabled	Resource name DNS AAAA record Disabled	DNS64 Disabled
Owner 730335211981			

We can verify that the route table is associated within the subnet and the internet gateway.

VPC > Route tables > rtb-0d05595093cb85894

rtb-0d05595093cb85894 / samundra-public-rt

Actions

DetailsInfo

Route table ID

rtb-0d05595093cb85894

VPC

vpc-0b2b5746239c5b5e0 | samundra-vpc

Main

No

Owner ID

730335211981

Explicit subnet associations

subnet-0cfd7710740c1afde / samundra-public-subnet

Edge associations

-

Routes

Subnet associations

Edge associations

Route propagation

Tags

Routes (2)

Both

Edit routes

Filter routes

< 1 >

Destination	Target	Status	Propagated
0.0.0.0/0	igw-03628407b33d870af	Active	No
10.0.0.0/16	local	Active	No

Internet gateway is associated with the vpc so that we can access the ec2 instance through the public network.

VPC > Internet gateways > igw-03628407b33d870af

igw-03628407b33d870af / samundra-igw

Actions

DetailsInfo

Internet gateway ID

igw-03628407b33d870af

State

Attached

VPC ID

vpc-0b2b5746239c5b5e0 | samundra-vpc

Owner

730335211981

Tags

Manage tags

Search tags

Key	Value
Name	samundra-igw

We can also verify the ec2 instance that we have created.



EC2 > Instances > i-0622a838c08065bc5

### Instance summary for i-0622a838c08065bc5 (samundraec2) [Info](#)

Updated less than a minute ago

Instance ID

i-0622a838c08065bc5 (samundraec2)

IPv6 address

—

Hostname type

IP name: ip-10-0-1-230.us-east-2.compute.internal

Answer private resource DNS name

—

Auto-assigned IP address

3.147.59.173 [Public IP]

IAM Role

—

IMDSv2

Optional

EC2 recommends setting IMDSv2 to required | [Learn more](#)

Public IPv4 address

3.147.59.173 | [open address](#)

Instance state

**Running**

Private IP DNS name (IPv4 only)

ip-10-0-1-230.us-east-2.compute.internal

Instance type

t2.micro

VPC ID

vpc-0b2b5746239c5b5e0 (samundra-vpc) | [open address](#)

Subnet ID

subnet-0cfd710740c1afde (samundra-public-subnet) | [open address](#)

Instance ARN

arn:aws:ec2:us-east-2:730335211981:instance/i-0622a838c08065bc5

Private IPv4 addresses

10.0.1.230

Public IPv4 DNS

ec2-3-147-59-173.us-east-2.compute.amazonaws.com | [open address](#)

Elastic IP addresses

—

AWS Compute Optimizer finding

User: arn:aws:iam::730335211981:user/Internusers is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: \* because no identity-based policy allows the compute-optimizer:GetEnrollmentStatus action

[Retry](#)

Auto Scaling Group name

—

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie pref](#)

We can further verify through the ec2 instances which ip has been assigned and the route table within that ec2.

```
ubuntu@ip-10-0-1-230:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 10.0.1.230 netmask 255.255.255.0 broadcast 10.0.1.255
    inet6 fe80::f5:3fff:fe9c:dbbd prefixlen 64 scopeid 0x20<link>
    ether 02:f5:3f:9c:db:bd txqueuelen 1000 (Ethernet)
    RX packets 775 bytes 376153 (376.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 697 bytes 83804 (83.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 176 bytes 14828 (14.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 176 bytes 14828 (14.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ubuntu@ip-10-0-1-230:~$ ip route show
default via 10.0.1.1 dev eth0 proto dhcp src 10.0.1.230 metric 100
10.0.1.0/24 dev eth0 proto kernel scope link src 10.0.1.230
10.0.1.1 dev eth0 proto dhcp scope link src 10.0.1.230 metric 100
```

We can now delete the our created vpc and ec2 instance from the terraform destroy command.

```
aws_route_table.public: Destroying... [id=rtb-0d05595093cb85894]
aws_route_table.public: Destruction complete after 2s
aws_internet_gateway.samundra-gw: Destroying... [id=igw-03628407b33d870af]
aws_instance.web: Still destroying... [id=i-0622a838c08065bc5, 10s elapsed]
aws_internet_gateway.samundra-gw: Still destroying... [id=igw-03628407b33d870af, 10s elapsed]
aws_instance.web: Still destroying... [id=i-0622a838c08065bc5, 20s elapsed]
aws_internet_gateway.samundra-gw: Still destroying... [id=igw-03628407b33d870af, 20s elapsed]
aws_instance.web: Still destroying... [id=i-0622a838c08065bc5, 30s elapsed]
aws_internet_gateway.samundra-gw: Still destroying... [id=igw-03628407b33d870af, 30s elapsed]
aws_instance.web: Destruction complete after 34s
aws_subnet.public: Destroying... [id=subnet-0cfd710740c1afde]
aws_security_group.samundra_sg: Destroying... [id=sg-00700af96d8d9a307]
aws_subnet.public: Destruction complete after 2s
aws_security_group.samundra_sg: Destruction complete after 3s
aws_internet_gateway.samundra-gw: Destruction complete after 35s
aws_vpc.samundra: Destroying... [id=vpc-0b2b5746239c5b5e0]
aws_vpc.samundra: Destruction complete after 1s

Destroy complete! Resources: 7 destroyed.
root@samundra:~/terraformdemo/vpc-terraform#
```

For further verification of the configuration please visit at;  
<https://github.com/samundra77/samundra-terraform-repo>