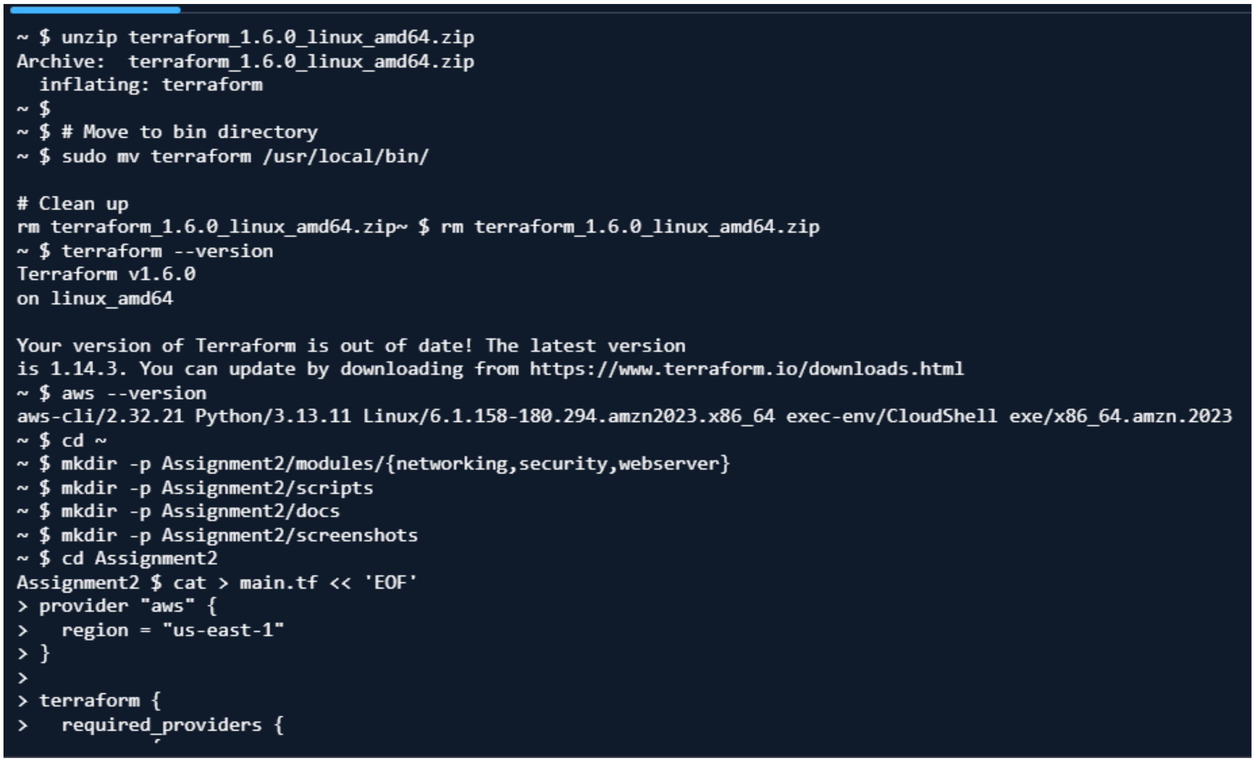
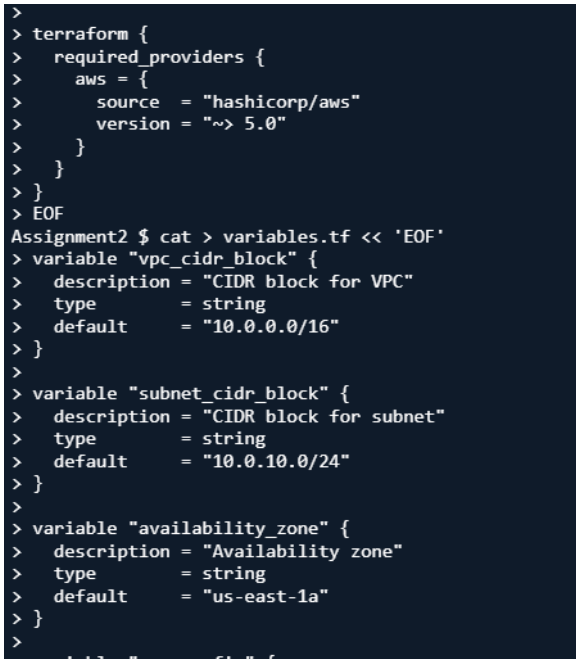
NAME: SANA TARIQ
CLASS: BSE 5B
REG.NO: 058
SUBJECT: CLOUD COMPUTING

ASSIGNMENT#02

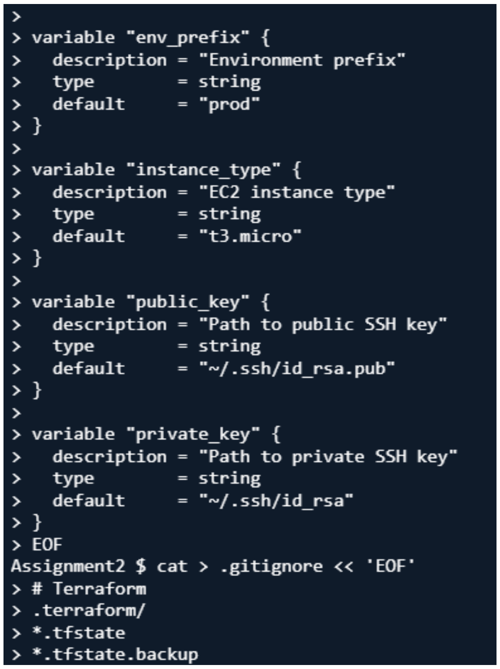
Part 1: Infrastructure Setup



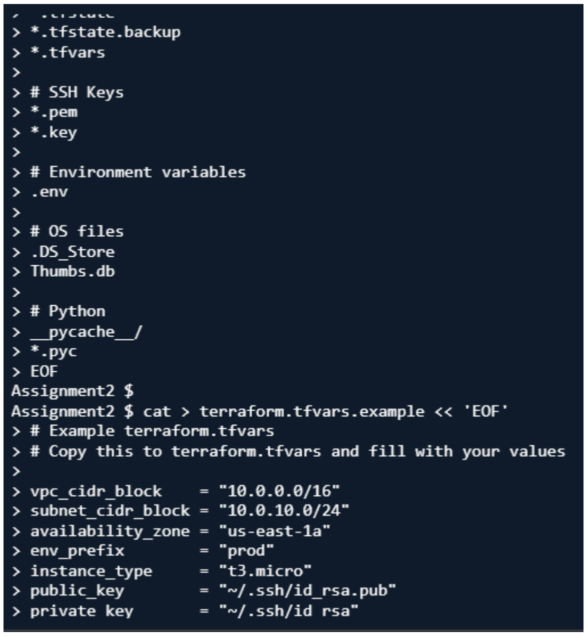
```
~ $ unzip terraform_1.6.0_linux_amd64.zip
Archive:  terraform_1.6.0_linux_amd64.zip
  inflating: terraform
~ $
~ $ # Move to bin directory
~ $ sudo mv terraform /usr/local/bin/

# Clean up
rm terraform_1.6.0_linux_amd64.zip~ $ rm terraform_1.6.0_linux_amd64.zip
~ $ terraform --version
Terraform v1.6.0
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.14.3. You can update by downloading from https://www.terraform.io/downloads.html
~ $ aws --version
aws-cli/2.32.21 Python/3.13.11 Linux/6.1.158-180.294.amzn2023.x86_64 exec-env/CloudShell exe/x86_64.amzn.2023
~ $ cd ~
~ $ mkdir -p Assignment2/modules/{networking,security,webserver}
~ $ mkdir -p Assignment2/scripts
~ $ mkdir -p Assignment2/docs
~ $ mkdir -p Assignment2/screenshots
~ $ cd Assignment2
Assignment2 $ cat > main.tf << 'EOF'
> provider "aws" {
>   region = "us-east-1"
> }
>
> terraform {
>   required_providers {
```



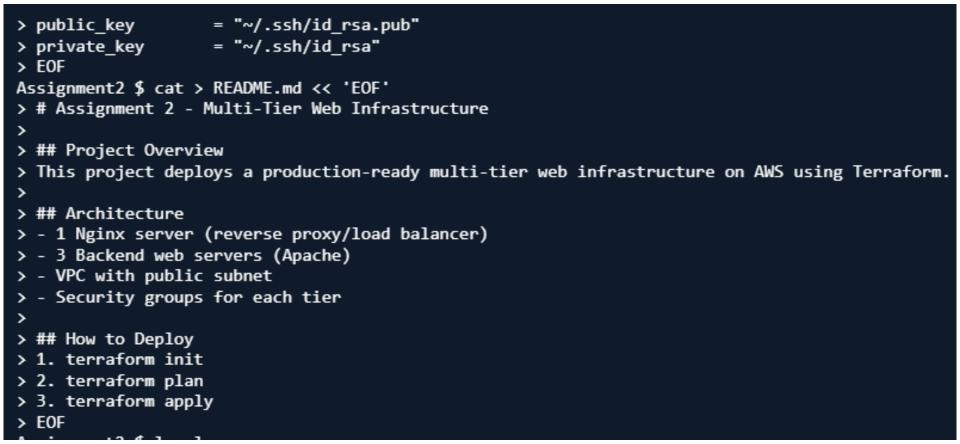
```
>
> terraform {
>   required_providers {
>     aws = {
>       source  = "hashicorp/aws"
>       version = "~> 5.0"
>     }
>   }
> }
> EOF
Assignment2 $ cat > variables.tf << 'EOF'
> variable "vpc_cidr_block" {
>   description = "CIDR block for VPC"
>   type        = string
>   default     = "10.0.0.0/16"
> }
>
> variable "subnet_cidr_block" {
>   description = "CIDR block for subnet"
>   type        = string
>   default     = "10.0.10.0/24"
> }
>
> variable "availability_zone" {
>   description = "Availability zone"
>   type        = string
>   default     = "us-east-1a"
> }
>
```



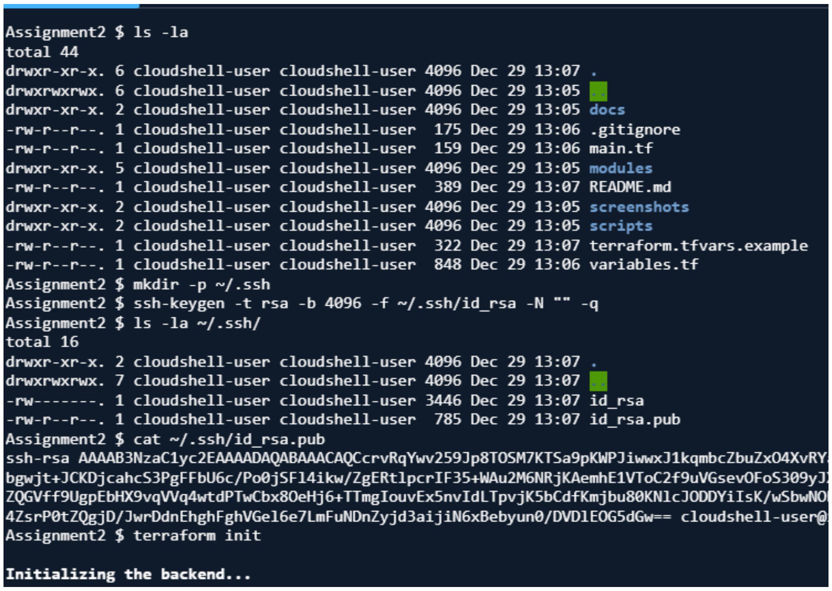
```
>
> variable "env_prefix" {
>   description = "Environment prefix"
>   type        = string
>   default     = "prod"
> }
>
> variable "instance_type" {
>   description = "EC2 instance type"
>   type        = string
>   default     = "t3.micro"
> }
>
> variable "public_key" {
>   description = "Path to public SSH key"
>   type        = string
>   default     = "~/.ssh/id_rsa.pub"
> }
>
> variable "private_key" {
>   description = "Path to private SSH key"
>   type        = string
>   default     = "~/.ssh/id_rsa"
> }
> EOF
Assignment2 $ cat > .gitignore << 'EOF'
> # Terraform
> .terraform/
> *.tfstate
> *.tfstate.backup
```



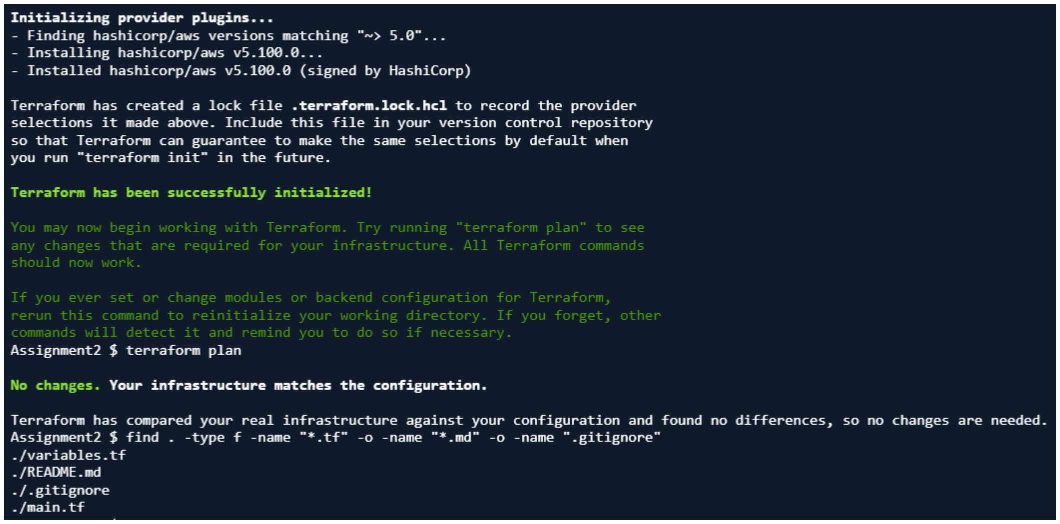
```
> *.tfstate.backup
> *.tfvars
>
> # SSH Keys
> *.pem
> *.key
>
> # Environment variables
> .env
>
> # OS files
> .DS_Store
> Thumbs.db
>
> # Python
> __pycache__/
> *.pyc
> EOF
Assignment2 $
Assignment2 $ cat > terraform.tfvars.example << 'EOF'
> # Example terraform.tfvars
> # Copy this to terraform.tfvars and fill with your values
>
> vpc_cidr_block    = "10.0.0.0/16"
> subnet_cidr_block = "10.0.10.0/24"
> availability_zone = "us-east-1a"
> env_prefix        = "prod"
> instance_type     = "t3.micro"
> public_key        = "~/.ssh/id_rsa.pub"
> private key       = "~/.ssh/id rsa"
```



```
> public_key        = "~/.ssh/id_rsa.pub"
> private_key       = "~/.ssh/id_rsa"
> EOF
Assignment2 $ cat > README.md << 'EOF'
> # Assignment 2 - Multi-Tier Web Infrastructure
>
> ## Project Overview
> This project deploys a production-ready multi-tier web infrastructure on AWS using Terraform.
>
> ## Architecture
> - 1 Nginx server (reverse proxy/load balancer)
> - 3 Backend web servers (Apache)
> - VPC with public subnet
> - Security groups for each tier
>
> ## How to Deploy
> 1. terraform init
> 2. terraform plan
> 3. terraform apply
> EOF
```

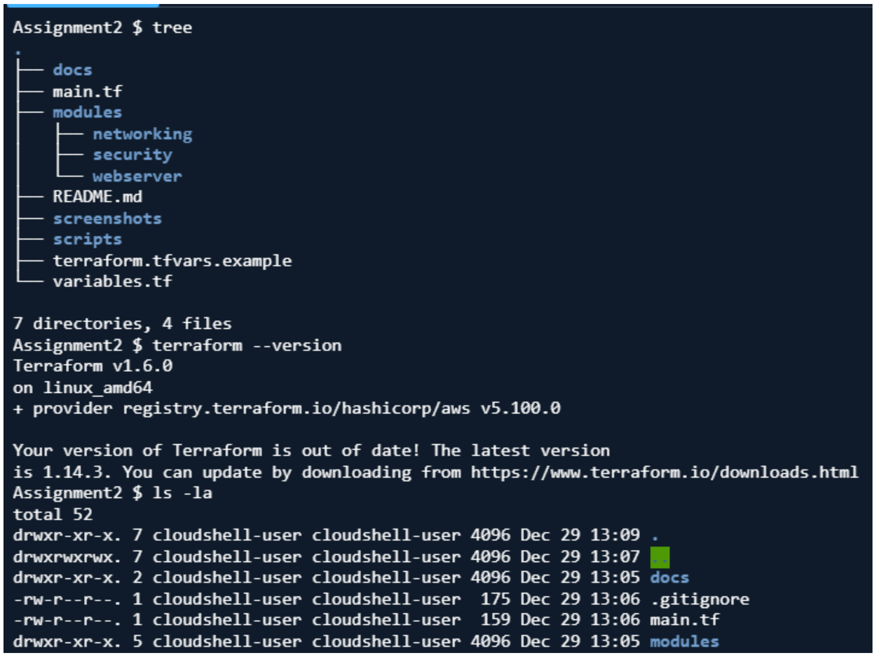


```
Assignment2 $ ls -la
total 44
drwxr-xr-x. 6 cloudshell-user cloudshell-user 4096 Dec 29 13:07 .
drwxrwxrwx. 6 cloudshell-user cloudshell-user 4096 Dec 29 13:05 ..
drwxr-xr-x. 2 cloudshell-user cloudshell-user 4096 Dec 29 13:05 docs
-rw-r--r--. 1 cloudshell-user cloudshell-user  175 Dec 29 13:06 .gitignore
-rw-r--r--. 1 cloudshell-user cloudshell-user  159 Dec 29 13:06 main.tf
drwxr-xr-x. 5 cloudshell-user cloudshell-user 4096 Dec 29 13:05 modules
-rw-r--r--. 1 cloudshell-user cloudshell-user  389 Dec 29 13:07 README.md
drwxr-xr-x. 2 cloudshell-user cloudshell-user 4096 Dec 29 13:05 screenshots
drwxr-xr-x. 2 cloudshell-user cloudshell-user 4096 Dec 29 13:05 scripts
-rw-r--r--. 1 cloudshell-user cloudshell-user  322 Dec 29 13:07 terraform.tfvars.example
-rw-r--r--. 1 cloudshell-user cloudshell-user  848 Dec 29 13:06 variables.tf
Assignment2 $ mkdir -p ~/.ssh
Assignment2 $ ssh-keygen -t rsa -b 4096 -f ~/.ssh/id_rsa -N "" -q
Assignment2 $ ls -la ~/.ssh/
total 16
drwxr-xr-x. 2 cloudshell-user cloudshell-user 4096 Dec 29 13:07 .
drwxrwxrwx. 7 cloudshell-user cloudshell-user 4096 Dec 29 13:07 ..
-rw-------. 1 cloudshell-user cloudshell-user 3446 Dec 29 13:07 id_rsa
-rw-r--r--. 1 cloudshell-user cloudshell-user  785 Dec 29 13:07 id_rsa.pub
Assignment2 $ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCcrvRqYwv259Jp8TOSM7KTSa9pKWPJiwwxJ1kqmbcZbuZxO4XvRY
bgwjt+JCKDjcahcS3PgFFbU6c/Po0jSFl4ikw/ZgERtlpcrIF35+WAu2M6NRjKAemhE1VToC2f9uVGsevOFoS309yJ
ZQGVff9UgpEbHX9vqVVq4wtdPTwCbx8OeHj6+TTmgIouvEx5nvIdLTpvjK5bCdfKmjbu80KNlcJODDYiIsK/wSbwNO
4ZsrP0tZQgjD/JwrDdnEhghFghVGel6e7LmFuNDnZyjd3aijiN6xBebyun0/DVDlEOG5dGw== cloudshell-user@
Assignment2 $ terraform init

Initializing the backend...
```

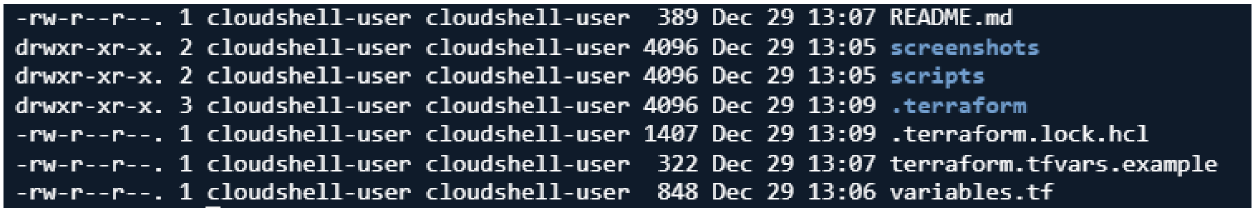


```
Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 5.0"...
- Installing hashicorp/aws v5.100.0...
- Installed hashicorp/aws v5.100.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

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.
Assignment2 $ terraform plan

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.
Assignment2 $ find . -type f -name "*.tf" -o -name "*.md" -o -name ".gitignore"
./variables.tf
./README.md
./.gitignore
./main.tf
```

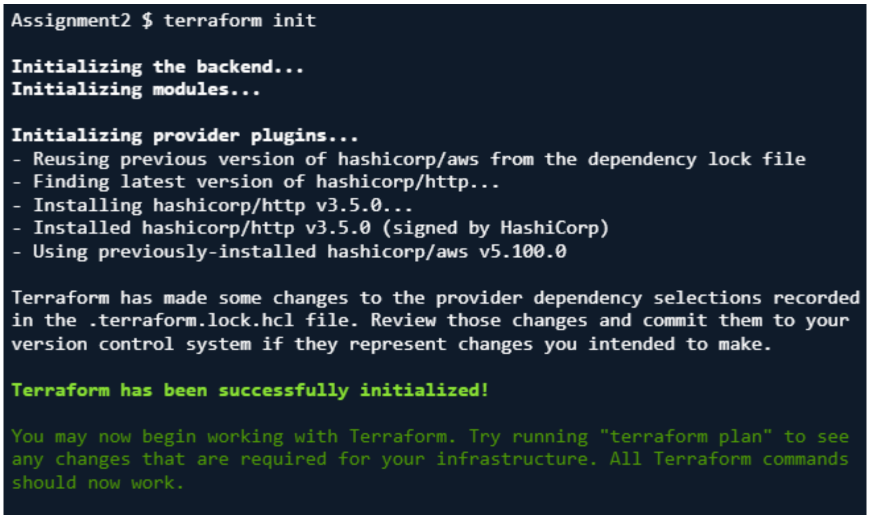
Part 2: Webserver Module



```
Assignment2 $ tree
.
├── docs
├── main.tf
├── modules
│   ├── networking
│   ├── security
│   └── webserver
├── README.md
├── screenshots
├── scripts
├── terraform.tfvars.example
└── variables.tf

7 directories, 4 files
Assignment2 $ terraform --version
Terraform v1.6.0
on linux_amd64
+ provider registry.terraform.io/hashicorp/aws v5.100.0

Your version of Terraform is out of date! The latest version
is 1.14.3. You can update by downloading from https://www.terraform.io/downloads.html
Assignment2 $ ls -la
total 52
drwxr-xr-x. 7 cloudshell-user cloudshell-user 4096 Dec 29 13:09 .
drwxrwxrwx. 7 cloudshell-user cloudshell-user 4096 Dec 29 13:07 ..
drwxr-xr-x. 2 cloudshell-user cloudshell-user 4096 Dec 29 13:05 docs
-rw-r--r--. 1 cloudshell-user cloudshell-user  175 Dec 29 13:06 .gitignore
-rw-r--r--. 1 cloudshell-user cloudshell-user  159 Dec 29 13:06 main.tf
drwxr-xr-x. 5 cloudshell-user cloudshell-user 4096 Dec 29 13:05 modules
```

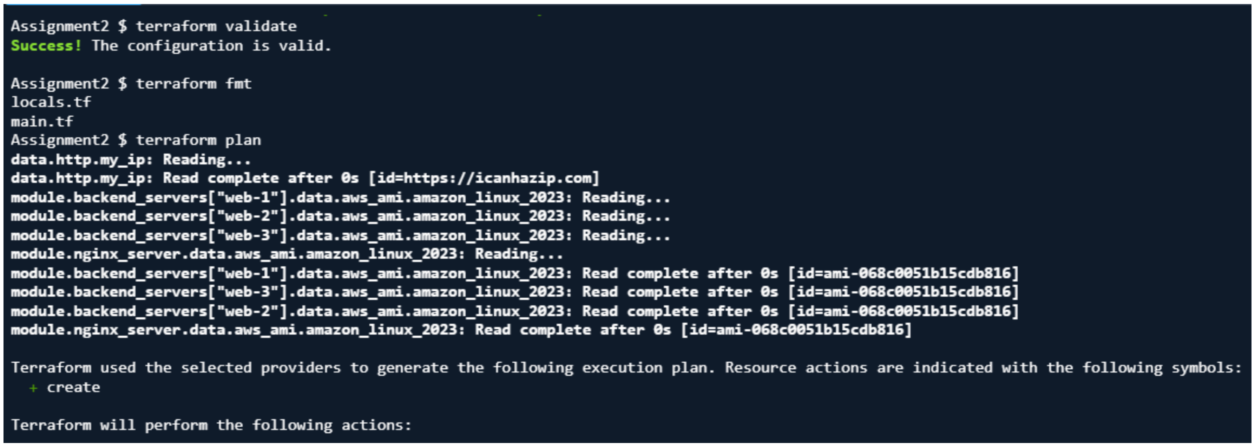


```
-rw-r--r--. 1 cloudshell-user cloudshell-user  389 Dec 29 13:07 README.md
drwxr-xr-x. 2 cloudshell-user cloudshell-user 4096 Dec 29 13:05 screenshots
drwxr-xr-x. 2 cloudshell-user cloudshell-user 4096 Dec 29 13:05 scripts
drwxr-xr-x. 3 cloudshell-user cloudshell-user 4096 Dec 29 13:09 .terraform
-rw-r--r--. 1 cloudshell-user cloudshell-user 1407 Dec 29 13:09 .terraform.lock.hcl
-rw-r--r--. 1 cloudshell-user cloudshell-user  322 Dec 29 13:07 terraform.tfvars.example
-rw-r--r--. 1 cloudshell-user cloudshell-user  848 Dec 29 13:06 variables.tf
```

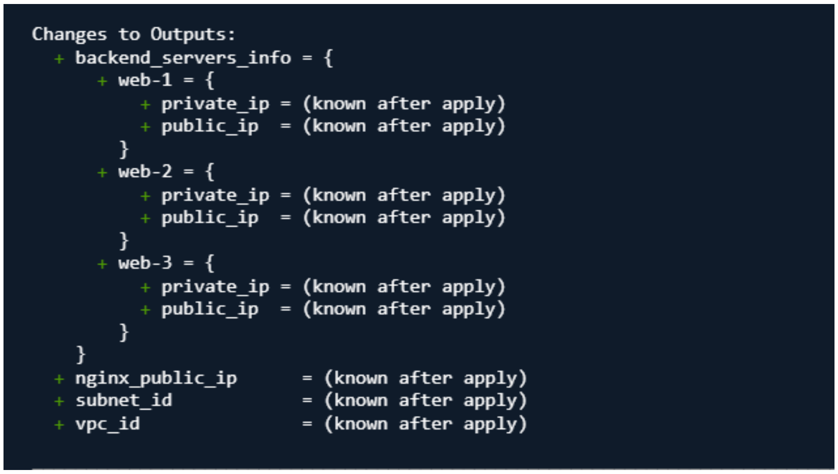


```
Assignment2 $ terraform init

Initializing the backend...
Initializing modules...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Finding latest version of hashicorp/http...
- Installing hashicorp/http v3.5.0...
- Installed hashicorp/http v3.5.0 (signed by HashiCorp)
- Using previously-installed hashicorp/aws v5.100.0

Terraform has made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

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.
```

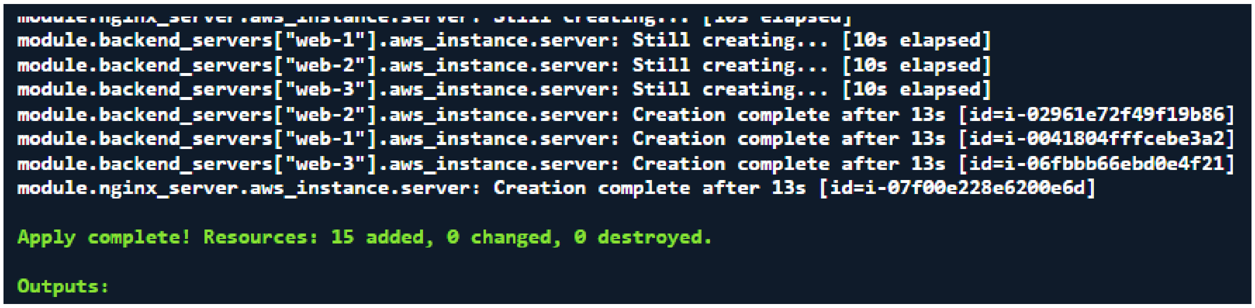
Part 3: Server Configuration Scripts



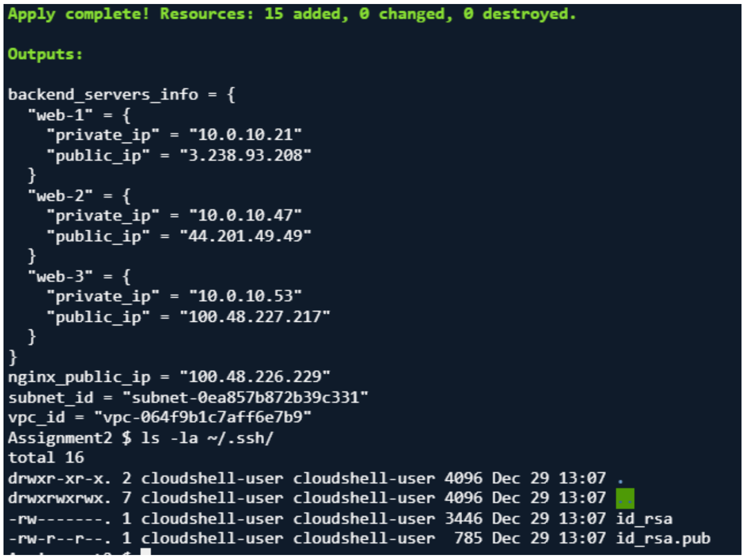
```
Assignment2 $ terraform validate
Success! The configuration is valid.

Assignment2 $ terraform fmt
locals.tf
main.tf
Assignment2 $ terraform plan
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 0s [id=https://icanhazip.com]
module.backend_servers["web-1"].data.aws_ami.amazon_linux_2023: Reading...
module.backend_servers["web-2"].data.aws_ami.amazon_linux_2023: Reading...
module.backend_servers["web-3"].data.aws_ami.amazon_linux_2023: Reading...
module.nginx_server.data.aws_ami.amazon_linux_2023: Reading...
module.backend_servers["web-1"].data.aws_ami.amazon_linux_2023: Read complete after 0s [id=ami-068c0051b15cdb816]
module.backend_servers["web-3"].data.aws_ami.amazon_linux_2023: Read complete after 0s [id=ami-068c0051b15cdb816]
module.backend_servers["web-2"].data.aws_ami.amazon_linux_2023: Read complete after 0s [id=ami-068c0051b15cdb816]
module.nginx_server.data.aws_ami.amazon_linux_2023: Read complete after 0s [id=ami-068c0051b15cdb816]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:
```



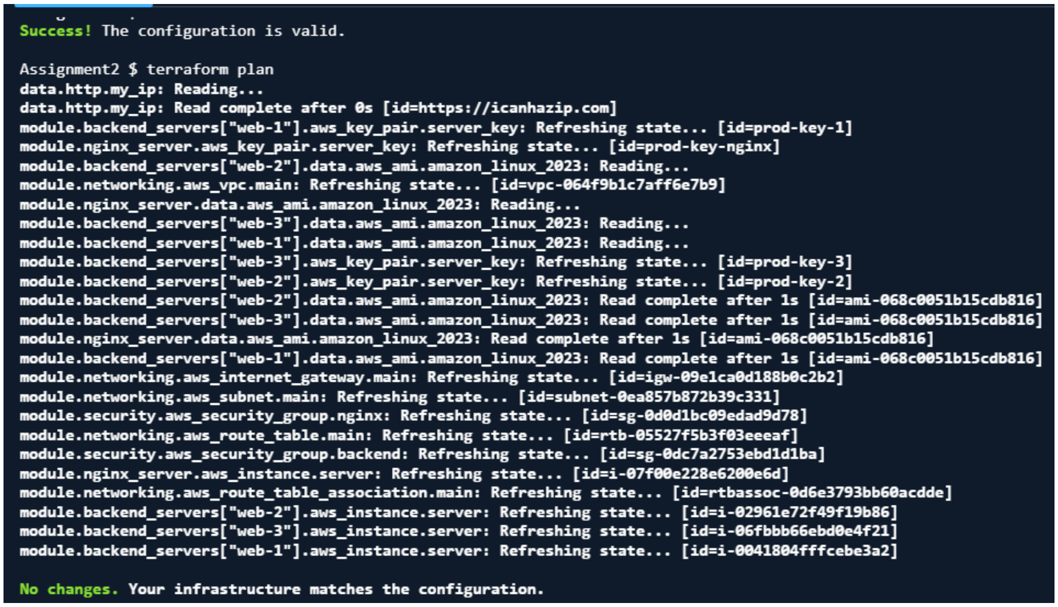
```
Changes to Outputs:
  + backend_servers_info = {
      + web-1 = {
          + private_ip = (known after apply)
          + public_ip  = (known after apply)
        }
      + web-2 = {
          + private_ip = (known after apply)
          + public_ip  = (known after apply)
        }
      + web-3 = {
          + private_ip = (known after apply)
          + public_ip  = (known after apply)
        }
    }
  + nginx_public_ip      = (known after apply)
  + subnet_id            = (known after apply)
  + vpc_id               = (known after apply)
```



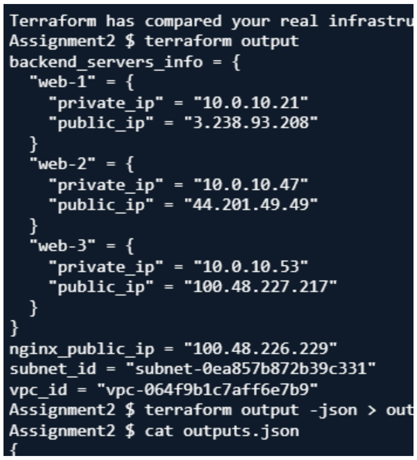
```
module.nginx_server.aws_instance.server: Still creating... [10s elapsed]
module.backend_servers["web-1"].aws_instance.server: Still creating... [10s elapsed]
module.backend_servers["web-2"].aws_instance.server: Still creating... [10s elapsed]
module.backend_servers["web-3"].aws_instance.server: Still creating... [10s elapsed]
module.backend_servers["web-2"].aws_instance.server: Creation complete after 13s [id=i-02961e72f49f19b86]
module.backend_servers["web-1"].aws_instance.server: Creation complete after 13s [id=i-0041804fffcebe3a2]
module.backend_servers["web-3"].aws_instance.server: Creation complete after 13s [id=i-06fbbb66ebd0e4f21]
module.nginx_server.aws_instance.server: Creation complete after 13s [id=i-07f00e228e6200e6d]

Apply complete! Resources: 15 added, 0 changed, 0 destroyed.

Outputs:
```



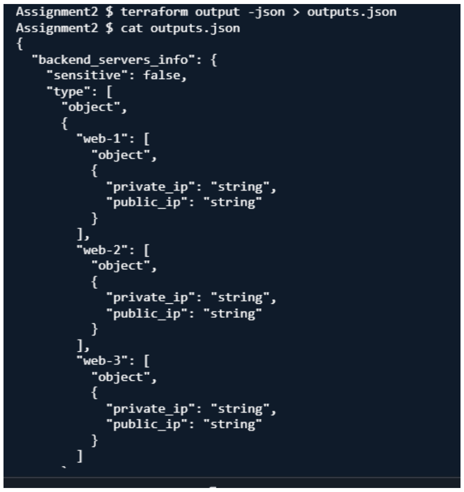
```
Apply complete! Resources: 15 added, 0 changed, 0 destroyed.

Outputs:

backend_servers_info = {
  "web-1" = {
    "private_ip" = "10.0.10.21"
    "public_ip" = "3.238.93.208"
  }
  "web-2" = {
    "private_ip" = "10.0.10.47"
    "public_ip" = "44.201.49.49"
  }
  "web-3" = {
    "private_ip" = "10.0.10.53"
    "public_ip" = "100.48.227.217"
  }
}
nginx_public_ip = "100.48.226.229"
subnet_id = "subnet-0ea857b872b39c331"
vpc_id = "vpc-064f9b1c7aff6e7b9"
Assignment2 $ ls -la ~/.ssh/
total 16
drwxr-xr-x. 2 cloudshell-user cloudshell-user 4096 Dec 29 13:07 .
drwxrwxrwx. 7 cloudshell-user cloudshell-user 4096 Dec 29 13:07 ..
-rw-------. 1 cloudshell-user cloudshell-user 3446 Dec 29 13:07 id_rsa
-rw-r--r--. 1 cloudshell-user cloudshell-user  785 Dec 29 13:07 id_rsa.pub
```



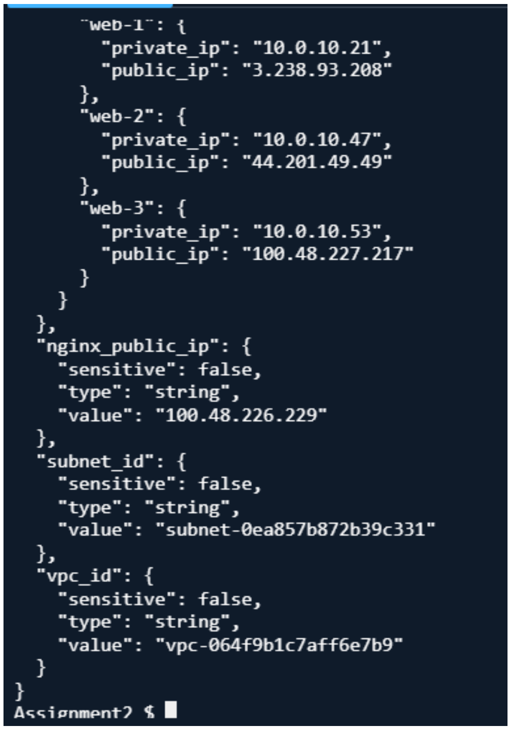
```
Success! The configuration is valid.

Assignment2 $ terraform plan
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 0s [id=https://icanhazip.com]
module.backend_servers["web-1"].aws_key_pair.server_key: Refreshing state... [id=prod-key-1]
module.nginx_server.aws_key_pair.server_key: Refreshing state... [id=prod-key-nginx]
module.backend_servers["web-2"].data.aws_ami.amazon_linux_2023: Reading...
module.networking.aws_vpc.main: Refreshing state... [id=vpc-064f9b1c7aff6e7b9]
module.nginx_server.data.aws_ami.amazon_linux_2023: Reading...
module.backend_servers["web-3"].data.aws_ami.amazon_linux_2023: Reading...
module.backend_servers["web-1"].data.aws_ami.amazon_linux_2023: Reading...
module.backend_servers["web-3"].aws_key_pair.server_key: Refreshing state... [id=prod-key-3]
module.backend_servers["web-2"].aws_key_pair.server_key: Refreshing state... [id=prod-key-2]
module.backend_servers["web-2"].data.aws_ami.amazon_linux_2023: Read complete after 1s [id=ami-068c0051b15cdb816]
module.backend_servers["web-3"].data.aws_ami.amazon_linux_2023: Read complete after 1s [id=ami-068c0051b15cdb816]
module.nginx_server.data.aws_ami.amazon_linux_2023: Read complete after 1s [id=ami-068c0051b15cdb816]
module.backend_servers["web-1"].data.aws_ami.amazon_linux_2023: Read complete after 1s [id=ami-068c0051b15cdb816]
module.networking.aws_internet_gateway.main: Refreshing state... [id=igw-09e1ca0d188b0c2b2]
module.networking.aws_subnet.main: Refreshing state... [id=subnet-0ea857b872b39c331]
module.security.aws_security_group.nginx: Refreshing state... [id=sg-0d0d1bc09edad9d78]
module.networking.aws_route_table.main: Refreshing state... [id=rtb-05527f5b3f03eeeaf]
module.security.aws_security_group.backend: Refreshing state... [id=sg-0dc7a2753ebd1d1ba]
module.nginx_server.aws_instance.server: Refreshing state... [id=i-07f00e228e6200e6d]
module.networking.aws_route_table_association.main: Refreshing state... [id=rtbassoc-0d6e3793bb60acdde]
module.backend_servers["web-2"].aws_instance.server: Refreshing state... [id=i-02961e72f49f19b86]
module.backend_servers["web-3"].aws_instance.server: Refreshing state... [id=i-06fbbb66ebd0e4f21]
module.backend_servers["web-1"].aws_instance.server: Refreshing state... [id=i-0041804fffcebe3a2]

No changes. Your infrastructure matches the configuration.
```



```
Terraform has compared your real infrastru
Assignment2 $ terraform output
backend_servers_info = {
  "web-1" = {
    "private_ip" = "10.0.10.21"
    "public_ip" = "3.238.93.208"
  }
  "web-2" = {
    "private_ip" = "10.0.10.47"
    "public_ip" = "44.201.49.49"
  }
  "web-3" = {
    "private_ip" = "10.0.10.53"
    "public_ip" = "100.48.227.217"
  }
}
nginx_public_ip = "100.48.226.229"
subnet_id = "subnet-0ea857b872b39c331"
vpc_id = "vpc-064f9b1c7aff6e7b9"
Assignment2 $ terraform output -json > out
Assignment2 $ cat outputs.json
{
```

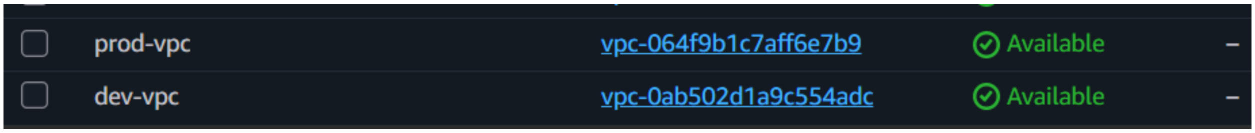


```
Assignment2 $ terraform output -json > outputs.json
Assignment2 $ cat outputs.json
{
  "backend_servers_info": {
    "sensitive": false,
    "type": [
      "object",
      {
        "web-1": [
          "object",
          {
            "private_ip": "string",
            "public_ip": "string"
          }
        ],
        "web-2": [
          "object",
          {
            "private_ip": "string",
            "public_ip": "string"
          }
        ],
        "web-3": [
          "object",
          {
            "private_ip": "string",
            "public_ip": "string"
          }
        ]
```

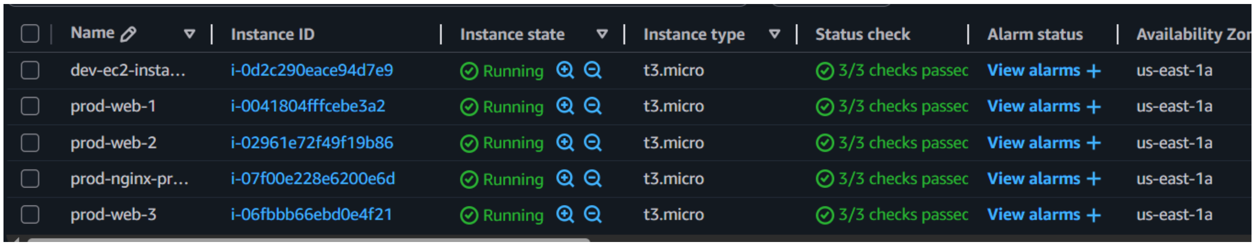


```
      "web-1": {
        "private_ip": "10.0.10.21",
        "public_ip": "3.238.93.208"
      },
      "web-2": {
        "private_ip": "10.0.10.47",
        "public_ip": "44.201.49.49"
      },
      "web-3": {
        "private_ip": "10.0.10.53",
        "public_ip": "100.48.227.217"
      }
    }
  },
  "nginx_public_ip": {
    "sensitive": false,
    "type": "string",
    "value": "100.48.226.229"
  },
  "subnet_id": {
    "sensitive": false,
    "type": "string",
    "value": "subnet-0ea857b872b39c331"
  },
  "vpc_id": {
    "sensitive": false,
    "type": "string",
    "value": "vpc-064f9b1c7aff6e7b9"
  }
}
Assignment2 $
```

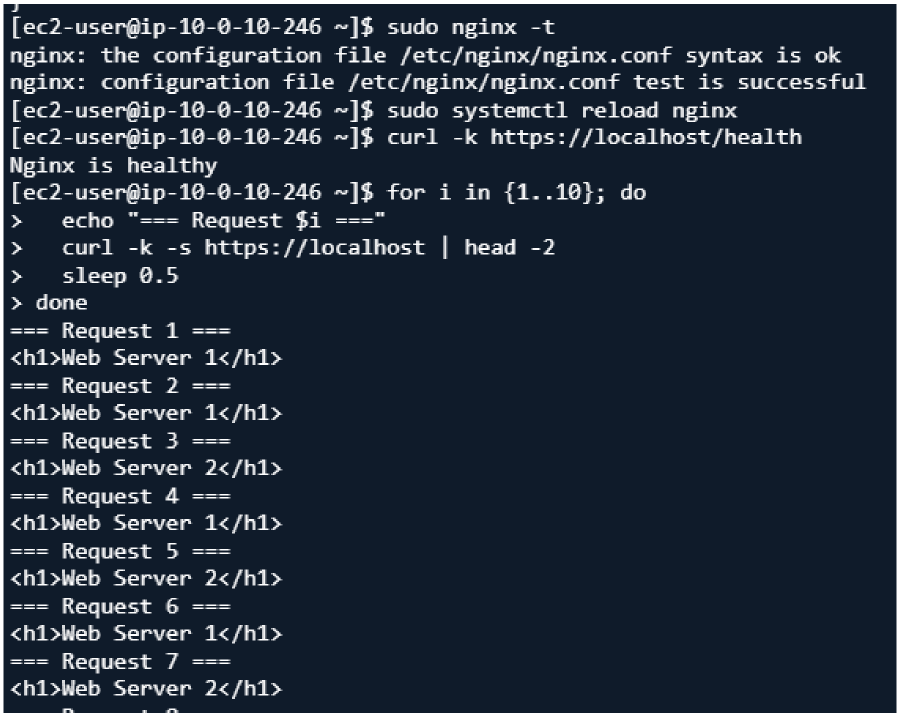
Part 4: Infrastructure Deployment



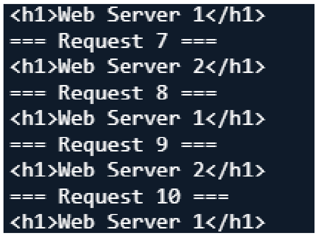
□ prod-vpc	vpc-064f9b1c7aff6e7b9	✓ Available	–
□ dev-vpc	vpc-0ab502d1a9c554adc	✓ Available	–



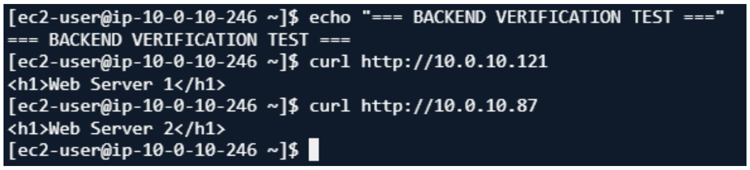
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zor
□	dev-ec2-insta...	i-0d2c290eace94d7e9	✓ Running	t3.micro	✓ 3/3 checks passec	View alarms +	us-east-1a
□	prod-web-1	i-0041804fffcebe3a2	✓ Running	t3.micro	✓ 3/3 checks passec	View alarms +	us-east-1a
□	prod-web-2	i-02961e72f49f19b86	✓ Running	t3.micro	✓ 3/3 checks passec	View alarms +	us-east-1a
□	prod-nginx-pr...	i-07f00e228e6200e6d	✓ Running	t3.micro	✓ 3/3 checks passec	View alarms +	us-east-1a
□	prod-web-3	i-06fbbb66ebd0e4f21	✓ Running	t3.micro	✓ 3/3 checks passec	View alarms +	us-east-1a



```
[ec2-user@ip-10-0-10-246 ~]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-10-0-10-246 ~]$ sudo systemctl reload nginx
[ec2-user@ip-10-0-10-246 ~]$ curl -k https://localhost/health
Nginx is healthy
[ec2-user@ip-10-0-10-246 ~]$ for i in {1..10}; do
>   echo "=== Request $i ==="
>   curl -k -s https://localhost | head -2
>   sleep 0.5
> done
=== Request 1 ===
<h1>Web Server 1</h1>
=== Request 2 ===
<h1>Web Server 1</h1>
=== Request 3 ===
<h1>Web Server 2</h1>
=== Request 4 ===
<h1>Web Server 1</h1>
=== Request 5 ===
<h1>Web Server 2</h1>
=== Request 6 ===
<h1>Web Server 1</h1>
=== Request 7 ===
<h1>Web Server 2</h1>
```

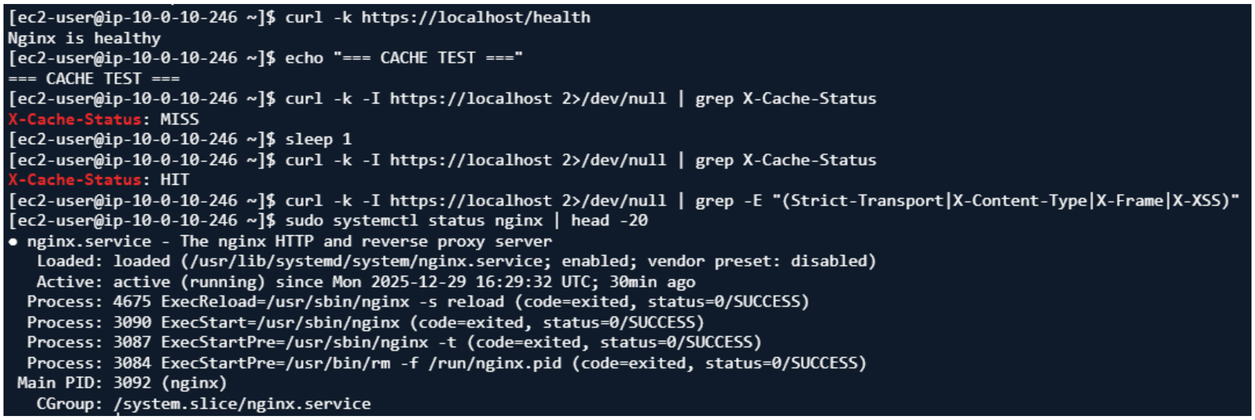


```
<h1>Web Server 1</h1>
=== Request 7 ===
<h1>Web Server 2</h1>
=== Request 8 ===
<h1>Web Server 1</h1>
=== Request 9 ===
<h1>Web Server 2</h1>
=== Request 10 ===
<h1>Web Server 1</h1>
```

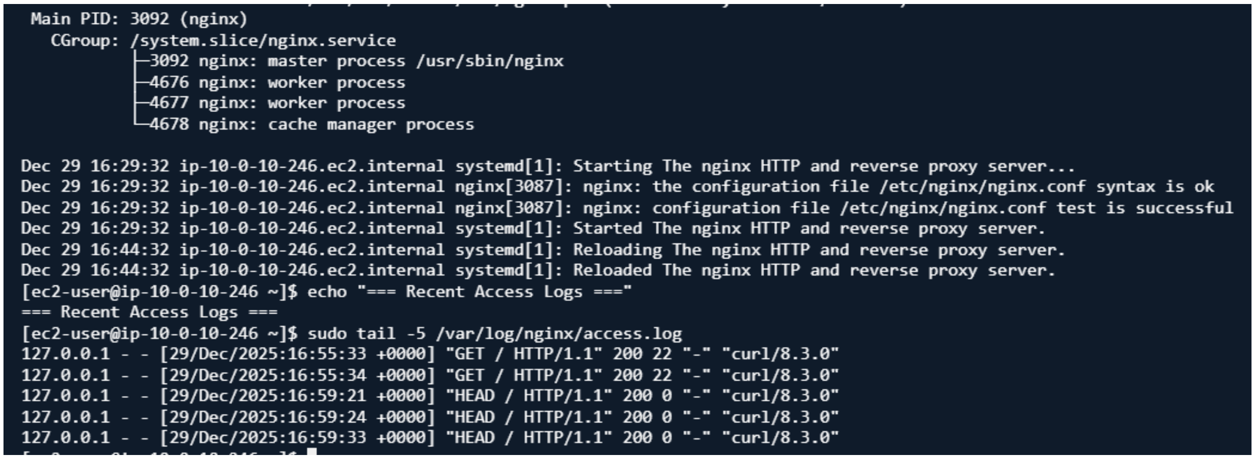


```
[ec2-user@ip-10-0-10-246 ~]$ echo "=== BACKEND VERIFICATION TEST ==="
=== BACKEND VERIFICATION TEST ===
[ec2-user@ip-10-0-10-246 ~]$ curl http://10.0.10.121
<h1>Web Server 1</h1>
[ec2-user@ip-10-0-10-246 ~]$ curl http://10.0.10.87
<h1>Web Server 2</h1>
[ec2-user@ip-10-0-10-246 ~]$ █
```

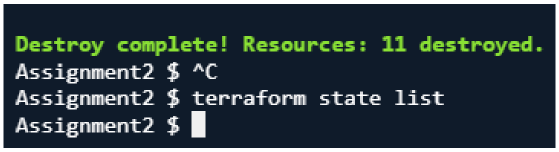
Part 5: Nginx Configuration & Testing



```
[ec2-user@ip-10-0-10-246 ~]$ curl -k https://localhost/health
Nginx is healthy
[ec2-user@ip-10-0-10-246 ~]$ echo "=== CACHE TEST ==="
=== CACHE TEST ===
[ec2-user@ip-10-0-10-246 ~]$ curl -k -I https://localhost 2>/dev/null | grep X-Cache-Status
X-Cache-Status: MISS
[ec2-user@ip-10-0-10-246 ~]$ sleep 1
[ec2-user@ip-10-0-10-246 ~]$ curl -k -I https://localhost 2>/dev/null | grep X-Cache-Status
X-Cache-Status: HIT
[ec2-user@ip-10-0-10-246 ~]$ curl -k -I https://localhost 2>/dev/null | grep -E "(Strict-Transport|X-Content-Type|X-Frame|X-XSS)"
[ec2-user@ip-10-0-10-246 ~]$ sudo systemctl status nginx | head -20
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2025-12-29 16:29:32 UTC; 30min ago
  Process: 4675 ExecReload=/usr/sbin/nginx -s reload (code=exited, status=0/SUCCESS)
  Process: 3090 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
  Process: 3087 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
  Process: 3084 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
 Main PID: 3092 (nginx)
   CGroup: /system.slice/nginx.service
```



```
 Main PID: 3092 (nginx)
   CGroup: /system.slice/nginx.service
           ├─3092 nginx: master process /usr/sbin/nginx
           ├─4676 nginx: worker process
           ├─4677 nginx: worker process
           └─4678 nginx: cache manager process

Dec 29 16:29:32 ip-10-0-10-246.ec2.internal systemd[1]: Starting The nginx HTTP and reverse proxy server...
Dec 29 16:29:32 ip-10-0-10-246.ec2.internal nginx[3087]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Dec 29 16:29:32 ip-10-0-10-246.ec2.internal nginx[3087]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Dec 29 16:29:32 ip-10-0-10-246.ec2.internal systemd[1]: Started The nginx HTTP and reverse proxy server.
Dec 29 16:44:32 ip-10-0-10-246.ec2.internal systemd[1]: Reloading The nginx HTTP and reverse proxy server.
Dec 29 16:44:32 ip-10-0-10-246.ec2.internal systemd[1]: Reloaded The nginx HTTP and reverse proxy server.
[ec2-user@ip-10-0-10-246 ~]$ echo "=== Recent Access Logs ==="
=== Recent Access Logs ===
[ec2-user@ip-10-0-10-246 ~]$ sudo tail -5 /var/log/nginx/access.log
127.0.0.1 - - [29/Dec/2025:16:55:33 +0000] "GET / HTTP/1.1" 200 22 "-" "curl/8.3.0"
127.0.0.1 - - [29/Dec/2025:16:55:34 +0000] "GET / HTTP/1.1" 200 22 "-" "curl/8.3.0"
127.0.0.1 - - [29/Dec/2025:16:59:21 +0000] "HEAD / HTTP/1.1" 200 0 "-" "curl/8.3.0"
127.0.0.1 - - [29/Dec/2025:16:59:24 +0000] "HEAD / HTTP/1.1" 200 0 "-" "curl/8.3.0"
127.0.0.1 - - [29/Dec/2025:16:59:33 +0000] "HEAD / HTTP/1.1" 200 0 "-" "curl/8.3.0"
```



```
Destroy complete! Resources: 11 destroyed.
Assignment2 $ ^C
Assignment2 $ terraform state list
Assignment2 $
```