Report on WordPress Deployment Using Terraform on AWS

1. Background

The objective of this assignment was to deploy a scalable and secure WordPress application using AWS services with Terraform. The aim was to leverage AWS infrastructure, such as ECS (Elastic Container Service), RDS (Relational Database Service), and ALB (Application Load Balancer), while utilizing Infrastructure-as-Code (IaC) principles to automate the deployment and management of these resources.

This project required provisioning a secure and robust architecture that could handle web traffic while ensuring ease of management, scalability, and cost-efficiency. Through the use of Terraform, the infrastructure was defined as code, enabling reproducibility, version control, and flexibility in managing resources. The deployment focused on utilizing ECS Fargate for container orchestration, a load balancer for managing traffic distribution, and an RDS MariaDB database for data storage.

2. Objectives

The main objectives of this assignment were as follows:

- **Deploy WordPress on ECS:** Set up ECS using the Fargate launch type to run WordPress containers without managing the underlying infrastructure.
- **Configure Load Balancer (ALB):** Use an Application Load Balancer to distribute incoming HTTP traffic across multiple ECS containers.
- **Set up RDS Database:** Deploy an RDS instance with MariaDB for data storage and ensure secure connectivity between ECS and RDS.
- **Provision Security Groups:** Create security groups to control access to the ECS service and RDS instance.
- **Implement IAM Roles:** Grant the necessary permissions to the ECS tasks through IAM roles to enable secure resource interactions.
- **Automate the Infrastructure Deployment with Terraform:** Use Terraform to define and manage the entire infrastructure, ensuring a repeatable and scalable deployment.

3. Tasks Completed

To accomplish the objectives, the following tasks were performed:

3.1 ALB (Application Load Balancer)

- An Application Load Balancer (ALB) was provisioned to manage incoming HTTP traffic.
- A target group was created and configured to handle health checks to ensure that traffic was only directed to healthy ECS instances.
- A listener on port 80 was set up to forward traffic to ECS containers based on the health check results.

3.2 ECS Cluster and Task Definition

- An ECS cluster was set up to manage the ECS containers.
- The ECS task definition was created to run the wordpress:latest Docker image using the Fargate launch type. The task was configured with resource specifications (256 CPU units, 512 MB of memory) to ensure efficient operation.
- The task was assigned IAM roles (LabRole) to grant permissions necessary for accessing other AWS resources such as RDS and ALB.

3.3 ECS Service

- The ECS service was configured with Fargate as the launch type, and the appropriate number of tasks was specified.
- The ECS service was linked to the ALB to ensure traffic distribution across all running containers.
- Networking configurations, including subnets and security groups, were implemented to secure and facilitate communication between the ECS containers and other resources.

3.4 RDS MariaDB Instance

- A MariaDB instance was provisioned in Amazon RDS to serve as the database backend for the WordPress application.
- Security groups were configured to allow ECS containers to connect to the RDS instance over the necessary port (3306).

- A database subnet group was used to define the network locations for the RDS instance, ensuring secure and reliable connectivity.

3.5 Security Groups and Networking

- Security groups were created to control inbound and outbound traffic to ECS instances and the RDS database.
- Subnets and VPC configurations were designed to ensure that the resources were deployed in a secure and isolated environment.

3.6 Changes to Terraform Files

To finalize the deployment and ensure the ECS tasks had the necessary permissions to interact with other AWS resources, two key lines were added to the ecs.tf file, which specified the IAM role ARNs for task execution:

execution_role_arn = "arn:aws:iam::178868834374: role/LabRole"
task_role_arn = "arn:aws:iam::178868834374: role/LabRole"

It was my Arn: 78868834374 taken from aws learner's lab. These changes ensured that the ECS tasks could interact with other AWS resources such as RDS and ALB without needing additional IAM policies or roles.

3.7 Output and Access

- The DNS name of the ALB was outputted to facilitate easy access to the WordPress application. This allowed users to access the application via a public URL, pointing to the ECS containers managed by the ALB.

4. Deployment and Testing Results

4.1 Deployment

The deployment process was carried out using Terraform, which successfully created the necessary AWS resources, including the ECS cluster, RDS database, ALB, and security configurations. Terraform's state was used to track the resources created, and all dependencies were managed appropriately.

The ECS service was correctly linked to the ALB, allowing for traffic distribution to the WordPress containers. The MariaDB instance was provisioned and connected to the WordPress application.

4.2 Testing

Once the infrastructure was deployed, the application was tested by accessing the WordPress site via the ALB's DNS name.

- **WordPress Accessibility:** The WordPress site was accessible through the public DNS name of the ALB, indicating that traffic was being correctly routed to the ECS containers.
- **Database Connectivity:** A successful connection was established between the WordPress application and the MariaDB RDS instance, confirming that the database was properly configured and accessible.

No errors or connectivity issues were observed during testing, and the deployment met the expected objectives.

Results:

Deploy the Infrastructure:

1. Use Terraform to deploy the infrastructure on AWS

Chrome File Edit View History Bookmarks Profiles Tab Window Help Sun Mar 30 5:10 PM

us-east-1.console.aws.amazon.com/cloudshell/home?region=us-east-1

CloudShell

us-east-1

```
~ $ terraform version
bash: terraform: command not found
~ $ wget https://releases.hashicorp.com/terraform/1.5.0/terraform_1.5.0_linux_amd64.zip
--2025-03-30 21:06:12--  https://releases.hashicorp.com/terraform/1.5.0/terraform_1.5.0_linux_amd64.zip
Resolving releases.hashicorp.com (releases.hashicorp.com)... 3.171.85.80, 3.171.85.65, 3.171.85.107, ...
Connecting to releases.hashicorp.com (releases.hashicorp.com)|3.171.85.80|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 20957558 (20M) [application/zip]
Saving to: 'terraform_1.5.0_linux_amd64.zip'

terraform_1.5.0_linux_amd64.zip                     100%[===================================================================================>]  19.99M  --.-KB/s    in 0.1s

2025-03-30 21:06:13 (161 MB/s) - 'terraform_1.5.0_linux_amd64.zip' saved [20957558/20957558]

~ $ unzip terraform_1.5.0_linux_amd64.zip
Archive:  terraform_1.5.0_linux_amd64.zip
  inflating: terraform
~ $ sudo mv terraform /usr/local/bin/
~ $ terraform version
Terraform v1.5.0
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.11.3. You can update by downloading from https://www.terraform.io/downloads.html
~ $
```

File upload successful
FP_terraform files.zip was successfully uploaded to the following directory: /home/cloudshell-user.

Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Chrome File Edit View History Bookmarks Profiles Tab Window Help Mon Mar 31 6:37 PM

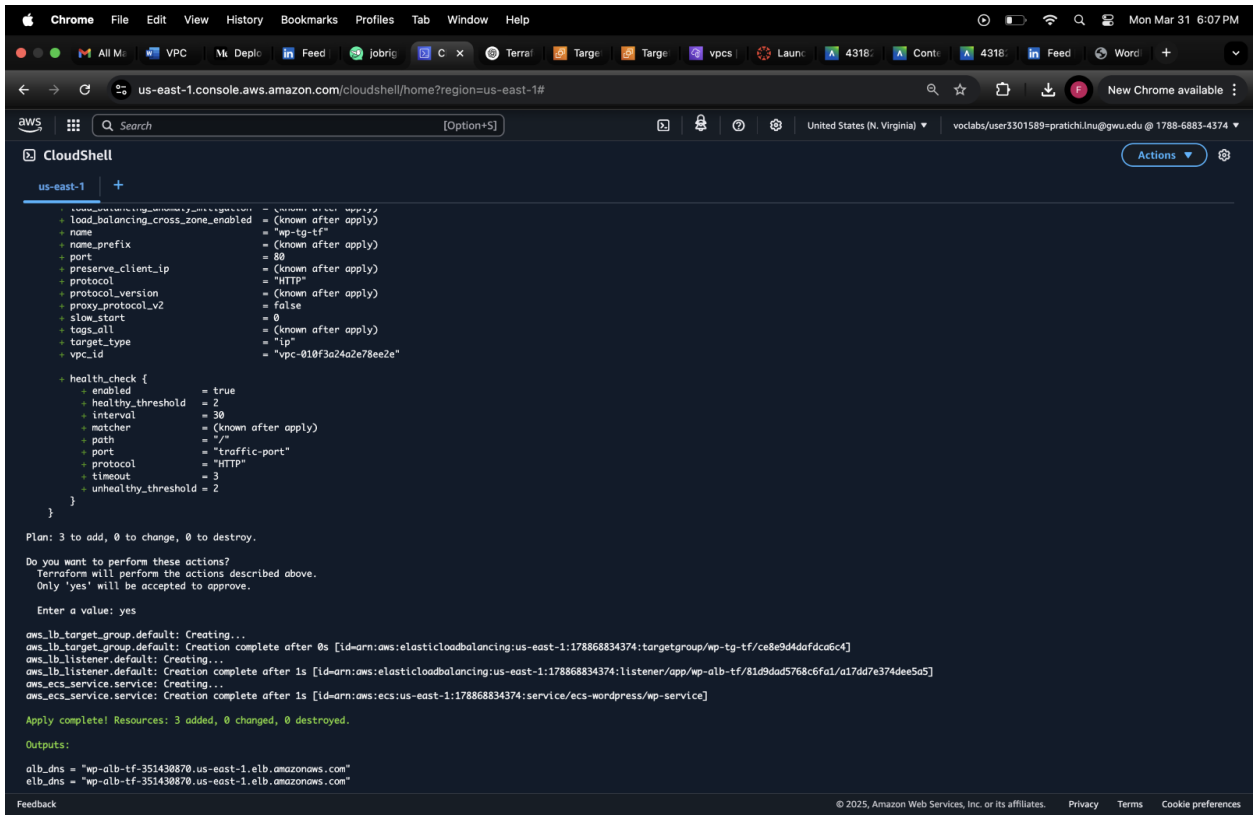
us-east-1.console.aws.amazon.com/cloudshell/home?region=us-east-1#

CloudShell

us-east-1

```
terraform_project $ terraform apply
aws_ecs_cluster.cluster: Refreshing state... [id=arn:aws:ecs:us-east-1:178868834374:cluster/ecs-wordpress]
aws_vpc.default: Refreshing state... [id=vpc-010f3a24a2e78ee2e]
aws_subnet.wp-public-b-tf: Refreshing state... [id=subnet-02883606ddd200fea]
aws_internet_gateway.default: Refreshing state... [id=igw-0887e183a44ffd4c4]
aws_subnet.wp-public-c-tf: Refreshing state... [id=subnet-0b8af61ffec4c4698]
aws_subnet.wp-public-a-tf: Refreshing state... [id=subnet-03b7d296ef1f3b10b]
aws_security_group.wp-alb-tf: Refreshing state... [id=sg-0c57fb8c46dd0073b]
aws_security_group.wp-db-sg-tf: Refreshing state... [id=sg-0d460c37fe02718fc]
aws_route_table.wp-rt-public-tf: Refreshing state... [id=rtb-086e6d0e65bc5612f]
aws_db_subnet_group.default: Refreshing state... [id=wp-db-subnet-tf]
aws_lb.default: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:178868834374:loadbalancer/app/wp-alb-tf/81d9dad5768c6fa1]
aws_route_table_association.b: Refreshing state... [id=rtbassoc-0864e12d364b4f1ef]
aws_route_table_association.a: Refreshing state... [id=rtbassoc-01f83b5f15455a8af]
aws_route_table_association.c: Refreshing state... [id=rtbassoc-00c5cba525156647c]
aws_db_instance.db: Refreshing state... [id=db-NG5KFEWK2WHLZEUTADIAVHT4KY]
data.template_file.wp-container: Reading...
data.template_file.wp-container: Read complete after 0s [id=fa48ce35787e4341458e8dca7d91e6e73b7c4a932622d337ac49c706c43b67cf]
aws_ecs_task_definition.task: Refreshing state... [id=wp-task]

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:

  # aws_ecs_service.service will be created
  + resource "aws_ecs_service" "service" {
      + availability_zone_rebalancing       = "DISABLED"
      + cluster                             = "arn:aws:ecs:us-east-1:178868834374:cluster/ecs-wordpress"
      + deployment_maximum_percent          = 200
      + deployment_minimum_healthy_percent  = 100
      + desired_count                       = 1
      + enable_ecs_managed_tags             = false
      + enable_execute_command              = false
      + iam_role                            = (known after apply)
      + id                                  = (known after apply)
      + launch_type                         = "FARGATE"
      + name                                = "wp-service"
      + platform_version                    = (known after apply)
      + scheduling_strategy                 = "REPLICA"
      + tags_all                            = (known after apply)
      + task_definition                     = "arn:aws:ecs:us-east-1:178868834374:task-definition/wp-task:1"
      + triggers                            = (known after apply)
      + wait_for_steady_state               = false

      + load_balancer {
          + container_name   = "wordpress"
          + container_port   = 80
```

Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



2. Verify that the infrastructure is deployed successfully (e.g., ALB, ECS tasks, RDS instance).

Chrome File Edit View History Bookmarks Profiles Tab Window Help Mon Mar 31 6:58 PM

us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#database:id=my-mariadb-db;is-cluster=false

Aurora and RDS > Databases > my-mariadb-db

Aurora and RDS

- Dashboard
- Databases
- Query Editor
- Performance insights
- Snapshots
- Exports in Amazon S3
- Automated backups
- Reserved instances
- Proxies
- Subnet groups
- Parameter groups
- Option groups
- Custom engine versions
- Zero-ETL integrations New
- Events
- Event subscriptions
- Recommendations 0
- Certificate update

my-mariadb-db

Modify | Actions

Summary

DB identifier	Status	Role	Engine	Recommendations
my-mariadb-db	Available	Instance	MariaDB	
CPU	**Class**	**Current activity**	**Region & AZ**	
2.47%	db.t3.micro	0 Connections	us-east-1a	

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups | Data migrations - *new* | Tags | Recommen

Connectivity & security

Endpoint & port

Endpoint
my-mariadb-db.cbjilmiec79a.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1a

VPC
wp-pvc-tf (vpc-010f3a24a2e78ee2e)

Subnet group
wp-db-subnet-tf

Subnets
subnet-0b8af61ffec4c4698
subnet-028836060dd200fea
subnet-03b7d296ef1f3b10b

Network type
IPv4

Security

VPC security groups
wp-db-tf (sg-0d460c37fe02718fc)
Active

Publicly accessible
No

Certificate authority Info
rds-ca-rsa2048-g1

Certificate authority date
May 25, 2061, 19:34 (UTC-04:00)

DB instance certificate expiration date
March 31, 2026, 17:39 (UTC-04:00)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Chrome File Edit View History Bookmarks Profiles Tab Window Help Mon Mar 31 6:59 PM

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancers:

EC2 > Load balancers

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
  - Key Pairs
  - Network Interfaces
- **Load Balancing**

Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Actions | Create load balancer

Filter load balancers

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
wp-alb-tf	wp-alb-tf-351430870.us-ea...	Active	vpc-010f3a24a2e78ee2e	3 Availability Zones	application	March 31, 2025, 17:36 (...

Load balancer: wp-alb-tf

Details | Listeners and rules | Network mapping | Resource map | Security | Monitoring | Integrations | Attributes | Capacity | Tags

Details

Load balancer type
Application

Scheme
Internet-facing

Status
Active

Hosted zone
Z35SXDOTRQ7X7K

VPC
vpc-010f3a24a2e78ee2e

Availability Zones
subnet-0b8af61ffec4c4698 us-east-1c (use1-az6)
subnet-028836060dd200fea us-east-1b (use1-az4)
subnet-03b7d296ef1f3b10b us-east-1a (use1-az2)

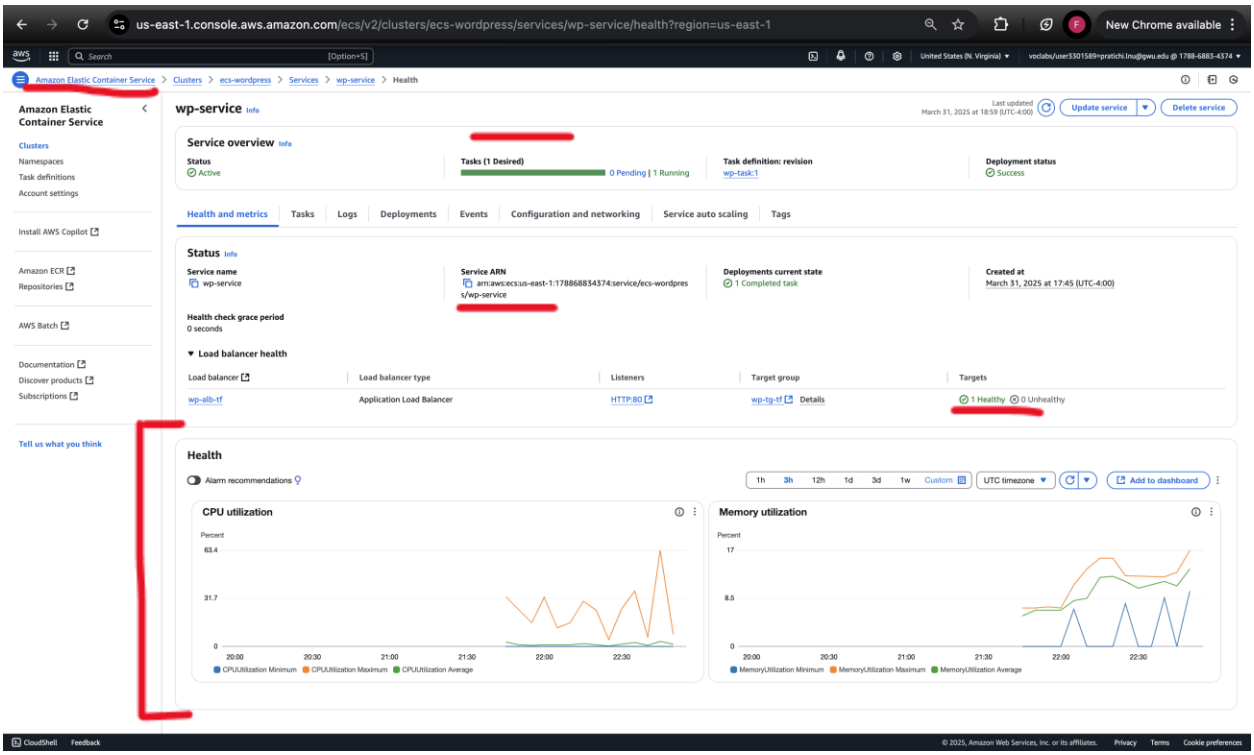
Load balancer IP address type
IPv4

Date created
March 31, 2025, 17:36 (UTC-04:00)

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:178868834374:loadbalancer/app/wp-alb-tf/81d9dad5768c6fa1

DNS name Info
wp-alb-tf-351430870.us-east-1.elb.amazonaws.com (A Record)

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



3. **Log the deployment details**, including date and time, and the output of Terraform commands used for deployment.

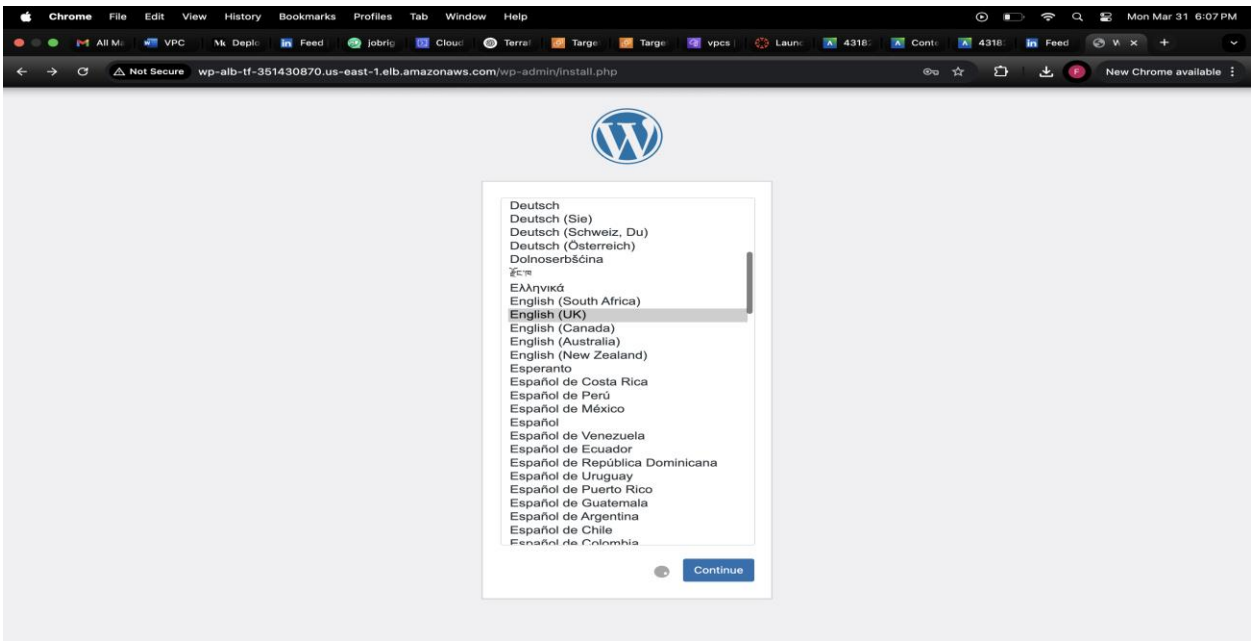
Log file using timestamp: tail -f terraform-deployment-2025-03-31_22-45-39.log

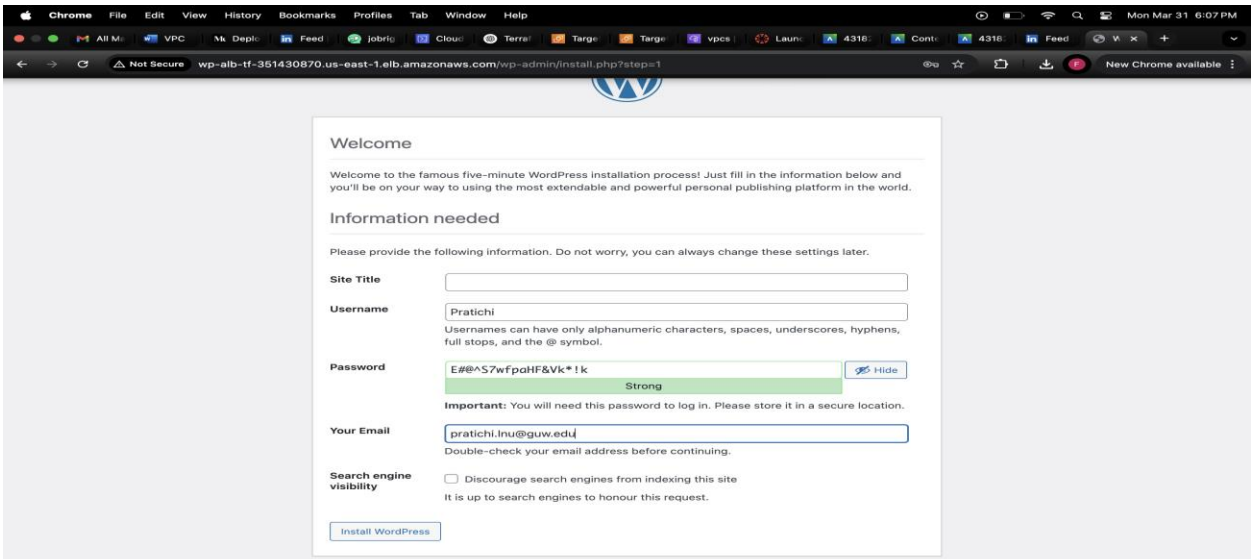
```
us-east-1.console.aws.amazon.com/cloudshell/home?region=us-east-1#
CloudShell
us-east-1
[4]+  Stopped                 tail -f terraform-deployment-2025-03-31_22-45-39.log
terraform_project $ ls
alb.tf  ecs.tf  outputs.tf  rds.tf  security-groups.tf  task-definitions  templates.tf  terraform_apply_output.lo  terraform-deployment-2025-03-31_22-45-39.log  terraform.tfstate  terraform.tfstate.backup  variables.tf
 vpc.tf
terraform_project $ cat terraform-deployment-2025-03-31_22-45-39.log
aws_vpc.default: Refreshing state... [id=vpc-010f3a24a2e78ee2e]
aws_ecs_cluster.cluster: Refreshing state... [id=arn:aws:ecs:us-east-1:178868834374:cluster/ecs-wordpress]
aws_security_group.wp-db-sg-tf: Refreshing state... [id=sg-0d460c37fe02718fc]
aws_internet_gateway.default: Refreshing state... [id=igw-0887e183a44ffd4c4]
aws_subnet.wp-public-a-tf: Refreshing state... [id=subnet-03b7d296ef1f3b10b]
aws_subnet.wp-public-c-tf: Refreshing state... [id=subnet-0b8af61ffec4c4698]
aws_subnet.wp-public-b-tf: Refreshing state... [id=subnet-02883606dd200fea]
aws_lb_target_group.default: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:178868834374:targetgroup/wp-tg-tf/ce8e9d4dafdca6c4]
aws_security_group.wp-alb-tf: Refreshing state... [id=sg-0c57fb8c46dd0073b]
aws_route_table.wp-rt-public-tf: Refreshing state... [id=rtb-086e6d0e65bc5612f]
aws_route_table_association.b: Refreshing state... [id=rtbassoc-0864e12d364b4f1ef]
aws_route_table_association.a: Refreshing state... [id=rtbassoc-01f83b5f15455a8af]
aws_route_table_association.c: Refreshing state... [id=rtbassoc-00c5cba525156647c]
aws_db_subnet_group.default: Refreshing state... [id=wp-db-subnet-tf]
aws_lb.default: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:178868834374:loadbalancer/app/wp-alb-tf/81d9dad5768c6fa1]
aws_lb_listener.default: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:178868834374:listener/app/wp-alb-tf/81d9dad5768c6fa1/a17dd7e374dee5a5]
aws_db_instance.db: Refreshing state... [id=db-NG5KFEWK2WHLZEUTADIAVHT4KY]
data.template_file.wp-container: Reading...
data.template_file.wp-container: Read complete after 0s [id=fa48ce35787e4341458e8dca7d91e6e73b7c4a932622d337ac49c706c43b67cf]
aws_ecs_task_definition.task: Refreshing state... [id=wp-task]
aws_ecs_service.service: Refreshing state... [id=arn:aws:ecs:us-east-1:178868834374:service/ecs-wordpress/wp-service]

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
  ~ update in-place

Terraform will perform the following actions:

  # aws_lb_target_group.default will be updated in-place
  ~ resource "aws_lb_target_group" "default" {
      id                   = "arn:aws:elasticloadbalancing:us-east-1:178868834374:targetgroup/wp-tg-tf/ce8e9d4dafdca6c4"
      name                 = "wp-tg-tf"
      tags                 = {}
      # (17 unchanged attributes hidden)

    ~ health_check {
        ~ path              = "/index.php" -> "/"
          # (8 unchanged attributes hidden)
      }

      # (4 unchanged blocks hidden)
  }
```

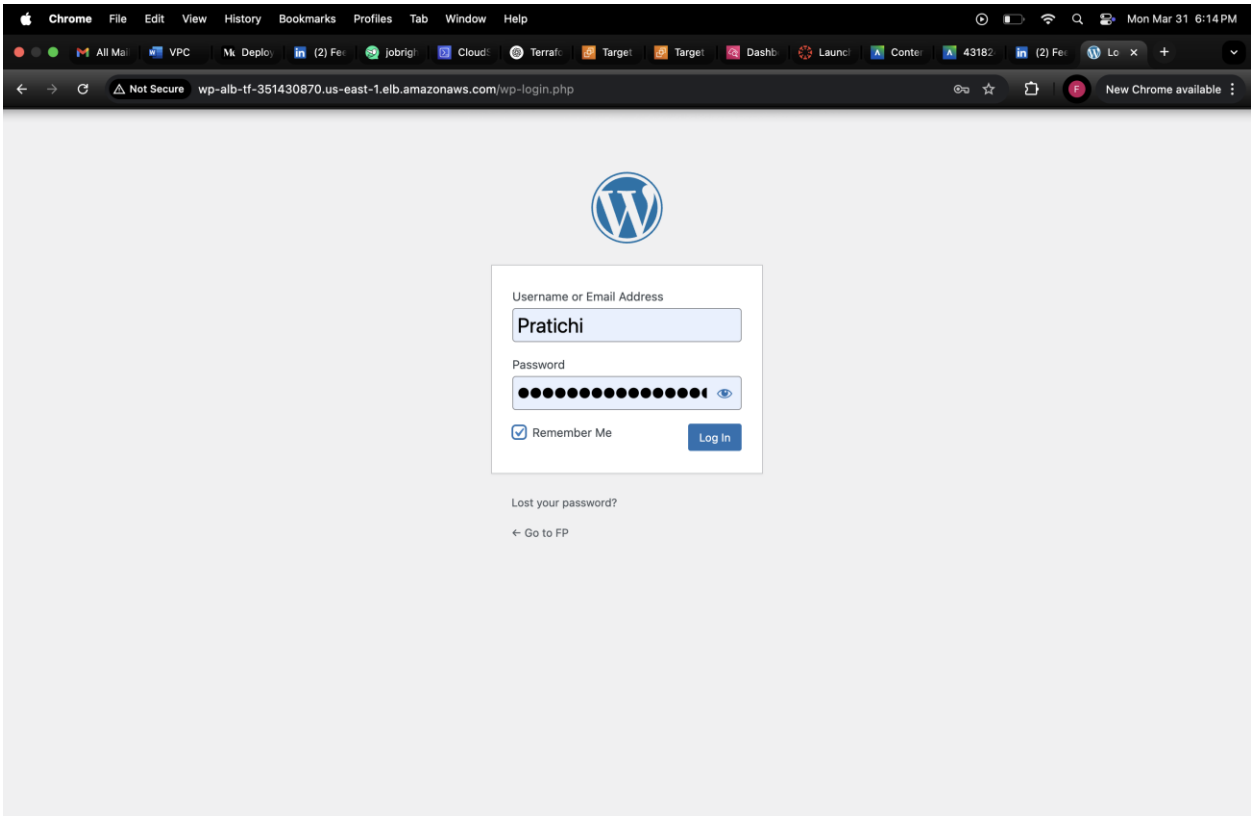
```
Chrome File Edit View History Bookmarks Profiles Tab Window Help   Mon Mar 31 6:50 PM
us-east-1.console.aws.amazon.com/cloudshell/home?region=us-east-1#
CloudShell
us-east-1
aws_db_subnet_group.default: Refreshing state... [id=wp-db-subnet-tf]
aws_lb.default: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:178868834374:loadbalancer/app/wp-alb-tf/81d9dad5768c6fa1]
aws_lb_listener.default: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:178868834374:listener/app/wp-alb-tf/81d9dad5768c6fa1/a17dd7e374dee5a5]
aws_db_instance.db: Refreshing state... [id=db-NG5KFEWK2WHLZEUTADIAVHT4KY]
data.template_file.wp-container: Reading...
data.template_file.wp-container: Read complete after 0s [id=fa48ce35787e4341458e8dca7d91e6e73b7c4a932622d337ac49c706c43b67cf]
aws_ecs_task_definition.task: Refreshing state... [id=wp-task]
aws_ecs_service.service: Refreshing state... [id=arn:aws:ecs:us-east-1:178868834374:service/ecs-wordpress/wp-service]

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
  ~ update in-place

Terraform will perform the following actions:

  # aws_lb_target_group.default will be updated in-place
  ~ resource "aws_lb_target_group" "default" {
      id                   = "arn:aws:elasticloadbalancing:us-east-1:178868834374:targetgroup/wp-tg-tf/ce8e9d4dafdca6c4"
      name                 = "wp-tg-tf"
      tags                 = {}
      # (17 unchanged attributes hidden)

    ~ health_check {
        ~ path              = "/index.php" -> "/"
          # (8 unchanged attributes hidden)
      }

      # (4 unchanged blocks hidden)
  }

Plan: 0 to add, 1 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:
aws_lb_target_group.default: Modifying... [id=arn:aws:elasticloadbalancing:us-east-1:178868834374:targetgroup/wp-tg-tf/ce8e9d4dafdca6c4]
aws_lb_target_group.default: Modifications complete after 1s [id=arn:aws:elasticloadbalancing:us-east-1:178868834374:targetgroup/wp-tg-tf/ce8e9d4dafdca6c4]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.

Outputs:

alb_dns = "wp-alb-tf-351430870.us-east-1.elb.amazonaws.com"
elb_dns = "wp-alb-tf-351430870.us-east-1.elb.amazonaws.com"
terraform_project $
```

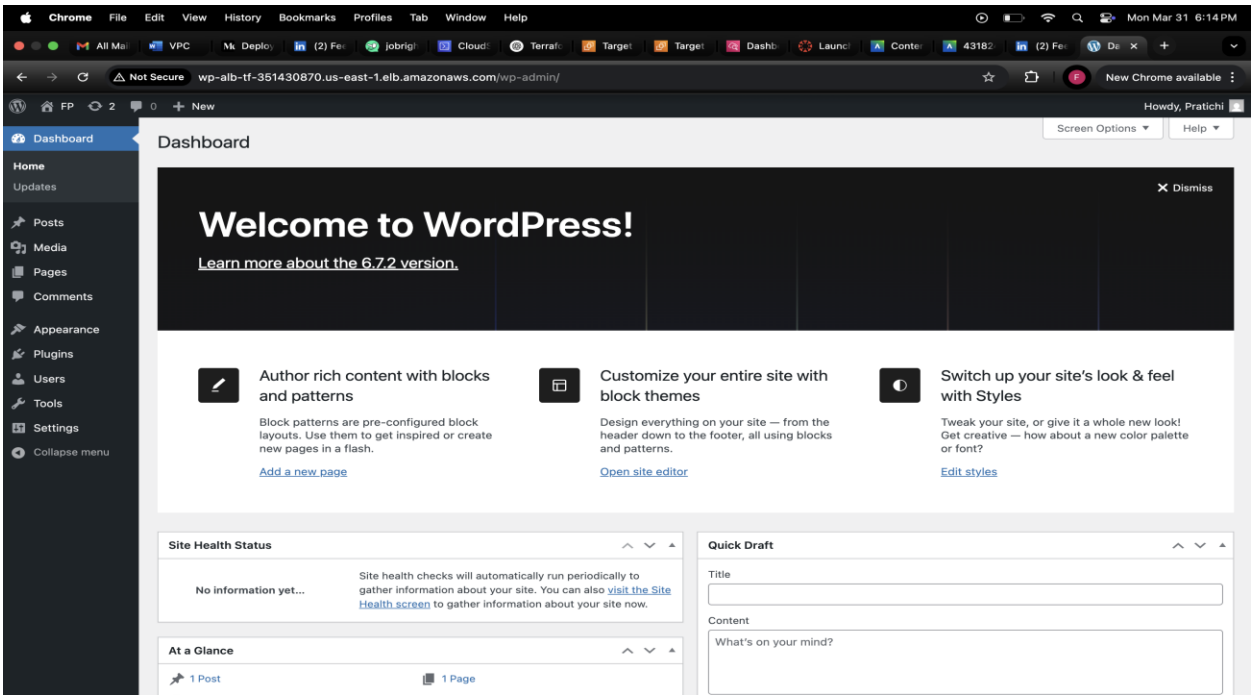
Test the WordPress Application:

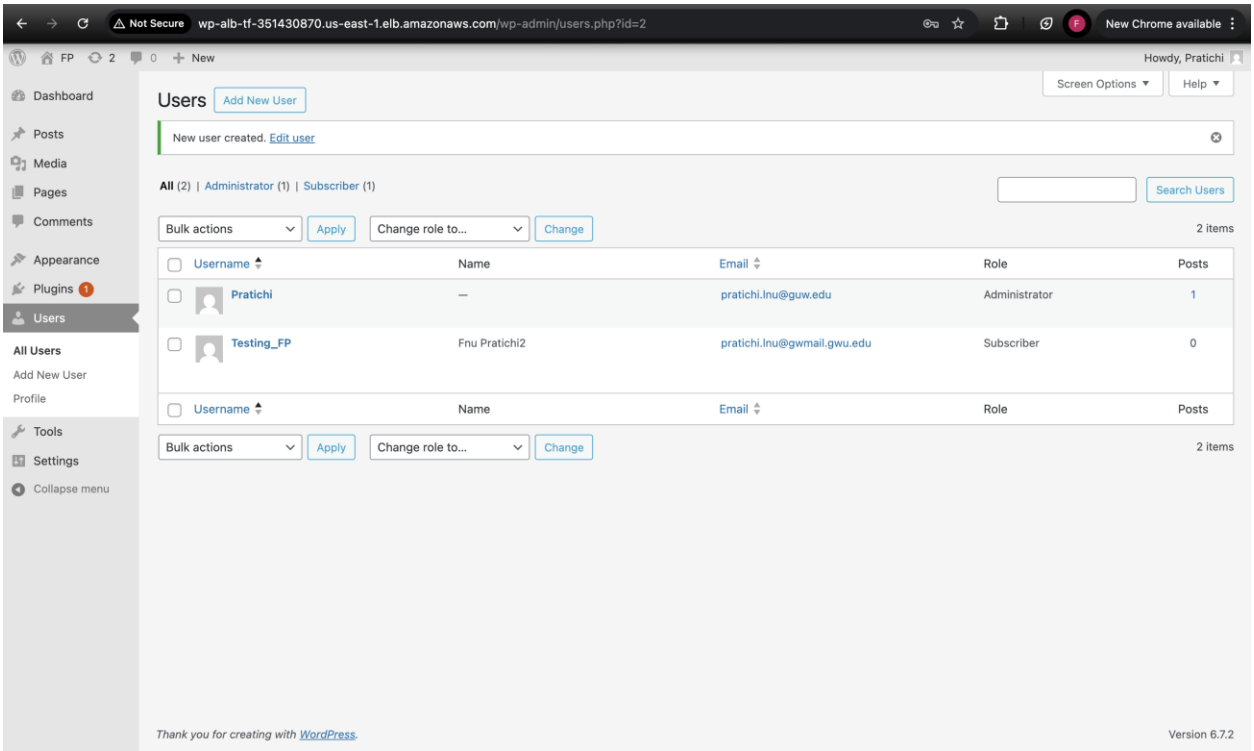
1. Use the ALB DNS name to access the WordPress application via a web browser.
2. Ensure the application is functional and properly connected to the RDS database.



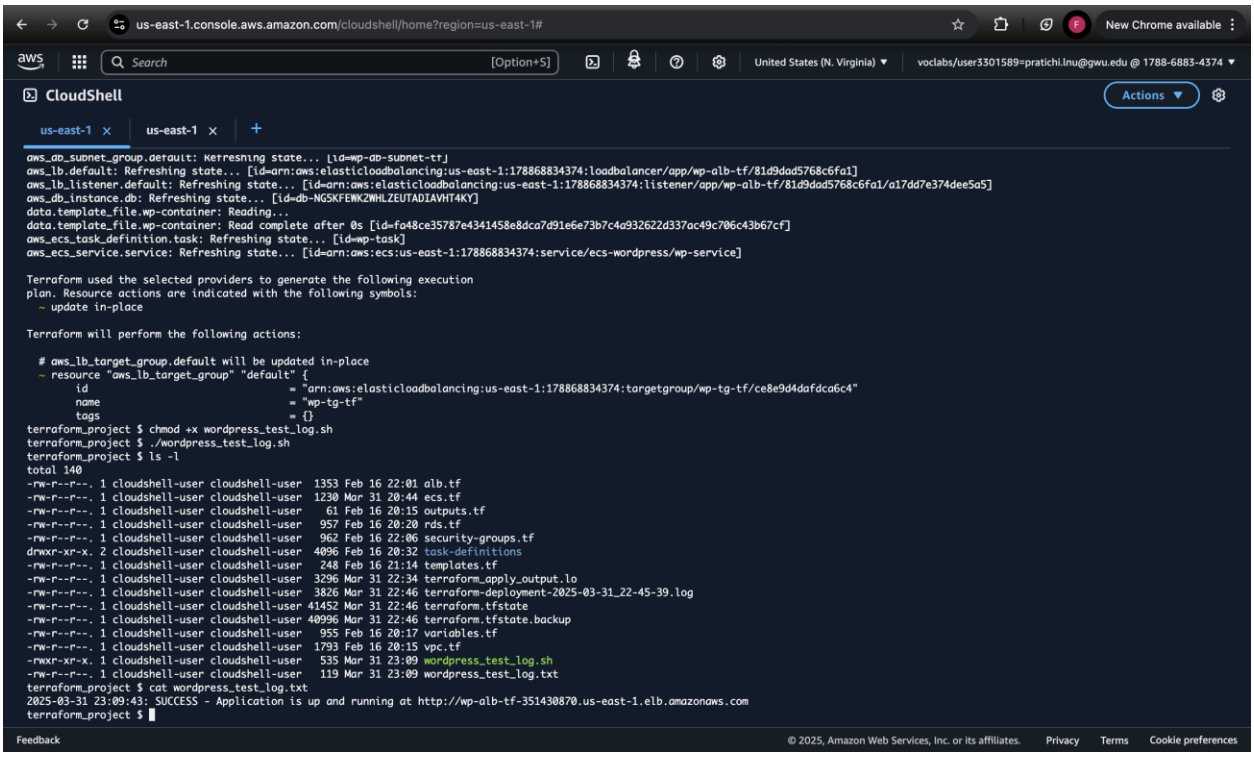

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon Mar 31 6:07 PM
Not Secure
wp-alb-tf-351430870.us-east-1.elb.amazonaws.com/wp-admin/install.php?step=1
New Chrome available
Welcome
Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.
Information needed
Please provide the following information. Do not worry, you can always change these settings later.
Site Title
Username
Pratichi
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, full stops, and the @ symbol.
Password
E#@^S7wfpaHF&Vk*!k
Hide
Strong
Important: You will need this password to log in. Please store it in a secure location.
Your Email
pratichi.lnu@guw.edu
Double-check your email address before continuing.
Search engine visibility
Discourage search engines from indexing this site
It is up to search engines to honour this request.
Install WordPress


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon Mar 31 6:14 PM
Not Secure
wp-alb-tf-351430870.us-east-1.elb.amazonaws.com/wp-login.php
New Chrome available
Username or Email Address
Pratichi
Password
Remember Me
Log In
Lost your password?
← Go to FP





4. **Log the results**, noting any issues or successful tests along with the time and date.



```
us-east-1.console.aws.amazon.com/cloudshell/home?region=us-east-1#
CloudShell
us-east-1
us-east-1
aws_db_subnet_group.default: Refreshing state... [id=wp-db-subnet-tf]
aws_lb.default: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:178868834374:loadbalancer/app/wp-alb-tf/81d9dad5768c6fa1]
aws_lb_listener.default: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:178868834374:listener/app/wp-alb-tf/81d9dad5768c6fa1/a17dd7e374dee5a5]
aws_db_instance.db: Refreshing state... [id=db-NG5KFEWK2WHLZEUTADIAVHT4KY]
data.template_file.wp-container: Reading...
data.template_file.wp-container: Read complete after 0s [id=fa48ce35787e4341458e8dca7d91e6e73b7c4a932622d337ac49c706c43b67cf]
aws_ecs_task_definition.task: Refreshing state... [id=wp-task]
aws_ecs_service.service: Refreshing state... [id=arn:aws:ecs:us-east-1:178868834374:service/ecs-wordpress/wp-service]

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
  ~ update in-place

Terraform will perform the following actions:

  # aws_lb_target_group.default will be updated in-place
  ~ resource "aws_lb_target_group" "default" {
        id                                 = "arn:aws:elasticloadbalancing:us-east-1:178868834374:targetgroup/wp-tg-tf/ce8e9d4dafdca6c4"
        name                               = "wp-tg-tf"
        tags                               = {}
terraform_project $ chmod +x wordpress_test_log.sh
terraform_project $ ./wordpress_test_log.sh
terraform_project $ ls -l
total 140
-rw-r--r--. 1 cloudshell-user cloudshell-user  1353 Feb 16 22:01 alb.tf
-rw-r--r--. 1 cloudshell-user cloudshell-user  1230 Mar 31 20:44 ecs.tf
-rw-r--r--. 1 cloudshell-user cloudshell-user    61 Feb 16 20:15 outputs.tf
-rw-r--r--. 1 cloudshell-user cloudshell-user   957 Feb 16 20:20 rds.tf
-rw-r--r--. 1 cloudshell-user cloudshell-user   962 Feb 16 22:06 security-groups.tf
drwxr-xr-x. 2 cloudshell-user cloudshell-user  4096 Feb 16 20:32 task-definitions
-rw-r--r--. 1 cloudshell-user cloudshell-user   248 Feb 16 21:14 templates.tf
-rw-r--r--. 1 cloudshell-user cloudshell-user  3296 Mar 31 22:34 terraform_apply_output.lo
-rw-r--r--. 1 cloudshell-user cloudshell-user  3826 Mar 31 22:46 terraform-deployment-2025-03-31_22-45-39.log
-rw-r--r--. 1 cloudshell-user cloudshell-user 41452 Mar 31 22:46 terraform.tfstate
-rw-r--r--. 1 cloudshell-user cloudshell-user 40996 Mar 31 22:46 terraform.tfstate.backup
-rw-r--r--. 1 cloudshell-user cloudshell-user   955 Feb 16 20:17 variables.tf
-rw-r--r--. 1 cloudshell-user cloudshell-user  1793 Feb 16 20:15 vpc.tf
-rwxr-xr-x. 1 cloudshell-user cloudshell-user   535 Mar 31 23:09 wordpress_test_log.sh
-rw-r--r--. 1 cloudshell-user cloudshell-user   119 Mar 31 23:09 wordpress_test_log.txt
terraform_project $ cat wordpress_test_log.txt
2025-03-31 23:09:43: SUCCESS - Application is up and running at http://wp-alb-tf-351430870.us-east-1.elb.amazonaws.com
terraform_project $
```

Destroy the Infrastructure:

terraform-destroy-2025-04-01_00-26-43.log

us-east-1.console.aws.amazon.com/cloudshell/home?region=us-east-1

New Chrome available

aws | Search [Option+S] | United States (N. Virginia) | voclabs/user3301589=pratichi.lnu@gwu.edu @ 1788-6883-4374

CloudShell | Actions

us-east-1 | us-east-1

```
terraform_project $ ls
alb.tf              terraform_1.5.6_linux_amd64.zip              terraform-destroy-2025-03-31_23-39-53.log  terraform-destroy-2025-03-31_23-58-01.log       terraform.tfstate.backup
ecs.tf              terraform_apply_output.lo                    terraform-destroy-2025-03-31_23-40-44.log  terraform-destroy-2025-04-01_00-02-02.log       variables.tf
outputs.tf          terraform-deployment-2025-03-31_22-45-39.log terraform-destroy-2025-03-31_23-41-18.log  terraform-destroy-2025-04-01_00-20-53.log       vpc.tf
rds.tf              terraform-deployment-2025-03-31_23-19-13.log terraform-destroy-2025-03-31_23-44-14.log  terraform-destroy-2025-04-01_00-26-43.log       wordpress_test_log.sh
security-groups.tf  terraform-destroy-2025-03-31_23-19-44.log   terraform-destroy-2025-03-31_23-46-55.log  terraform-destroy-2025-04-01_00-29-59.log       wordpress_test_log.txt
task-definitions    terraform-destroy-2025-03-31_23-21-13.log   terraform-destroy-2025-03-31_23-48-46.log  terraform-plan-destroy-2025-04-01_00-26-06.log
templates.tf        terraform-destroy-2025-03-31_23-23-18.log   terraform-destroy-2025-03-31_23-49-33.log  terraform.tfstate
terraform_project $ cat terraform-destroy-2025-04-01_00-26-43.log
aws_vpc.default: Refreshing state... [id=vpc-010f3a24a2e78ee2e]
aws_internet_gateway.default: Refreshing state... [id=igw-0887e183a44ffd4c4]
aws_subnet.wp-public-c-tf: Refreshing state... [id=subnet-0b8af61ffec4c4698]
aws_security_group.wp-db-sg-tf: Refreshing state... [id=sg-0d460c37fe02718fc]
aws_security_group.wp-alb-tf: Refreshing state... [id=sg-0c57fb8c46dd0073b]
aws_subnet.wp-public-a-tf: Refreshing state... [id=subnet-03b7d296ef1f3b10b]
aws_subnet.wp-public-b-tf: Refreshing state... [id=subnet-028836060dd200fea]
aws_db_subnet_group.default: Refreshing state... [id=wp-db-subnet-tf]
aws_lb.default: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:178868834374:loadbalancer/app/wp-alb-tf/81d9dad5768c6fa1]
aws_db_instance.db: Refreshing state... [id=db-NG5KFEWK2WHLZEUTADIAVHT4KY]

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

Terraform will perform the following actions:

  # aws_db_subnet_group.default will be destroyed
  - resource "aws_db_subnet_group" "default" {
      - arn                    = "arn:aws:rds:us-east-1:178868834374:subgrp:wp-db-subnet-tf" -> null
      - description            = "VPC Subnets" -> null
      - id                     = "wp-db-subnet-tf" -> null
      - name                   = "wp-db-subnet-tf" -> null
      - subnet_ids             = [
          - "subnet-028836060dd200fea",
          - "subnet-03b7d296ef1f3b10b",
          - "subnet-0b8af61ffec4c4698",
        ] -> null
      - supported_network_types = [
          - "IPV4",
        ] -> null
      - tags                   = {} -> null
      - tags_all               = {} -> null
```

Feedback | © 2025, Amazon Web Services, Inc. or its affiliates. | Privacy | Terms | Cookie preferences

us-east-1.console.aws.amazon.com/cloudshell/home?region=us-east-1

New Chrome available

aws | Search [Option+S] | United States (N. Virginia) | voclabs/user3301589=pratichi.lnu@gwu.edu @ 1788-6883-4374

CloudShell | Actions

us-east-1 | us-east-1

```
      - assign_generated_ipv6_cidr_block     = false -> null
      - cidr_block                           = "10.0.0.0/16" -> null
      - default_network_acl_id               = "acl-015687e2409353369" -> null
      - default_route_table_id               = "rtb-07be962cfda2fefcd" -> null
      - default_security_group_id            = "sg-07a30c444af02f8fa" -> null
      - dhcp_options_id                      = "dopt-039b80fd214138d9c" -> null
      - enable_dns_hostnames                 = true -> null
      - enable_dns_support                   = true -> null
      - enable_network_address_usage_metrics = false -> null
      - id                                   = "vpc-010f3a24a2e78ee2e" -> null
      - instance_tenancy                     = "default" -> null
      - ipv6_netmask_length                  = 0 -> null
      - main_route_table_id                  = "rtb-07be962cfda2fefcd" -> null
      - owner_id                             = "178868834374" -> null
      - tags                                 = {
          - "Name" = "wp-pvc-tf"
        } -> null
      - tags_all                             = {
          - "Name" = "wp-pvc-tf"
        } -> null
    }

Plan: 0 to add, 0 to change, 8 to destroy.
aws_internet_gateway.default: Destroying... [id=igw-0887e183a44ffd4c4]
aws_db_subnet_group.default: Destroying... [id=wp-db-subnet-tf]
aws_security_group.wp-db-sg-tf: Destroying... [id=sg-0d460c37fe02718fc]
aws_security_group.wp-alb-tf: Destroying... [id=sg-0c57fb8c46dd0073b]
aws_db_subnet_group.default: Destruction complete after 0s
aws_subnet.wp-public-b-tf: Destroying... [id=subnet-028836060dd200fea]
aws_subnet.wp-public-c-tf: Destroying... [id=subnet-0b8af61ffec4c4698]
aws_subnet.wp-public-a-tf: Destroying... [id=subnet-03b7d296ef1f3b10b]
aws_internet_gateway.default: Destruction complete after 0s
aws_security_group.wp-db-sg-tf: Destruction complete after 0s
aws_subnet.wp-public-c-tf: Destruction complete after 0s
aws_subnet.wp-public-b-tf: Destruction complete after 0s
aws_subnet.wp-public-a-tf: Destruction complete after 0s
aws_security_group.wp-alb-tf: Destruction complete after 0s
aws_vpc.default: Destroying... [id=vpc-010f3a24a2e78ee2e]
aws_vpc.default: Destruction complete after 1s

Destroy complete! Resources: 8 destroyed.
terraform_project $
```

Feedback | © 2025, Amazon Web Services, Inc. or its affiliates. | Privacy | Terms | Cookie preferences