

## Zone1 – Terraform apply:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  CODEWHISPERER REFERENCE LOG

module.project_rds_p.aws_rds_cluster_instance.udacity_instance[0]: Still creating... [8m40s elapsed]
module.project_rds_p.aws_rds_cluster_instance.udacity_instance[0]: Still creating... [8m50s elapsed]
module.project_rds_p.aws_rds_cluster_instance.udacity_instance[0]: Still creating... [9m0s elapsed]
module.project_rds_p.aws_rds_cluster_instance.udacity_instance[0]: Still creating... [9m10s elapsed]
module.project_rds_p.aws_rds_cluster_instance.udacity_instance[0]: Still creating... [9m20s elapsed]
module.project_rds_p.aws_rds_cluster_instance.udacity_instance[0]: Still creating... [9m30s elapsed]
module.project_rds_p.aws_rds_cluster_instance.udacity_instance[0]: Still creating... [9m40s elapsed]
module.project_rds_p.aws_rds_cluster_instance.udacity_instance[0]: Still creating... [9m50s elapsed]
module.project_rds_p.aws_rds_cluster_instance.udacity_instance[0]: Still creating... [10m0s elapsed]
module.project_rds_p.aws_rds_cluster_instance.udacity_instance[0]: Creation complete after 10m6s [id=udacity-db-instance-0]

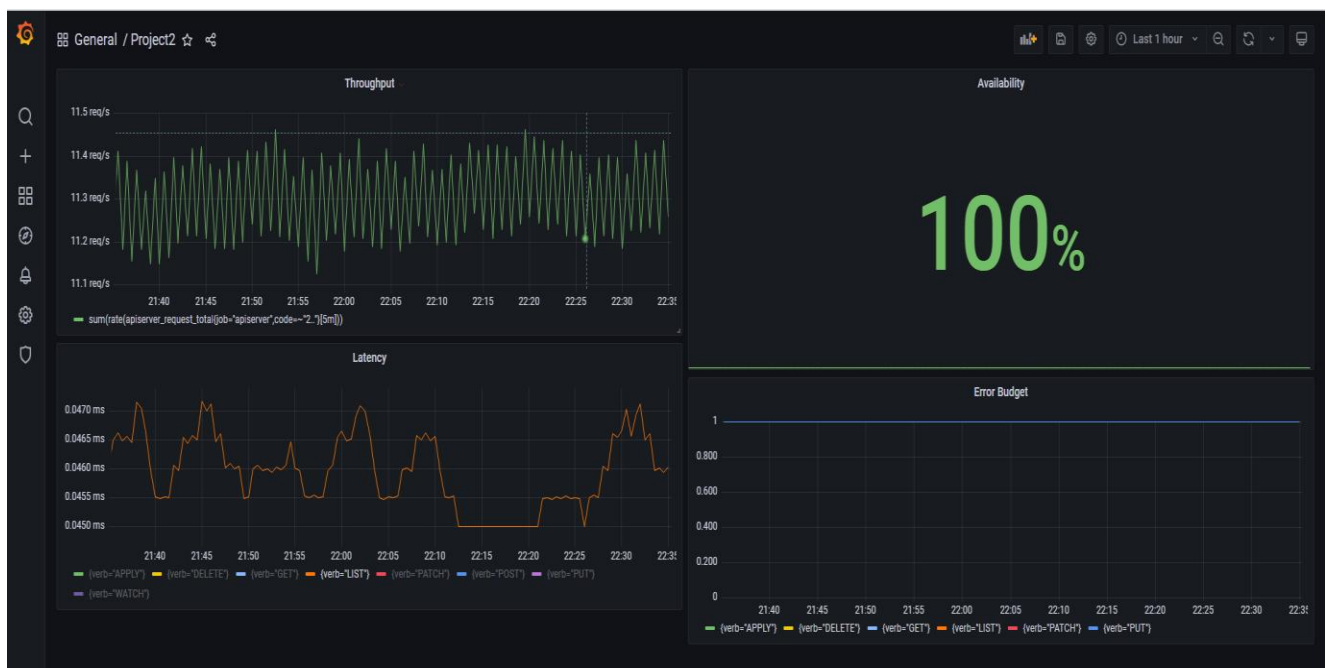
Apply complete! Resources: 3 added, 0 changed, 2 destroyed.

Outputs:

account_id = "661034656910"
caller_arn = "arn:aws:iam::661034656910:user/myAdmin"
caller_user = "AIDA7T2F4MSHF2FYF2UPT"
private_subnet_ids = [
  "subnet-09dcbedeb85b0e6b0",
  "subnet-09e4094669c3f94b2",
]
public_subnet_ids = [
  "subnet-0d45ed1143de9b3c9",
  "subnet-001103c87450ebde6",
]
vpc_id = "vpc-0c9c4907cb957ec4b"

Vinoth@Vinoth MINGW64 ~/Desktop/Udacity/08-SRE/02-HA_DR/Planning-for-High-Availability-and-Incident-Response/project/starter-code/zone1 (master)
$ []
```

## Grafana Screenshot:



## Zone2 – Terraform apply:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL CODEWHISPERER REFERENCE LOG

+ target_port = "3000"
}

+ session_affinity_config {
+   client_ip {
+     timeout_seconds = (known after apply)
+   }
+ }
}

}

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

kubernetes_service.grafana-external: Creating...
kubernetes_service.grafana-external: Creation complete after 3s [id=monitoring/grafana-external]

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

Outputs:

account_id = "661034656910"
caller_arn = "arn:aws:iam::661034656910:user/myAdmin"
caller_user = "AIDAZT2F4MSHF2FYF2UPT"

Vinoth@Vinoth MINGW64 ~/Desktop/Udacity/08-SRE/02-HA_DR/Planning-for-High-Availability-and-Incident-Response/project/starter-code/zone2 (master)
$
```

## LB – Zone2

The screenshot displays the AWS Management Console interface. On the left, a navigation pane shows various services, with 'Load Balancing' selected under the 'Network & Security' category. The main content area shows the 'Load balancers' page for the 'us-west-1' region. A table lists the available load balancers, with one instance selected. Below the table, the 'Details' tab for the selected load balancer is expanded, showing its configuration.

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created	Instance ID
a4f254a4bdcf84792bfb99604a5b4b00	a4f254a4bdcf84792bfb99604a5b4b00...	Active	vpc-0ccdcbedb629312	2 Availability Zones	network	March 18, 2023, 23:21 (UTC+05:30)	-

Load balancer: a4f254a4bdcf84792bfb99604a5b4b00			
Details			
Load balancer type Network	DNS name a4f254a4bdcf84792bfb99604a5b4b00-22375e0580f-cdb0a.elb.us-west-1.amazonaws.com (A Record)	Status Active	VPC vpc-0ccdcbedb629312
IP address type IPv4	Scheme Internet-facing	Availability Zones subnet-081aaf6fe860e90e5 us-west-1c (usw1-az3) subnet-06f21e00775ee4fd us-west-1a (usw1-az1)	Hosted zone Z24KFLUXS084VW

DB – High Availability Deployment:

Terraform Run:

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** CODEWHISPERER REFERENCE LOG

```
module.project_rds_s.aws_rds_cluster_instance.udacity_instance-s[1]: Still creating... [15m31s elapsed]
module.project_rds_s.aws_rds_cluster_instance.udacity_instance-s[1]: Still creating... [15m41s elapsed]
module.project_rds_s.aws_rds_cluster_instance.udacity_instance-s[1]: Still creating... [15m51s elapsed]
module.project_rds_s.aws_rds_cluster_instance.udacity_instance-s[1]: Still creating... [16m1s elapsed]
module.project_rds_s.aws_rds_cluster_instance.udacity_instance-s[1]: Creation complete after 16m6s [id=udacity-db-instance-1-s]

Apply complete! Resources: 6 added, 0 changed, 3 destroyed.

Outputs:

account_id = "661034656910"
caller_arn = "arn:aws:iam::661034656910:user/myAdmin"
caller_user = "AIDAZT2F4MSHF2FYF2UPT"
private_subnet_ids = [
  "subnet-06d1e6c094605e45f",
  "subnet-035446ac01a34f3ee",
]
public_subnet_ids = [
  "subnet-06f21e007775ee4fd",
  "subnet-081aa6ffe860e90e5",
]
vpc_id = "vpc-0ccdcbcdeb629312"

Vinoth@Vinoth MINGW64 ~/Desktop/Udacity/08-SRE/02-HA_DR/Planning-for-High-Availability-and-Incident-Response/project/starter-code/zone1 (master)
$
```

Zone1:

aws

Services

Q Search

[Alt+S]

Ohio

voclabs/user1419518+e821d1ee-bd23-11e9-a26d-af17923f5d7e @ 66...

Amazon 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

Events

Event subscriptions

Recommendations

Certificate update

RDS > Databases

Upgrade required for your database

You can manually upgrade your database, or RDS will automatically upgrade it for you after end-of-life (EOL) date. To upgrade manually, choose **Modify** for the DB instance or cluster. If you have any questions, contact [AWS Support](#)

Database deprecations (1)

You have 1 Aurora MySQL database with engine version 1 (with MySQL 5.6 compatibility) that will reach end-of-life (EOL). Starting on February 28, 2023, RDS will automatically upgrade to engine version 2 (with MySQL 5.7 compatibility) during an upcoming maintenance window. See [documentation](#)

Consider creating a Blue/Green Deployment to minimize downtime during upgrades

You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. [RDS User Guide](#) [Aurora User Guide](#)

Databases

Group resources

Modify

Actions

Restore from S3

Create database

Filter by databases

< 1 >

	DB identifier	Role	Engine	Region & AZ	Size	Status	Actions	CPU	Current activity	Mail
	udacity-db-cluster	Regional cluster	Aurora MySQL	us-east-2	2 instances	Available	4 Actions	-		non
	udacity-db-instance-1	Writer instance	Aurora MySQL	us-east-2b	db.t2.small	Available	-	15.00%	3 Selects/Sec	non
	udacity-db-instance-0	Reader instance	Aurora MySQL	us-east-2a	db.t2.small	Available	-	21.00%	2 Selects/Sec	non

## Zone2:

The screenshot shows the Amazon RDS console interface. On the left is a navigation menu with options like Dashboard, Databases, Query Editor, Performance Insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, Recommendations, and Certificate update. The main content area is titled 'RDS > Databases'. At the top, there are two informational banners: one about upgrading the database engine and another about creating a Blue/Green Deployment. Below these is a 'Databases' section with a search bar and a table of database instances. The table has columns for DB identifier, Role, Engine, Region & AZ, Size, Status, Actions, CPU, Current activity, and Maintenance. Three instances are listed: 'udacity-db-cluster-s' (Replica cluster), 'udacity-db-instance-0-s' (Writer instance), and 'udacity-db-instance-1-s' (Reader instance). All are running Aurora MySQL in the us-west-1 region.

DB identifier	Role	Engine	Region & AZ	Size	Status	Actions	CPU	Current activity	Maintenance
udacity-db-cluster-s	Replica cluster	Aurora MySQL	us-west-1	2 instances	Available	4 Actions	-	-	non-
udacity-db-instance-0-s	Writer instance	Aurora MySQL	us-west-1a	db.t2.small	Available	-	13.77%	3 Selects/Sec	non-
udacity-db-instance-1-s	Reader instance	Aurora MySQL	us-west-1c	db.t2.small	Available	-	30.66%	2 Selects/Sec	non-

## Terraform Destroy:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  CODEWHISPERER REFERENCE LOG

There is no explicit declaration for local provider name "aws" in module.vpc_west, so Terraform is assuming you mean to pass a configuration for "hashicorp/aws".

If you also control the child module, add a required_providers entry named "aws" with the source address "hashicorp/aws".

(and one more similar warning elsewhere)

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

module.vpc_west.aws_internet_gateway.this[0]: Destroying... [id=igw-01bafd637f8695599]
module.vpc_west.aws_internet_gateway.this[0]: Destruction complete after 2s
module.vpc_west.aws_vpc.this: Destroying... [id=vpc-0ccdcbbcedeb629312]
module.vpc_west.aws_vpc.this: Still destroying... [id=vpc-0ccdcbbcedeb629312, 10s elapsed]
module.vpc_west.aws_vpc.this: Still destroying... [id=vpc-0ccdcbbcedeb629312, 20s elapsed]
module.vpc_west.aws_vpc.this: Still destroying... [id=vpc-0ccdcbbcedeb629312, 30s elapsed]
module.vpc_west.aws_vpc.this: Still destroying... [id=vpc-0ccdcbbcedeb629312, 40s elapsed]
module.vpc_west.aws_vpc.this: Still destroying... [id=vpc-0ccdcbbcedeb629312, 50s elapsed]
module.vpc_west.aws_vpc.this: Still destroying... [id=vpc-0ccdcbbcedeb629312, 1m0s elapsed]
module.vpc_west.aws_vpc.this: Still destroying... [id=vpc-0ccdcbbcedeb629312, 1m10s elapsed]
module.vpc_west.aws_vpc.this: Still destroying... [id=vpc-0ccdcbbcedeb629312, 1m20s elapsed]
module.vpc_west.aws_vpc.this: Still destroying... [id=vpc-0ccdcbbcedeb629312, 1m30s elapsed]
module.vpc_west.aws_vpc.this: Destruction complete after 1m38s

Destroy complete! Resources: 2 destroyed.

Vinoth@Vinoth MINGW64 ~/Desktop/Udacity/08-SRE/02-HA_DR/Planning-for-High-Availability-and-Incident-Response/project/starter-code/zone1 (master)
$
```