

Step 1: IAM User Setup

First, I created an IAM user called project-admin with programmatic access. This gave me an access key that I would use to interact with AWS services via CLI.

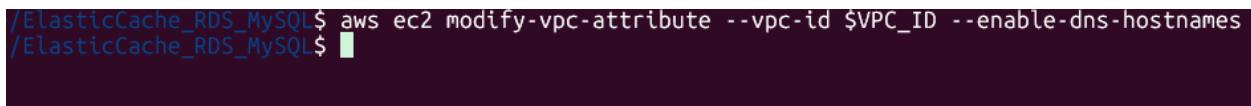


Step 2: VPC Infrastructure Setup

I started by creating the foundational network infrastructure using the AWS CLI.



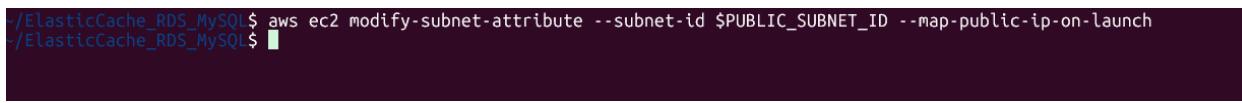
I then enabled DNS hostnames on the VPC:



Step 3: Creating Subnets



I configured it to auto-assign public IPs on launch.



I created two private subnets for high availability:



```
~/ElasticCache_RDS_MySQL$ PRIVATE_SUBNET_2=$(aws ec2 create-subnet \
--vpc-id $VPC_ID \
--cidr-block 10.0.3.0/24 \
--availability-zone us-east-1b \
--tag-specifications 'ResourceType=subnet,Tags=[{"Key=Name,Value=rds-private-subnet-2"}]')
```

These would host my RDS and ElasticCache instances.

Step 4: Internet Gateway and NAT Gateway

Internet Gateway:

I created and attached an Internet Gateway to my VPC to allow internet access for the public subnet.

```
~/ElasticCache_RDS_MySQL$ IGW_ID=$(aws ec2 create-internet-gateway \
--tag-specifications 'ResourceType=internet-gateway,Tags=[{"Key=Name,Value=rds-internet-gateway}]"' \
--query 'InternetGateway.InternetGatewayId' \
--output text)
```

```
~/ElasticCache_RDS_MySQL$ aws ec2 attach-internet-gateway --internet-gateway-id $IGW_ID --vpc-id $VPC_ID
~/ElasticCache_RDS_MySQL$
```



Elastic IP and NAT Gateway:

I allocated an Elastic IP and created a NAT Gateway in the public subnet. This allows resources in my private subnets to access the internet for updates while remaining private.



Step 5: Route Tables Configuration:

Public Route Table

I created a public route table and added a route to the Internet Gateway. I associated it with my public subnet.

```
~/ElasticCache_RDS_MySQL$ PUBLIC_RT=$(aws ec2 create-route-table \
--vpc-id $VPC_ID \
--tag-specifications 'ResourceType=route-table,Tags=[{"Key=Name,Value=rds-public-route-table"}]' \
--query 'RouteTable.RouteTableId' \
```

```
~/ElasticCache_RDS_MySQL$ aws ec2 associate-route-table --route-table-id $PUBLIC_RT --subnet-id $PUBLIC_SUBNET_ID
{
    "AssociationId": "[REDACTED]",
    "AssociationState": "associated",
    "State": "associated"
}
```

Private Route Table:

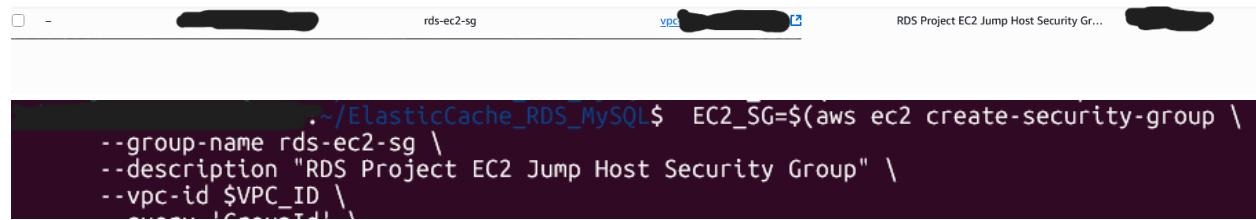
I created a private route table and added a route to the NAT Gateway. I associated it with both private subnets. At this point my VPC infrastructure was complete.

```
~/ElasticCache_RDS_MySQL$ EIP_ALLOC=$(aws ec2 allocate-address --domain vpc --query 'AllocationId' --output text)
```

Step 6: Security Groups

I created three security groups:

EC2 Jump Host Security Group:



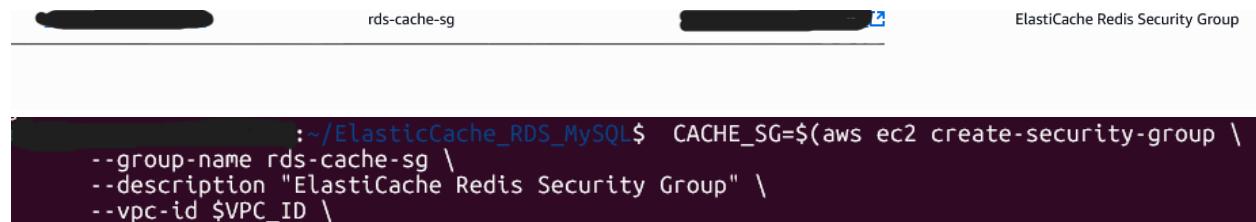
```
~/ElasticCache_RDS_MySQL$ EC2_SG=$(aws ec2 create-security-group \
--group-name rds-ec2-sg \
--description "RDS Project EC2 Jump Host Security Group" \
--vpc-id $VPC_ID \
--query 'GroupId')
```

RDS MySQL Security Group



```
~/ElasticCache_RDS_MySQL$ RDS_SG=$(aws ec2 create-security-group \
--group-name rds-mysql-sg \
--description "RDS MySQL Security Group" \
--vpc-id $VPC_ID \
--query 'GroupId')
```

ElasticCache Redis Security Group



```
~/ElasticCache_RDS_MySQL$ CACHE_SG=$(aws ec2 create-security-group \
--group-name rds-cache-sg \
--description "ElasticCache Redis Security Group" \
--vpc-id $VPC_ID \
--query 'GroupId')
```

This setup ensures that my databases are only accessible from my jump host, not directly from the internet.

Step 7: S3 Bucket and Data Upload

I created an S3 bucket to store my CSV data files:

```
~/ElasticCache_RDS_MySQL$ BUCKET_NAME="rds-mysql-data-$(date +%s)"  
~/ElasticCache_RDS_MySQL$ aws s3 mb s3://$BUCKET_NAME --  
make_bucket: rds-mysql-data-1763680631  
~/ElasticCache_RDS_MySQL$
```

 [rds-mysql-data-1763680631](#)  November 20, 2025, 17:17:25 (UTC-06:00)

I uploaded three CSV files containing my project data:

<input type="checkbox"/>	Name	Type	Last modified	Size
<input type="checkbox"/>	 data_science_team.csv	csv	November 20, 2025, 17:18:40 (UTC-06:00)	
<input type="checkbox"/>	 emp_record_table.csv	csv	November 20, 2025, 17:18:53 (UTC-06:00)	
<input type="checkbox"/>	 proj_table.csv	csv	November 20, 2025, 17:19:04 (UTC-06:00)	

Step 8: IAM Role for EC2

I created an IAM role that allows EC2 instances to assume it and access S3.

```
 rds-ec2-s3-access-role AWS Service: ec2  
  
~/ElasticCache_RDS_MySQL$ aws iam create-role \  
--role-name rds-ec2-s3-access-role \  
--assume-role-policy-document '{"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Principal": {"Service": "ec2.amazonaws.com"}, "Action": "sts:AssumeRole"}]}'  
"Role": {  
    "Path": "/",  
    "RoleName": "rds-ec2-s3-access-role",  
    "Arn": "arn:aws:iam::123456789012:role/rds-ec2-s3-access-role"  
}  
BUCKET_NAME=rds-mysql-data-1763680631  
~/ElasticCache_RDS_MySQL$ cat > ec2-trust-policy.json  
{  
    "Version": "2012-10-17",
```

Step 9: RDS MySQL Database:

DB Subnet Group & RDS Instance:

```
~/ElasticCache_RDS_MySQL$ aws rds create-db-subnet-group \  
--db-subnet-group-name rds-db-subnet-group \  
--db-subnet-group-description "RDS DB subnet group for project" \  
--db-subnet-associations Associate Subnet 1 Associate Subnet 2
```

 [rds-mysql](#)  Creating Instance MySQL Co... db.t3.micro

Step 10: ElasticCache Redis Cluster

I launched an ElasticCache Redis cluster:

```
./ElasticCache_RDS_SQL$ aws elasticache create-cache-subnet-group \
--cache-subnet-group-name rds-cache-subnet-group \
--cache-subnet-group-description "RDS Cache subnet group" \
--subnet-ids $PRIVATE_SUBNET_1 $PRIVATE_SUBNET_2
```

Step 11: EC2 Jump Host Instance

I created a key pair and downloaded the PEM file, setting permissions to 400.

```
./ElasticCache_RDS_SQL$ ssh -i rds-project-key.pem ec2-user@$PUBLIC_IP
+ '54.224.27.170' can't be established
```

Launched EC2 instance



Step 12: Connecting and Installing Tools

I SSH'd into my EC2 instance and installed the required tools

```
[ec2-user@ip-172-31-10-10 ~]$ sudo yum install mysql -y
```

```
[ec2-user@ip-172-31-10-10 ~]$ sudo amazon-linux-extras install redis6 -y
```

Step 13: Downloading Data from S3

From the EC2 instance, I downloaded my CSV files from S3:

```
aws s3 cp s3://$BUCKET_NAME/data_science_team.csv .
```

```
aws s3 cp s3://$BUCKET_NAME/emp_record_table.csv .
```

```
aws s3 cp s3://$BUCKET_NAME/proj_table.csv .
```

```
ls -lh *.csv
-r 1.1K Nov 20 23:18 data_science_team.csv
-r 1.7K Nov 20 23:18 emp_record_table.csv
-r 472 Nov 20 23:19 proj_table.csv
```

Step 14: Connecting to RDS and Creating Tables

I connected to my RDS MySQL instance:

Command: mysql -h <endpoint> -u admin -p scienceqtech_db

```
MySQL [scienceqtech_db]> █
```

I created the proj_table, emp_record_table, and data_science_team table with columns:

```
MySQL [scienceqtech_db]> CREATE TABLE proj_table (
->     PROJECT_ID VARCHAR(10) PRIMARY KEY,
->     PROJ_Name VARCHAR(100),
->     DOMAIN VARCHAR(50),
->     START_DATE DATE,
->     CLOSURE_DATE DATE,
->     DEV_QTR VARCHAR(10),
->     STATUS VARCHAR(50)
-> );
```

```
MySQL [scienceqtech_db]> SHOW TABLES;
+-----+
| Tables_in_scienceqtech_db |
+-----+
| data_science_team          |
| emp_record_table           |
| proj_table                 |
+-----+
3 rows in set (0.01 sec)
```

Step 15: Loading Data into MySQL

I loaded data from my CSV files into the MySQL tables:

```
mysql -h <RDS_ENDPOINT> -u admin -p<PASSWORD> scienceqtech_db --local-infile=1 \
-e "LOAD DATA LOCAL INFILE './proj_table.csv' INTO TABLE proj_table
FIELDS TERMINATED BY ',' ENCLOSED BY '\"' LINES TERMINATED BY '\n' IGNORE 1
ROWS;"
```

Proj_table: 6 rows

Emp_record_table: 18 rows

Data_science_tema: 13 rows

```

+-----+
|      6 |
+-----+
[ec2-user@ip
+-----+
| COUNT(*) |
+-----+
|      18 |
+-----+
[ec2-user@ip
+-----+
| COUNT(*) |
+-----+
|      13 |
+-----+
[ec2-user@ip

```

```

MySQL [scienceqtech_db]> SELECT * FROM proj_table LIMIT 10;
+-----+-----+-----+-----+-----+-----+
| PROJECT_ID | PROJ_Name          | DOMAIN    | START_DATE | CLOSURE_DATE | DEV_QTR | STATUS   |
+-----+-----+-----+-----+-----+-----+
|       3     | Drug Discovery      | HEALTHCARE| 0000-00-00 | 0000-00-00 | Q1      | DONE     |
|       5     | Fraud Detection     | FINANCE   | 0000-00-00 | 0000-00-00 | Q1      | DONE     |
| P109      | Market Basket Analysis | RETAIL   | 0000-00-00 | 0000-00-00 | Q1      | DELAYED  |
|           | Supply Chain Management | AUTOMOTIVE| 0000-00-00 | 0000-00-00 | Q2      | WIP      |
|           | Early Detection of Lung Cancer | HEALTHCARE| 0000-00-00 | 0000-00-00 | Q3      | YTS      |
|           | Customer Sentiment Analysis | RETAIL   | 0000-00-00 | 0000-00-00 | Q2      | WIP      |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)

MySQL [scienceqtech_db]> ■

```

Step 16: Testing ElasticCache Redis

I connected to my Redis cluster and tested it:

Command: redis-cli -h <rds endpoint> ping

```

[ec2-user@ip ~]$ redis-cli -h [REDACTED] ping
PONG
[ec2-user@ip ~]$ redis-cli -h [REDACTED]
> SET test "Hello Redis"

```

Step 17: Python Caching Demo

Finally, I installed Python libraries and ran a caching demo:

Pip3 install pymysql redis –user

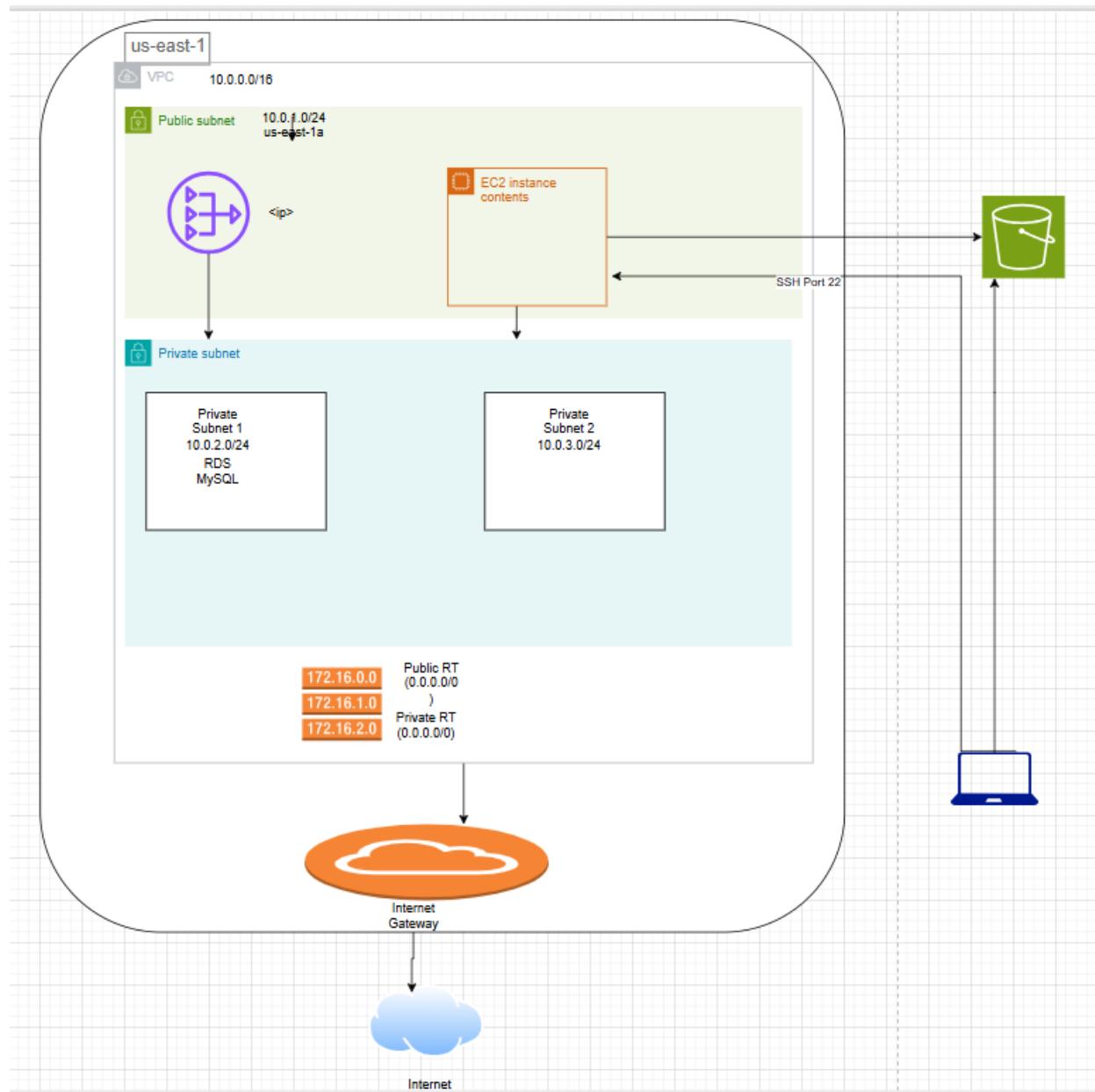
Python3 test_cache.py

Output:

```
[ec2-user@ip-10-0-1-158 ~]$ python3 test_cache.py
1st execution (should be MISS):
Cache MISS - Querying MySQL database...
Database query took: 0.0014 seconds
Cached for 600 seconds
Returned 10 rows
2nd execution (should be HIT):
Cache HIT! Retrieved from Redis
Returned 10 rows
Caching Demo Complete!
```

Using ElastiCache Redis dramatically speeds up repeated queries by caching results from RDS MySQL!

Architecture:



References:

Amazon VPC User Guide

<https://docs.aws.amazon.com/vpc/latest/userguide/>

VPC with Public and Private Subnets

https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Scenario2.html

NAT Gateways

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html>

Internet Gateways

https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html

Route Tables

https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Route_Tables.html

Security Groups

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-security-groups.html>

Amazon RDS

Amazon RDS User Guide

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/>

RDS MySQL Documentation

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_SQL.html

Creating a MySQL DB Instance

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_GettingStarted.CreatingConnecting.MySQL.html

DB Subnet Groups

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_VPC.WorkingWithRDSDInstanceinaVPC.html

RDS Security Best Practices

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_BestPractices.Security.html

Amazon ElastiCache

ElastiCache User Guide

<https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/>

ElastiCache for Redis

<https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/WhatIs.html>

Creating a Redis Cluster

<https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/GettingStarted.CreateCluster.html>

ElastiCache Subnet Groups

<https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/SubnetGroups.html>

Caching Strategies

<https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Strategies.html>

Amazon EC2

Amazon EC2 User Guide

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/>

Launching an EC2 Instance

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html

EC2 Key Pairs

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html>

Instance Profiles

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html>

Amazon S3

Amazon S3 User Guide

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/>

Creating a Bucket

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/create-bucket-overview.html>

Uploading Objects

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/upload-objects.html>

AWS IAM

IAM User Guide

<https://docs.aws.amazon.com/IAM/latest/UserGuide/>

Creating IAM Users

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html

IAM Roles for EC2

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html>

IAM Instance Profiles

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-ec2_instance-profiles.html

AWS CLI

AWS CLI User Guide

<https://docs.aws.amazon.com/cli/latest/userguide/>

AWS CLI Command Reference

<https://docs.aws.amazon.com/cli/latest/reference/>

EC2 CLI Reference

<https://docs.aws.amazon.com/cli/latest/reference/ec2/>

RDS CLI Reference

<https://docs.aws.amazon.com/cli/latest/reference/rds/>

ElastiCache CLI Reference

<https://docs.aws.amazon.com/cli/latest/reference/elasticache/>

MySQL Documentation

MySQL 8.0 Reference Manual

<https://dev.mysql.com/doc/refman/8.0/en/>

LOAD DATA Statement

<https://dev.mysql.com/doc/refman/8.0/en/load-data.html>

CREATE TABLE Statement

<https://dev.mysql.com/doc/refman/8.0/en/create-table.html>

MySQL Command-Line Client

<https://dev.mysql.com/doc/refman/8.0/en/mysql.html>

Redis Documentation

Redis Documentation

<https://redis.io/docs/>

Redis Commands

<https://redis.io/commands/>

Redis CLI

<https://redis.io/docs/ui/cli/>

Redis GET Command

<https://redis.io/commands/get/>

Redis SET Command

<https://redis.io/commands/set/>

Python Libraries

PyMySQL Documentation

<https://pymysql.readthedocs.io/en/latest/>

redis-py Documentation

<https://redis-py.readthedocs.io/en/stable/>

PyPI - PyMySQL

<https://pypi.org/project/PyMySQL/>

PyPI - redis

<https://pypi.org/project/redis/>

AWS Architecture & Best Practices

AWS Well-Architected Framework

<https://docs.aws.amazon.com/wellarchitected/latest/framework/>

Database Caching Strategies Using Redis

<https://aws.amazon.com/getting-started/hands-on/boosting-mysql-database-performance-with-amazon-elasticache-for-redis/>

VPC Security Best Practices

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-security-best-practices.html>

AWS Architecture Center

<https://aws.amazon.com/architecture/>

Tutorials & Hands-On Labs

Getting Started with Amazon RDS

<https://aws.amazon.com/getting-started/hands-on/create-mysql-db/>

Getting Started with ElastiCache

<https://aws.amazon.com/getting-started/hands-on/boosting-mysql-database-performance-with-amazon-elasticache-for-redis/>

VPC Tutorial

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-getting-started.html>

AWS Pricing

RDS Pricing

<https://aws.amazon.com/rds/mysql/pricing/>

ElastiCache Pricing

<https://aws.amazon.com/elasticache/pricing/>

EC2 Pricing

<https://aws.amazon.com/ec2/pricing/>

NAT Gateway Pricing

<https://aws.amazon.com/vpc/pricing/>

S3 Pricing

<https://aws.amazon.com/s3/pricing/>