**Cloud Formation Project**

Ample Technologies is a golden partner of AWS and utilizes most of the AWS services. Now, CTO is planning to set up database infrastructure on the AWS cloud and expects that infrastructure would expand in the future. There can be multiple changes to infrastructure depending on the future requirements. Engineers should keep track of future changes and easily roll back to the previous version if new infrastructure changes fail.

Ample uses MySQL as a database engine and would like to create a database in the private subnets. You may use the below configurations to set up the complete infrastructure.

**Database Configuration**

|  |  |
| --- | --- |
| Storage Space | 20 GB |
| Database Instance Class | Db.t2.micro |
| Database name | Ampledb |
| Database Engine | MySQL |
| Engine Version | 8.0.42 |
| PubliclyAccessible | False |
| Username | awsuser |
| Password | Aws123456789 |
| Port | 3306 |
| AllowUpgradeVersion | False |
| BackUp Retention Period | 7 Days |

VPC Configuration

|  |  |
| --- | --- |
| CIDR Block | 192.168.0.0/16 |
| EnableDNSHostnames | False |
| EnableDNSSupport | False |
| No. of Private Subnets | 2 |
| Private Subnet 1 CIDR | 192.168.1.0/24 |
| Private Subnet 1 AZ | us-east-1a |
| Private Subnet 2 CIDR | 192.168.2.0/24 |
| Private Subnet 2 AZ | us-east-1b |

Security Group Configuration

|  |  |
| --- | --- |
| SG Ingress CIDR | 0.0.0.0/0 |
| SG Ingress FromPort | 3306 |
| SG Ingress ToPort | 3306 |
| SG Ingress IPProtocol | TCP |
| SG Egress CIDR | 0.0.0.0/0 |
| SG Egress FromPort | 80 |
| SG Egress ToPort | 80 |
| SG Egress IPProtocol | TCP |

To-Dos:

* List down AWS services to be used?
* Suggest your approach.
* Design your plan
* Implement your plan
* Verify if the complete infrastructure is running

Deliverables:

* Setup Complete Infrastructure

**What is Cloud Formation?**

**A computer screen with many different colored boxes

AI-generated content may be incorrect.**

**How cloud Formation works**

A diagram of a diagram

AI-generated content may be incorrect.

A diagram of a computer

AI-generated content may be incorrect.

**Cloud Formation Rollbacks**

A person standing in front of a computer screen

AI-generated content may be incorrect.

**Step 1: AWS Service**

* 1. Cloud Formation
  2. RDS Database
  3. Security Group
  4. 2 Private subnets
  5. VPC
  6. Subnet Group

**The Design**

**A screenshot of a computer

AI-generated content may be incorrect.**

The services would be created in the following order:

1. VPC
2. Private subnet 1
3. Private subnet 2
4. Security Group
5. Subnet Group
6. RDS DB

**Why Cloud formation instead of manually managing the services individually?**

Cloud Formation help you to keep track of your Aws services in one dashboard.

**Note**: We have no need for I AM role because most of the services are networking services. So, by default, they communicate with each other.

Copy the code below and load it on your VSCode editor. But ensure it is properly formatted in the editor, least it would throw errors. To do this, click the VSCode extensions and enter AWS on the extensions search bar

A screenshot of a computer

AI-generated content may be incorrect.

Enter AWS in the extensions search bar.

A screenshot of a computer

AI-generated content may be incorrect.

Click on the AWS Cloud formation snippet and install

A screenshot of a computer

AI-generated content may be incorrect.

Copy the code below and paste in the VSCode editor

AWSTemplateFormatVersion: 2010-09-09

Description: Live project cloud formation

Resources:

  rdsVPC:

    Type: AWS::EC2::VPC

    Properties:

      CidrBlock: "192.168.0.0/16"

      EnableDnsHostnames: false

      EnableDnsSupport: false

      Tags:

        - Key: "Name"

          Value: "rdsVPC"

  PrivateSubnet1:

    Type: AWS::EC2::Subnet

    Properties:

      AvailabilityZone: "us-east-1a"

      CidrBlock: "192.168.1.0/24"

      Tags:

        - Key: "Name"

          Value: "PrivateSubnet1"

      VpcId:

        Ref: rdsVPC  # Required

  PrivateSubnet2:

    Type: AWS::EC2::Subnet

    Properties:

      AvailabilityZone: "us-east-1b"

      CidrBlock: "192.168.2.0/24"

      Tags:

        - Key: "Name"

          Value: "PrivateSubnet2"

      VpcId:

        Ref: rdsVPC  # Required

  rdsSG:

    Type: AWS::EC2::SecurityGroup

    Properties:

      GroupDescription: "rdsSG" # Required

      GroupName: "rdsSG"

      SecurityGroupEgress:

        - CidrIp: "0.0.0.0/0"

          Description: "Egress rules"

          FromPort: "80"

          IpProtocol: "tcp"

          ToPort: "80"

      SecurityGroupIngress:

        - CidrIp: "0.0.0.0/0"

          Description: "Ingress rules"

          FromPort: "3306"

          IpProtocol: "tcp"

          ToPort: "3306"

      Tags:

        - Key: "Name"

          Value: "rdsSG"

      VpcId:

        Ref: rdsVPC  # Required

  rdsSubnetGroup:

    Type: AWS::RDS::DBSubnetGroup

    Properties:

      DBSubnetGroupDescription: "Group of subnets" # Required

      DBSubnetGroupName: "rdsSubnetGroup"

      SubnetIds: # Required

        - Ref: PrivateSubnet1  # Required

        - Ref: PrivateSubnet2

      Tags:

        - Key: "Name"

          Value: "rdsSubnetGroup"

  rdsDB:

    Type: AWS::RDS::DBInstance

    Properties:

      AllocatedStorage: "20"

      AllowMajorVersionUpgrade: false

      AutoMinorVersionUpgrade: false

      BackupRetentionPeriod: "7"

      DBInstanceClass: "db.t3.micro"

      DBInstanceIdentifier: "myRDSInstance"

      DBName: "Ampledb"

      DBSubnetGroupName:

        Ref: rdsSubnetGroup  # Required

      Engine: "MySQL"

      EngineVersion: "8.0.42"

      MasterUserPassword: "Aws123456789"

      MasterUsername: "awsuser"

      Port: "3306"

      PubliclyAccessible: false

      Tags:

        - Key: "Name"

          Value: "rdsDB"

      VPCSecurityGroups:

        - Ref: rdsSG

The VPC, Private Subnet1, Private Subnet2

A screenshot of a computer screen

AI-generated content may be incorrect.

The Security Group

A screenshot of a computer program

AI-generated content may be incorrect.

The RDS

A screenshot of a computer program

AI-generated content may be incorrect.

Whenever you are finished writing the code for the template, copy it and go to google and enter **lint.yaml**

This site would actually help to validate the code

Click on the first link

A screenshot of a computer

AI-generated content may be incorrect.

Paste your code inside the validator.

Click on the Go button

A screenshot of a computer

AI-generated content may be incorrect.

Click on the Go

A screenshot of a computer program

AI-generated content may be incorrect.

Next is to go to Cloud Formation in AWS console.

Click on Create stack

A screenshot of a computer

AI-generated content may be incorrect.

Choose an existing template option.

Template source: Upload a template file and click on choose file

A screenshot of a computer

AI-generated content may be incorrect.

My file: new.yml and click open

A screenshot of a computer

AI-generated content may be incorrect.

Click on next

A screenshot of a computer

AI-generated content may be incorrect.

Give it a name

A screenshot of a computer

AI-generated content may be incorrect.

Click on next

A screenshot of a computer

AI-generated content may be incorrect.

Make your region is North Virginia Region and Click on Submit to create the stack

A screenshot of a computer

AI-generated content may be incorrect.

Resources creation in progress

A screenshot of a computer

AI-generated content may be incorrect.

Now click on the event tab

You can see the resources being created on the event tab

A screenshot of a computer

AI-generated content may be incorrect.

**Project Complete**