**Implementation Documentation**

Chris Szabla – 9001341

INFO2350

Nikola Cedic

08/05/2025

# Implementation Guide

## Step 1: Create a Custom VPC

1. Go to VPC Dashboard → Create VPC
2. Select VPC only
3. Settings:
4. Name: MyProjectVPC
5. CIDR block: 10.0.0.0/16
6. Tenancy: Default
7. Click Create VPC

## Step 2: Create Subnets

### a. Public Subnet

* Name: PublicSubnet
* CIDR: 10.0.1.0/24
* AZ: us-east-1a
* Assign to MyProjectVPC

### b. Private Subnet

* Name: PrivateSubnet
* CIDR: 10.0.2.0/24
* AZ: us-east-1a
* Assign to MyProjectVPC

## Step 3: Create an Internet Gateway and Nat Gateway

1. Go to VPC → Internet Gateways → Create
   1. Name: MyIGW
   2. Attach to MyProjectVPC
2. Go to VPC > NAT Gateways
   1. Click Create NAT Gateway
   2. Subnet: Select the PublicSubnet
   3. Elastic IP: Allocate a new EIP

## Step 4: Configure Route Tables

### a. Public Route Table

1. Create route table → Name: PublicRouteTable
2. Associate with PublicSubnet
3. Add route:
   1. Destination: 0.0.0.0/0
   2. Target: Internet Gateway (MyIGW)

### b. Private Route Table

1. Create route table → Name: PrivateRouteTable
2. Associate with PrivateSubnet
3. Add route to private table
   1. 0.0.0.0/0
   2. Target: Select NAT Gateway

## Step 5: Launch and Configure EC2 Web Server

1. Go to EC2 → Launch Instance

* Name: WebServer
* AMI: Amazon Linux 2023
* Instance Type: t2.micro

1. Network Settings:

* VPC: MyProjectVPC
* Subnet: PublicSubnet
* Auto-assign Public IP: Enable

1. Create Security Group:
2. Name: WebSG
3. Inbound Rules:

* HTTP (80): 0.0.0.0/0
* HTTPS (443): 0.0.0.0/0
* SSH (22): Your IP only

1. Outbound: Allow All (default)
2. Allocate & associate an Elastic IP (under EC2 → Elastic IPs)
3. Install Apache + PHP + MySQL Client

* sudo dnf install -y httpd php php-mysqlnd mysql
* sudo systemctl enable --now httpd

## Step 6: Launch EC2 Database (Private Subnet)

* Instance Type: t2.micro
* OS: Amazon Linux 2023
* Subnet: PrivateSubnet
* No public IP
* Security Group (DBSG):
  1. Inbound: Allow 3306 from WebSG only
  2. Outbound: default

### Allow SSH from Web EC2 to DB EC2

* Convert PPK to PEM (if needed):
  1. Open PuTTYgen
  2. Load .ppk
  3. File → Conversions → Export OpenSSH Key → Save as mykey.pem

### Upload PEM to Web EC2:

* Manually create it with nano mykey.pem
  + Nano mykey.pem
  + Copy contents of mykey.pem

### Set Permissions on Web EC2:

* chmod 500 ~/mykey.pem

### SSH to DB from Web EC2:

* ssh -i mykey.pem [ec2-user@10.0.2.239](mailto:ec2-user@10.0.2.239)

# Install MySQL Server on DB EC2

* From Web Server, ssh into DB
  + ssh -i mykey.pem [ec2-user@10.0.2.239](mailto:ec2-user@10.0.2.239)
* sudo dnf update -y
* wget https://dev.mysql.com/get/mysql80-community-release-el9-1.noarch.rpm
* sudo dnf install -y mysql80-community-release-el9-1.noarch.rpm
* sudo dnf config-manager --enable mysql80-community
* sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022
* sudo dnf clean packages
* sudo dnf install -y mysql-server --nogpgcheck
* sudo systemctl enable --now mysqld

# Secure MySQL

## Get temp password:

* sudo grep 'temporary password' /var/log/mysqld.log

## Secure install:

* sudo mysql\_secure\_installation
  + Validate password plugin: Yes
  + Change root password: No
  + Remove anonymous users: Yes
  + Disallow root login remotely: No
  + Remove test DB: Yes
  + Reload privilege tables: Yes

## Create Database and User

* In MySQL shell:
  + CREATE DATABASE DBServer;
  + CREATE USER 'cszabla'@'10.0.1.161' IDENTIFIED BY 'Secret55!';
  + GRANT ALL PRIVILEGES ON DBServer.\* TO 'cszabla'@'10.0.1.161';
  + FLUSH PRIVILEGES;

# Connect to DB from WebServer

* From the Web Server
  + sudo dnf install -y mysql
  + mysql -h 10.0.2.239 -u cszabla -p

# Create a Test Table on DB Server

* USE DBServer;
* CREATE TABLE messages (
* id INT AUTO\_INCREMENT PRIMARY KEY,
* content VARCHAR(255) NOT NULL
* );
* INSERT INTO messages (content) VALUES ('Hello from MySQL!'), ('This is your test app!');

# Create a Test Web Page to interact with DB Server

## Install php

* sudo dnf install -y httpd php php-mysqlnd mysql
* sudo systemctl enable --now httpd

## Modify index.php on the Web Server

* <?php
* $host = 'PRIVATE\_IP\_OF\_DB';
* $db = 'DBServer';
* $user = 'cszabla';
* $pass = 'Secret55!';
* $conn = new mysqli($host, $user, $pass, $db);
* if ($conn->connect\_error) {
* die("Connection failed: " . $conn->connect\_error);
* }
* $result = $conn->query("SELECT \* FROM messages");
* while ($row = $result->fetch\_assoc()) {
* echo $row['content'] . "<br>";
* }
* $conn->close();
* ?>

# Test Web Server

In a browser, connect to <http://10.0.1.161>

# Create a CloudFormation IaC Template and Deploy Stack

* Open AWS CloudFormation Console
* Click “Create Stack” → “With new resources (standard)”
* Click the link: “IaC Generator”
  + Select your AWS resources from your environment
    - VPC
    - EC2 Instances
    - Subnets
    - Routes
    - Internet Gateway
    - Security Groups
  + Click Generate template
  + Download the resulting YAML template
    - Add logic to collect Parameters before deploying stack
    - Key pair
    - Instance type
    - AMI
    - Completely remove hardcode IDs for all resources
    - ImageID
    - KeyName
    - Remove fixed IP using !GetAtt
    - E.g value: !GetAtt DBServerInstance.PrivateIP
  + Configure Outputs to show IP’s of newly created resources using !ref and !GetAtt
* Choose an existing template
  + Select and upload the modified yaml template file
* Configure Stack Details
  + Stack name: csza1341-WebStack
  + Parameters: Default parameters left
* Stack Options
  + IAM roles: Default IAM roles used
* Click Create Stack