Experiment 2 — Launch an EC2 Instance and Access via Browser

Course: Cloud Computing Lab **Level:** Beginner **Duration:** 30–45 minutes
Date: 2025-10-08

Aim

To launch an EC2 virtual machine in an existing VPC/subnet, install a web server (Apache), deploy a test web page, and access it through a browser.

Description

In this experiment, students will create an EC2 instance (a virtual server) inside a VPC subnet, configure security group rules to allow SSH and HTTP access, install the Apache web server, and verify the site is accessible from the internet. This exercise covers compute provisioning, security groups (firewall rules), remote administration via SSH, and basic web hosting.

Prerequisites

- An AWS account with permissions to create EC2 instances and security groups.
- An existing VPC and subnet (you can use the VPC created in Experiment 1) or the default VPC.
- A key pair (.pem) to SSH into the instance (create one in EC2 if needed).
- AWS Management Console access (or AWS CLI configured).
- Basic knowledge of SSH and Linux package installation.

Procedure — Step by step (Console)

Step 1: Open the AWS Management Console → EC2 → Instances → Click *Launch instances*.

Step 2: Configure instance details: • Name: **WebServer** • AMI: Choose an Ubuntu Server (e.g., Ubuntu 22.04 LTS) or Amazon Linux. • Instance type: **t3.micro** (or t2.micro). • Key pair: Select an existing key pair or create a new one (download the .pem file). • Network: Select your VPC (e.g., **MyVPC**) and Subnet (e.g., **MySubnet**). • Auto-assign Public IP: **Enable** (so the instance gets a public IPv4 address).

Step 3: Configure Security Group: Create or select a security group (e.g., **WebSG**) with inbound rules: • SSH (TCP 22) — Source: Your lab IP (or 0.0.0.0/0 for testing, but not recommended). • HTTP (TCP 80) — Source: 0.0.0.0/0. • HTTPS (TCP 443) — optional.

Step 4: Review and Launch the instance. Wait until the instance state is *running* and system status checks pass.

Step 5: Note the instance's Public IPv4 address. Connect via SSH from your terminal: chmod 400 mykey.pem ssh -i mykey.pem ubuntu@ (*If using Amazon Linux, connect as ec2-user instead of ubuntu.*)

Step 6: On the instance, install and start Apache (Ubuntu example): sudo apt update -y sudo apt install apache2 -y sudo systemctl enable --now apache2 Replace the default page with a test page: echo "WebServer - Server-1Deployed in MyVPC" | sudo tee /var/www/html/index.html

Step 7: From your laptop browser, open http:/// and verify the test page loads.

Step 8 (Optional): To harden access, restrict SSH to lab IP ranges, configure HTTPS, and use IAM/Session Manager for access without opening port 22.

Procedure — Equivalent AWS CLI commands (optional)

```
# 1. Create security group (replace VPC_ID)
SG_ID=$(aws ec2 create-security-group --group-name WebSG --description "Web SG" --vpc-id $VPC_ID --
aws ec2 authorize-security-group-ingress --group-id $SG_ID --protocol tcp --port 22 --cidr <YOUR_IP
aws ec2 authorize-security-group-ingress --group-id $SG_ID --protocol tcp --port 80 --cidr 0.0.0.0/

# 2. Launch EC2 (replace AMI_ID and SUBNET_ID)
INSTANCE_ID=$(aws ec2 run-instances --image-id ami-<ubuntu-ami-id> --instance-type t3.micro --key-n
aws ec2 create-tags --resources $INSTANCE_ID --tags Key=Name,Value=WebServer

# 3. Get public IP
aws ec2 describe-instances --instance-ids $INSTANCE_ID --query 'Reservations[0].Instances[0].Public
```

Expected Result

After completing the procedure, students will have a running EC2 instance named 'WebServer' with Apache serving a test page. The page should be reachable at http:// (subject to security group rules). Students should be able to SSH into the instance, view and edit the web files under /var/www/html, and restart the Apache service if required.

Verification / Checklist

- EC2 instance 'WebServer' is in 'running' state and placed in the intended subnet.
- Instance has a Public IPv4 address (or an Elastic IP attached).
- Security group allows inbound HTTP (80) and SSH (22) from permitted sources.
- SSH connection to the instance succeeds and Apache responds to curl/http requests.
- Browser displays the custom test page at http:///.

Troubleshooting

- If the page doesn't load, check security group inbound rules, route table (0.0.0.0/0 → IGW), and that the subnet assigns public IPs.
- If SSH times out, verify the .pem key permissions (chmod 400), correct username (ubuntu/ec2-user), and SG inbound rule for port 22.
- Use 'sudo systemctl status apache2' and 'sudo journalctl -u apache2' to inspect service issues.

Instructor: Cloud Computing Lab • Prepared by: ChatGPT

Generated on 2025-10-08 15:27:02