DevOps Intern Exercise #1 | Deploy a React App to an AWS EC2 Instance

Written by Danzig James A. Orcales on March 23, 2025

In this exercise, you'll learn how to deploy a React application to an AWS EC2 instance. This hands-on task will guide you through setting up an AWS account, launching and configuring a virtual machine (EC2 instance), and deploying a web application. Along the way, you'll gain valuable experience with cloud infrastructure, resource management, and security configurations, building foundational skills for real-world DevOps scenarios.

Table of Contents

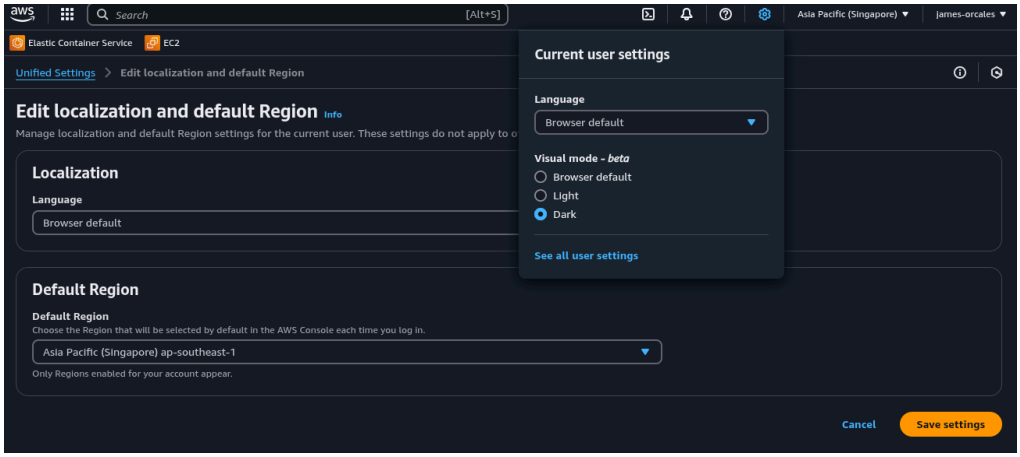
Set Up an EC2 Instance

Sign Up for an AWS Account

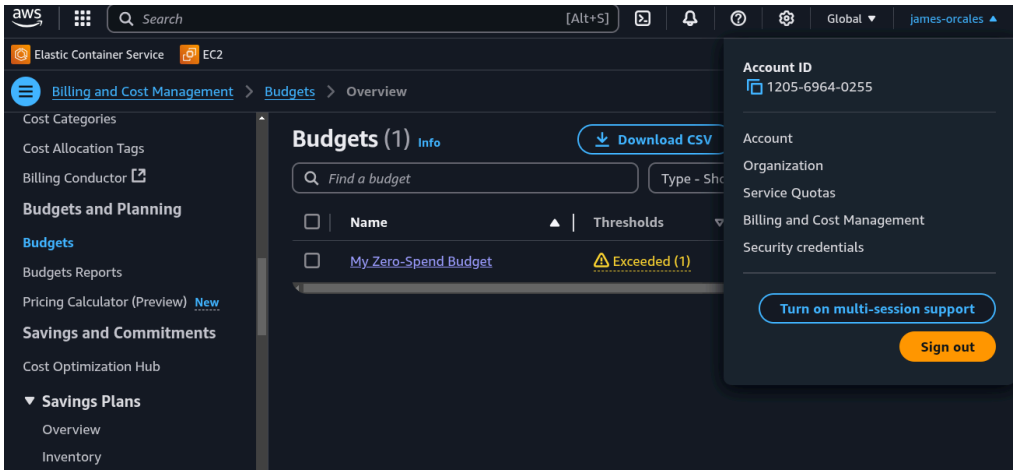
Sign up for a free-tier AWS account. An accessible credit card to use is Maya Virtual Card. AWS requires your card to *contain* at least $1 verification of billing details. You can use GCash to transfer money to your new Maya card. Use the phone number associated with your Maya account rather than the 16-digit number on your card.

Set Up your AWS Account

- Under user settings, set the account region to Singapore (ap-southeast-1).



- Under Billing and Cost Management > Budgets > Overview, create a new budget. Select the default “Zero Spend Budget” template. You will receive an email if any of your resources expenses exceed $0.01. **Note that services in AWS will not automatically shutdown when the budget is exceeded.** Though during this exercise, you may see fees incurred by the EC2 instance that we will start, this is covered by the free-tier plan. You can verify that no funds are deducted by checking your bank account balance.



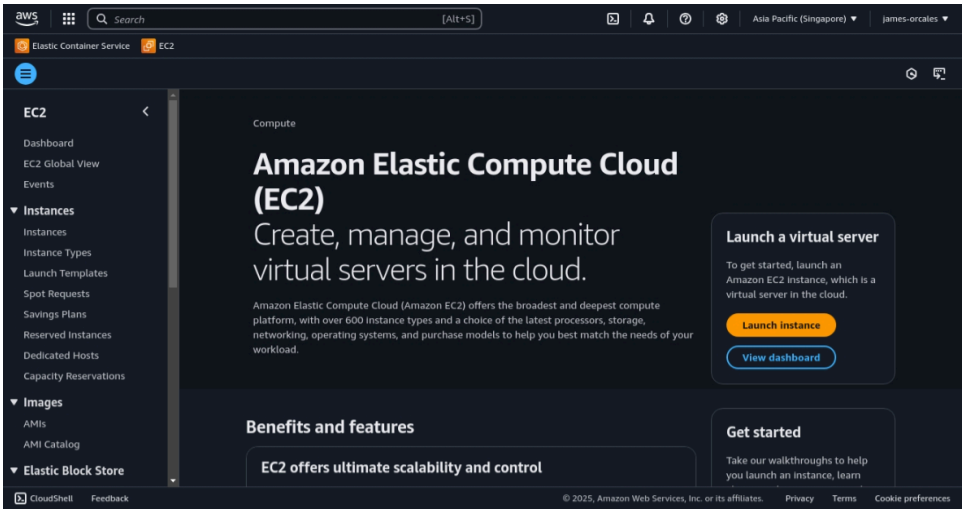
Launch an EC2 Instance

An Amazon Elastic Compute Cloud (EC2) instance is a virtual machine running on AWS infrastructure, acting like a computer you can configure and control remotely. Similar to leaving a physical laptop running continuously in your closet, an EC2 instance operates 24/7 unless stopped, consuming resources and incurring charges. However, unlike a physical device, EC2 offers flexible configuration for CPU, memory, and storage, and it's accessible globally via the internet, making it highly scalable and efficient for hosting applications or performing computations on demand.

In the AWS Console home, search for the EC2 service and launch an instance. Configure the instance as follows:

- **Amazon Machine Image**: Amazon Linux 2023 AMI 64-bit (x86)
- **Instance Type**: t2.micro

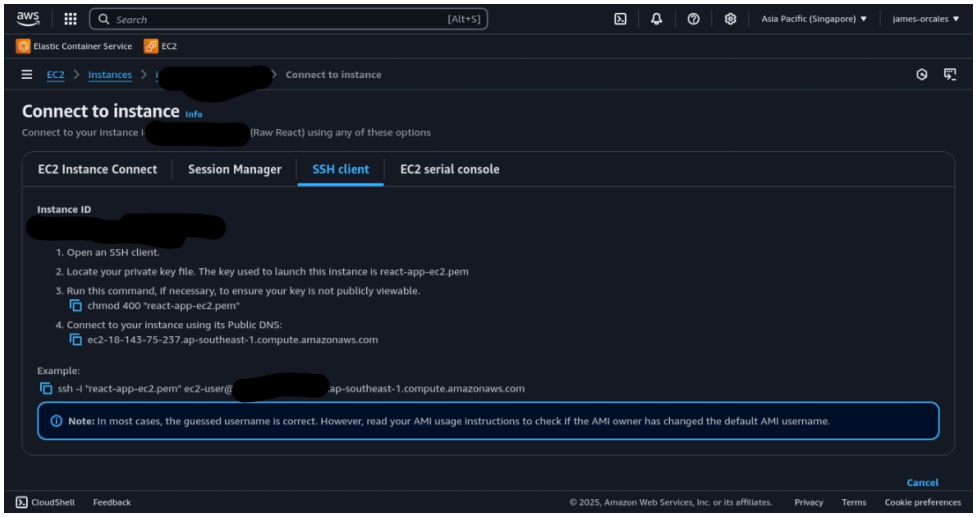
- **Create a new key pair** (if you don't have one already). This is what you use to log into the instance through SSH. Remember to save the file:
    - **Key pair type:** RSA
    - **Private key file format**: .pem
- **Create a security group.** This sets up the firewall for your instance.
    - **Allow SSH traffic from:** My IP
      Your IP would be the only one allowed to SSH into the instance.
    - **Allow HTTP traffic from the Internet**: true
      This is set to allow traffic from anywhere. We'll manually update the security group later to only allow HTTP connections from your IP.
- **Storage:** 1x8 GiB gp3



Deploy the React App

Create the project

- Under EC2 > Instances, select your newly created instance and click connect. Execute the sample command as is to SSH into your instance.



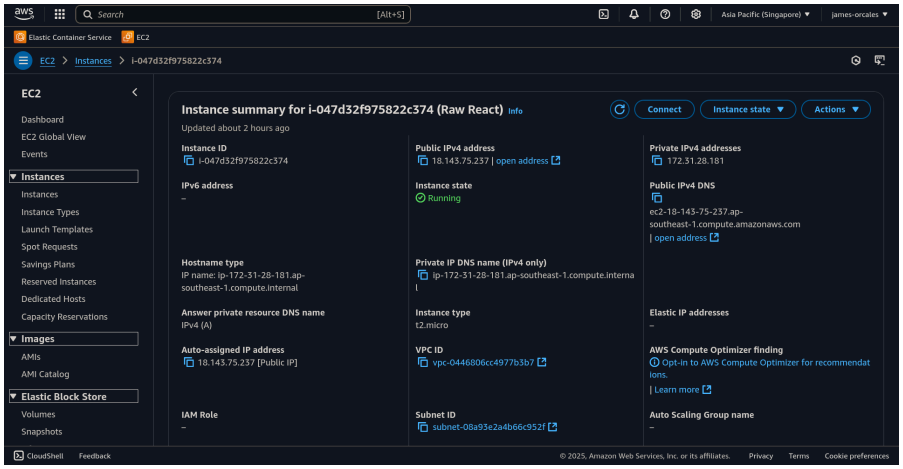
- Execute these commands.

```
sudo dnf update && \
sudo dnf install nodejs.x86_64 nodejs-npm.x86_64 -y && \
npm create vite@latest -y
# Proceed with the vite project setup. Make sure to select React.
```

Serve the React App

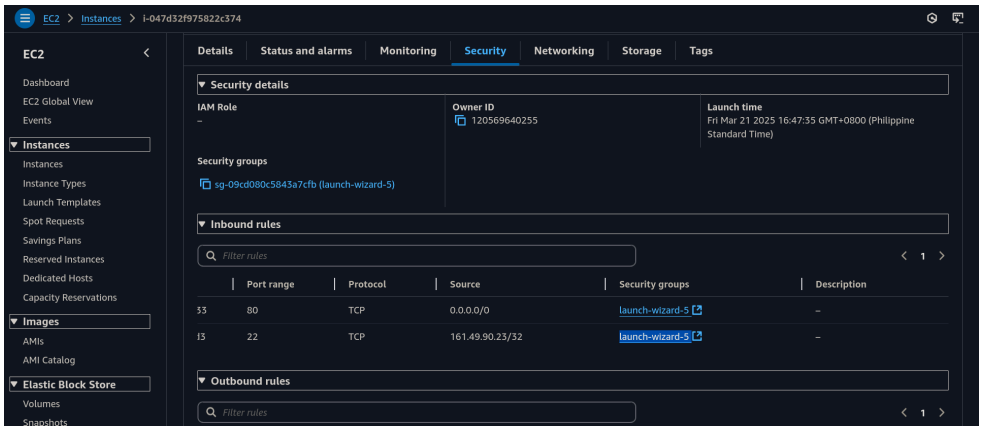
Execute the commands below. By default, vite serves on port 5173. Go to the instance summary of your EC2 instance and take note of its public IP address. Then in your web browser, enter the following url http://<PUBLIC_IP_ADDRESS>:5173

```
cd <PROJECT_NAME> && \
npm install && \
npm run dev -- --host
```



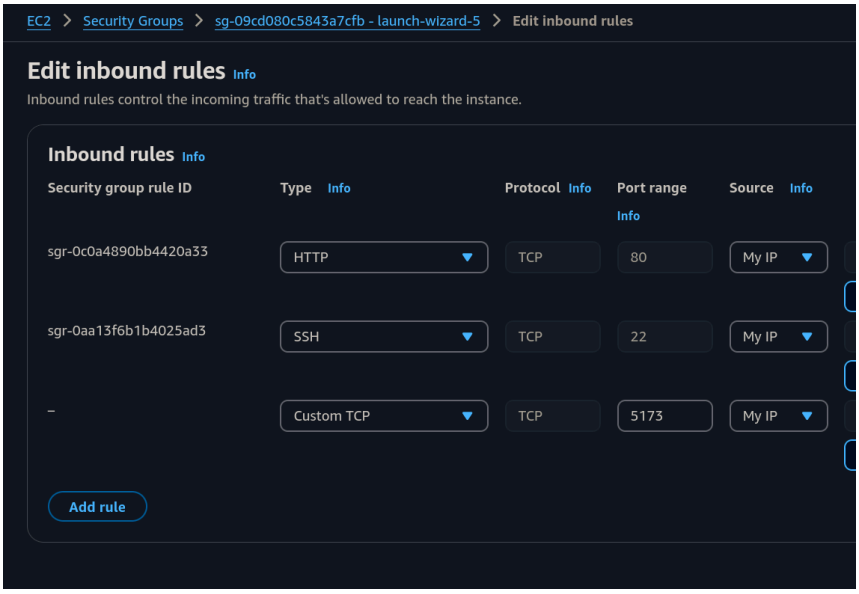
Set Up the EC2 Instance Security Group

If you tried the URL from the previous step, it's not going to resolve as we have yet to open the port of our EC2 instance. In the same instance summary dashboard, under Security > Inbound rules, click on the instance's associated security group.



Edit the inbound rules by adding a Custom TCP rule for port 5173. Set the rule's source to 'My IP' to ensure access is restricted to the device you're currently using to access the AWS console. Keep in mind that this IP address is

specific to your current network. Connecting to a different network will result in a different IP address. To find your current IP address, run `curl ifconfig.me` on your local machine. Now you can access the web app.



Conclusion

This exercise demonstrates the process of deploying a React application to an AWS EC2 instance. By following the outlined steps, you gain hands-on experience with AWS account setup, EC2 configuration, and application deployment. These skills are valuable for building scalable and accessible web applications, preparing you for real-world DevOps challenges. Successfully completing this task showcases your ability to combine technical proficiency with cloud computing to deliver effective solutions.