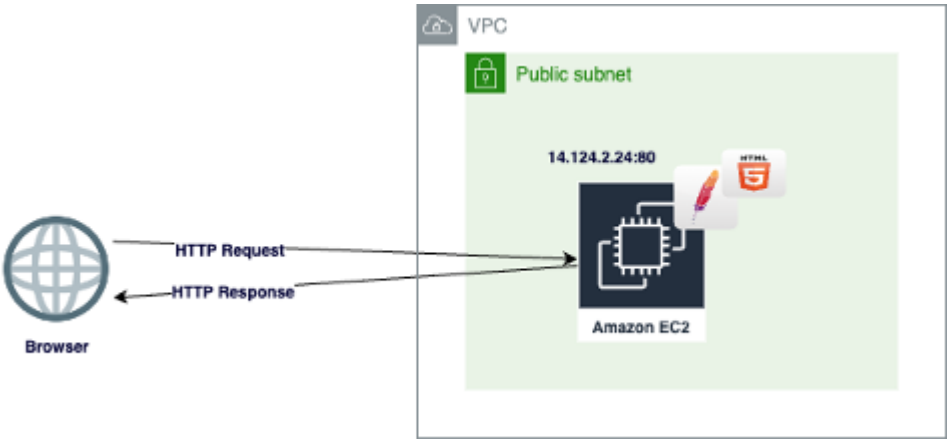
Demo – Deploy and Serve a Static Website

Nicolas El Khoury

Introduction

In this demo, we are going to deploy a simple HTML website on an AWS EC2 Ubuntu Machine. To do so, we are going to:

- Create a simple HTML page.
- Create the networking and compute resources on AWS.
- Install the Apache2 webserver.
- Deploy the HTML page.
- Configure the webserver to serve the application on port 80.



The webpage to be deployed is nothing but a simple HTML page:

```
<!DOCTYPE html>
<html>
    <head>
        <title>My First Application</title>
    </head>
    <body>
        <p>I have no idea what I'm doing.</p>
    </body>
</html>
```

Solution

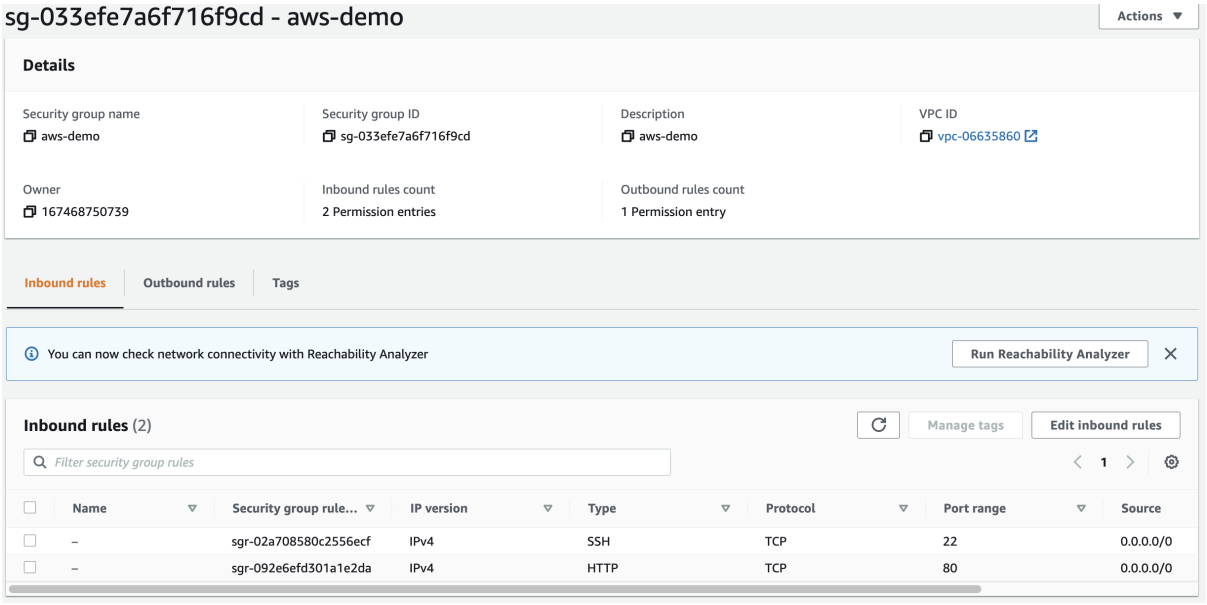
Compute and Networking Resources

SSH Keypair

1. Navigate to the **EC2** service, **Key Pairs** option from the left menu.
2. Create a Keypair.
3. The key will be automatically downloaded. Move it to a hidden directory.
4. Modify the permissions to read only: `chmod 400 <keyName>.pem`

Security Group

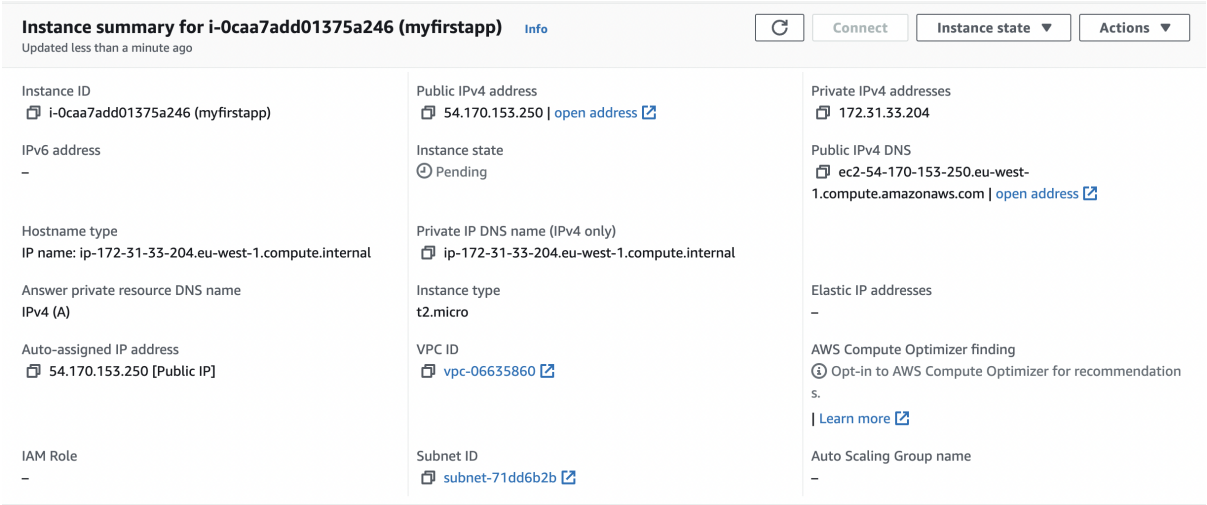
1. Navigate to the **Security group** option from the left menu.
2. Specify a name: aws-demo.
3. Attach it to the default VPC.
4. Enable ports **22** and **80** to all IPv4 addresses.

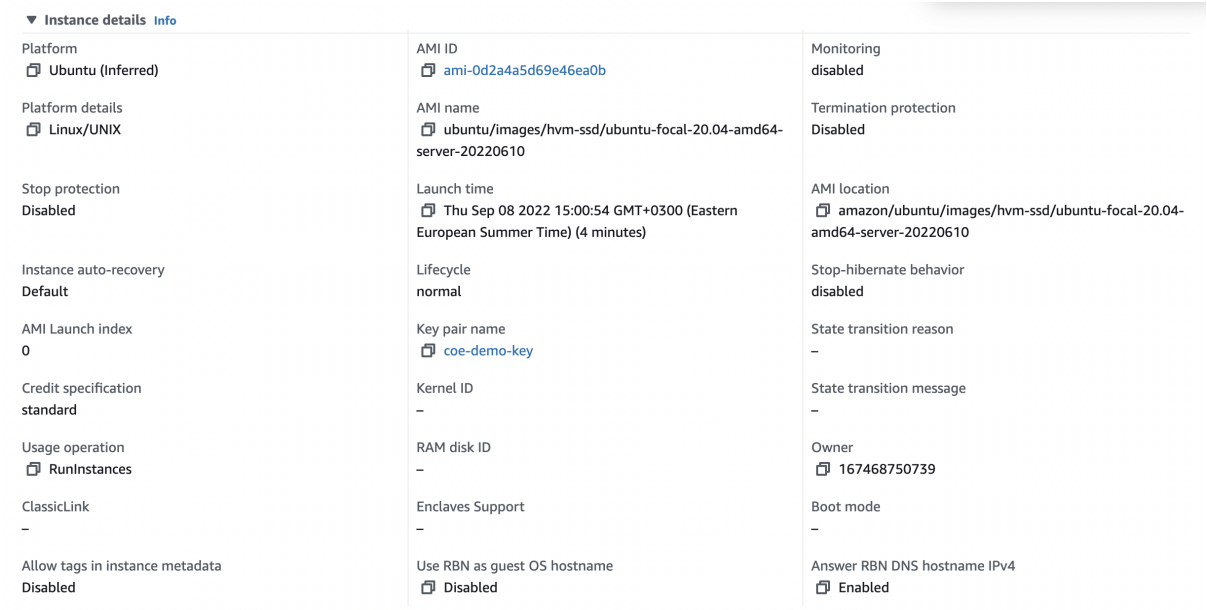


EC2 instance

Navigate to **AWS EC2** —> **instances** —> **Launch instances**, with the following parameters:

- **Name**: webserver
- **AMI**: Ubuntu Server 20.04 LTS (HVM), SSD Volume Type
- **Instance Type**: t3.medium (Or any type of your choice)
- **Key pair name**: aws-demo
- **Network Settings**:
  - **Select existing security group**: aws-demo
- **Configure storage**: 1 x 25 GiB gp2 Root volume





▼ Instance details Info

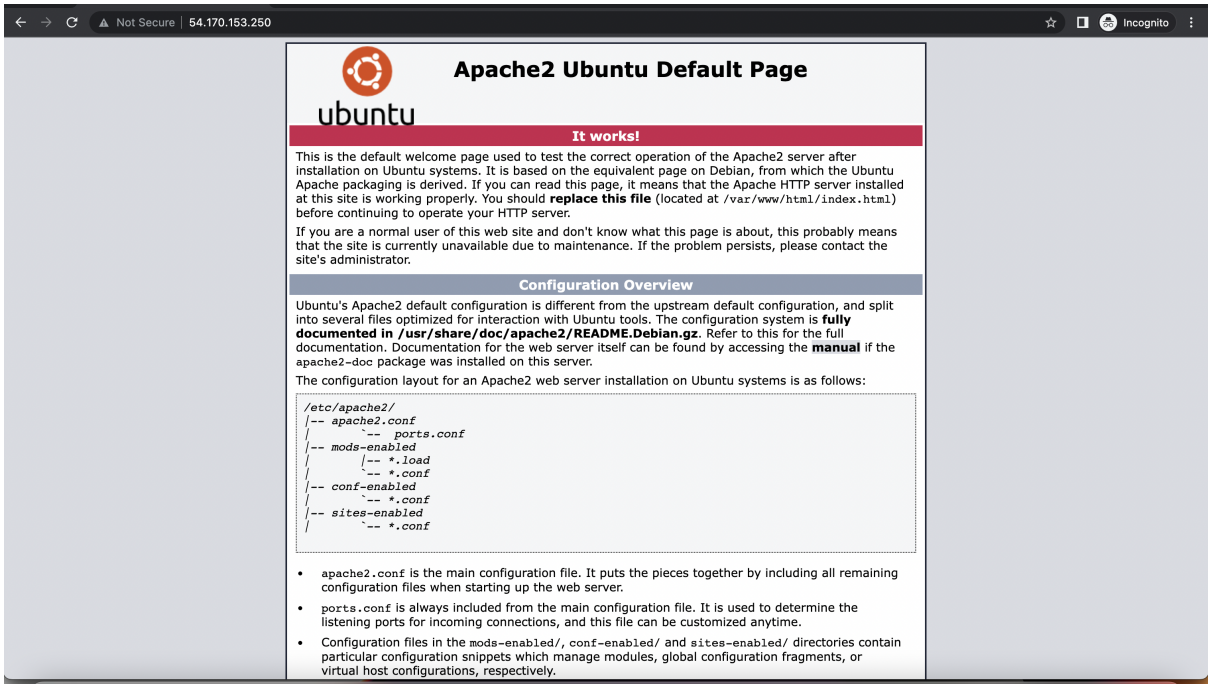
Platform Ubuntu (Inferred)	AMI ID ami-0d2a4a5d69e46ea0b	Monitoring disabled
Platform details Linux/UNIX	AMI name ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-20220610	Termination protection Disabled
Stop protection Disabled	Launch time Thu Sep 08 2022 15:00:54 GMT+0300 (Eastern European Summer Time) (4 minutes)	AMI location amazon/ubuntu/images/hvm-ssd/ubuntu-focal-20.04-amd64-server-20220610
Instance auto-recovery Default	Lifecycle normal	Stop-hibernate behavior disabled
AMI Launch index 0	Key pair name coe-demo-key	State transition reason –
Credit specification standard	Kernel ID –	State transition message –
Usage operation RunInstances	RAM disk ID –	Owner 167468750739
ClassicLink –	Enclaves Support –	Boot mode –
Allow tags in instance metadata Disabled	Use RBN as guest OS hostname Disabled	Answer RBN DNS hostname IPv4 Enabled

SSH to the machine: `ssh ubuntu@<Public IP address> -i <path to key>.pem`

```
Warning: Permanently added '54.170.153.250' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-1029-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Sep  8 12:06:30 UTC 2022

  System load:  0.0               Processes:             105
  Usage of /:   19.2% of 7.58GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for eth0: 172.31.33.204
  Swap usage:   0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-33-204:~$
```

Apache2 Installation and configuration

1. Update the local package index to reflect the latest upstream changes: `sudo apt-get update`
2. Install the **Apache2** webserver: `sudo apt-get install -y apache2`
3. Verify that the deployment worked by performing a request to the machine using its public IP. The Apache2 default page must be loaded on the browser:



Application Deployment

Perform the following steps to deploy the application:

```
# Create a directory
sudo mkdir /var/www/myfirstapp
# Change the owership to www-data
sudo chown -R www-data:www-data /var/www/myfirstapp
# Change the directory permissions
sudo chmod -R 755 /var/www/myfirstapp
# Create the index.html file and paste the HTML code into it
sudo nano /var/www/myfirstapp/index.html
# Create the log directory
sudo mkdir /var/log/myfirstapp
# Change the ownership of the directory
sudo chown -R www-data:www-data /var/log/myfirstapp/
```

Webserver Configuration

1. Create the virtual host file: `sudo nano /etc/apache2/sites-available/myfirstapp.conf`

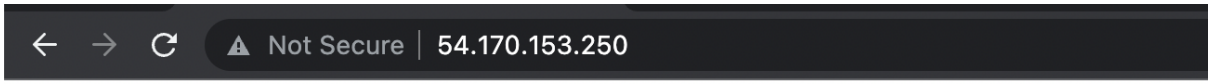
2. Paste the configuration below:

```
<VirtualHost *:80>
    DocumentRoot /var/www/myfirstapp
    ErrorLog /var/log/myfirstapp/error.log
    CustomLog /var/log/myfirstapp/requests.log combined
</VirtualHost>
```

Enable the configuration:

```
# Enable the site configuration
sudo a2ensite myfirstapp.conf
# Disable the default configuration
sudo a2dissite 000-default.conf
# Test the configuration
sudo apache2ctl configtest
# Restart apache
sudo systemctl restart apache2
```

Perform a request on the server. The response will change this time, loading the custom page that was configured.



I have no idea what I'm doing.