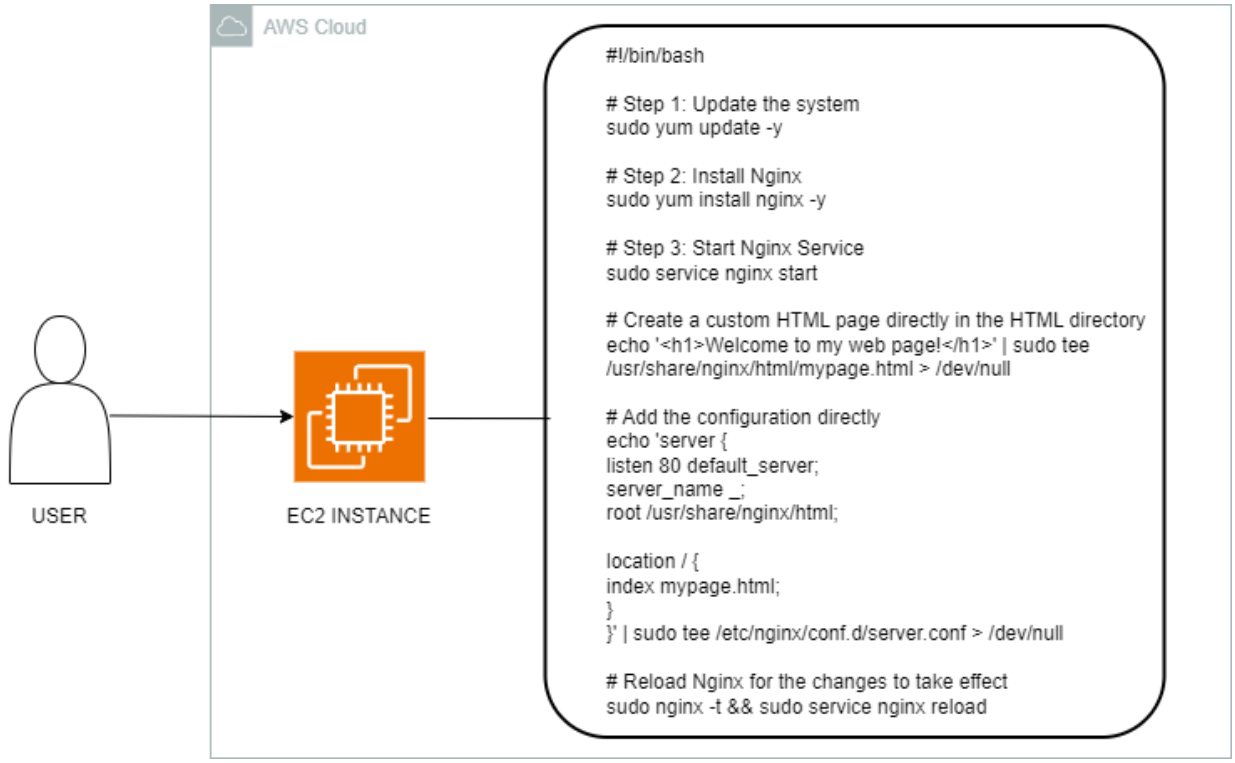
Launching an EC2 Instance with User Data

EC2 Bootstrapping involves configuring an EC2 instance to undergo automated installation and configuration procedures after launch but before it becomes operational. This is accomplished by providing a script through the User Data section of the Meta-data service, which is then executed by the operating system of the EC2 instance.



Objectives

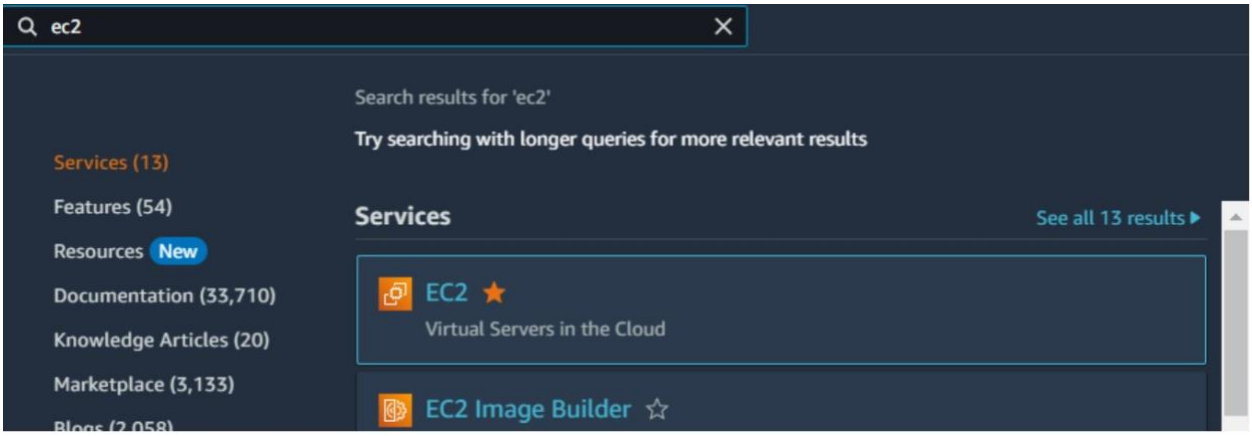
In this lab, you will:

- Gain hands-on experience with User Data, exploring its powerful capabilities in automating tasks during the launch of an instance and witnessing its practical application in configuring a simple web server.

Lab Steps

Launching an EC2 Instance with User Data

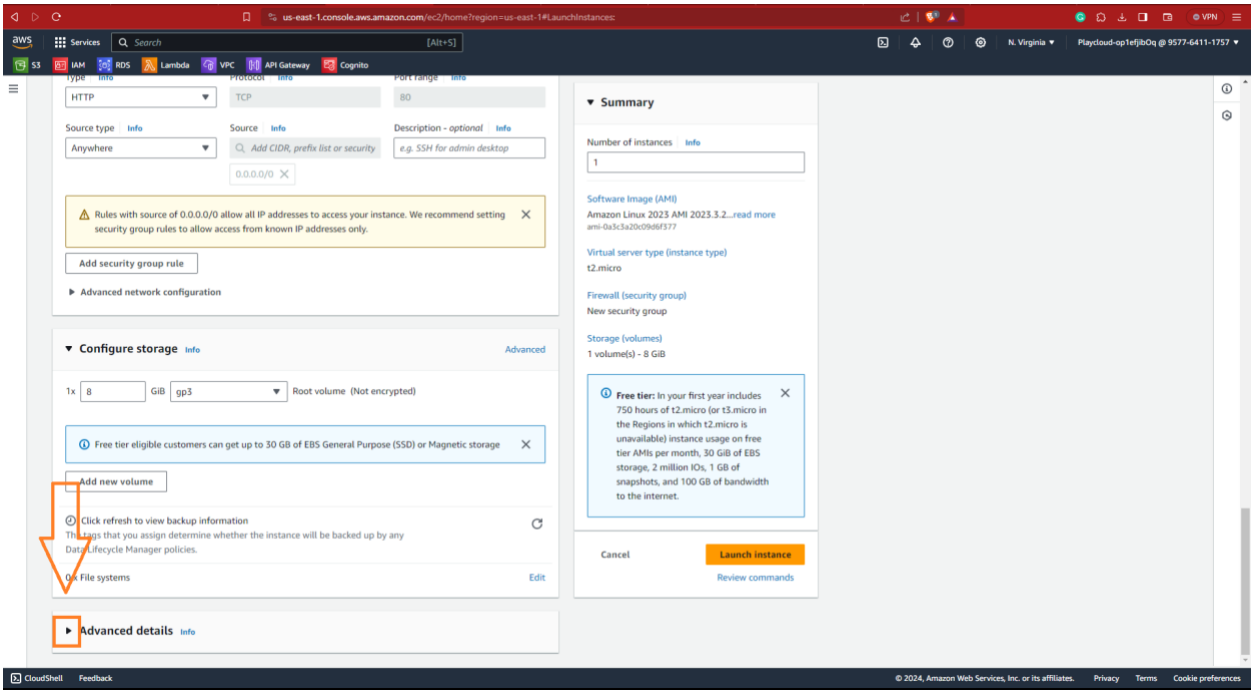
1. Search "***ec2***" in the AWS Management Console search bar. Click **EC2** on the search results.



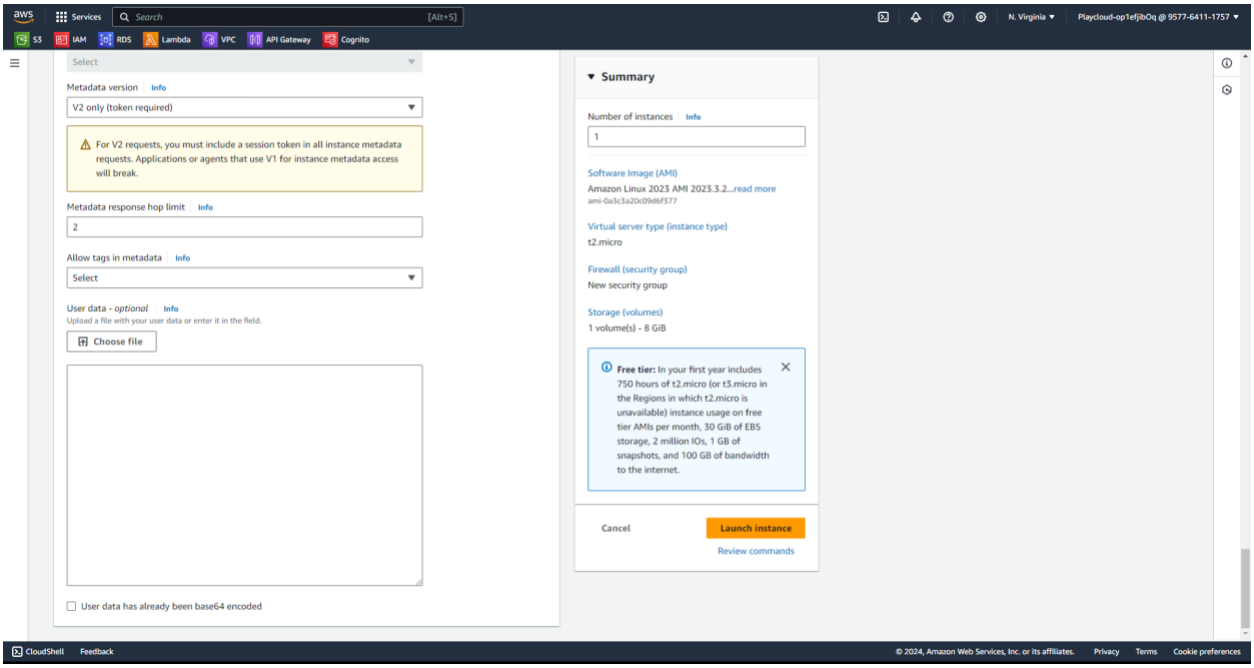
2. Use the following configurations:

- Name: **my-web-server**
- AMI: **Amazon Linux**
- Instance type: **t2.micro**
- Key pair: (**If you don't already have a key pair, please create a new one.**)
    - Key pair name: **web-server-key-pair**
    - Key pair type: **RSA**
    - Private key file format: **.pem**
- Network settings: **(**Click **"Edit")**
    - Security group name: **WebServerSG**
    - Description: enter '***Allows SSH and HTTP access***'.
    - Inbound Security Group Rules: **(Add security group rule)**
        - Inbound rule 1:
            - Type: **SSH**
            - Source Type: **My IP**
        - Inbound rule 2
            - Type: **HTTP**
            - Source Type: **Anywhere (0.0.0.0/0)**

3. Next, click the dropdown of the "**Advanced details."**



4. Scroll down until you see “**User data.- optional”**



5. In the text box, enter the following script:

#!/bin/bash

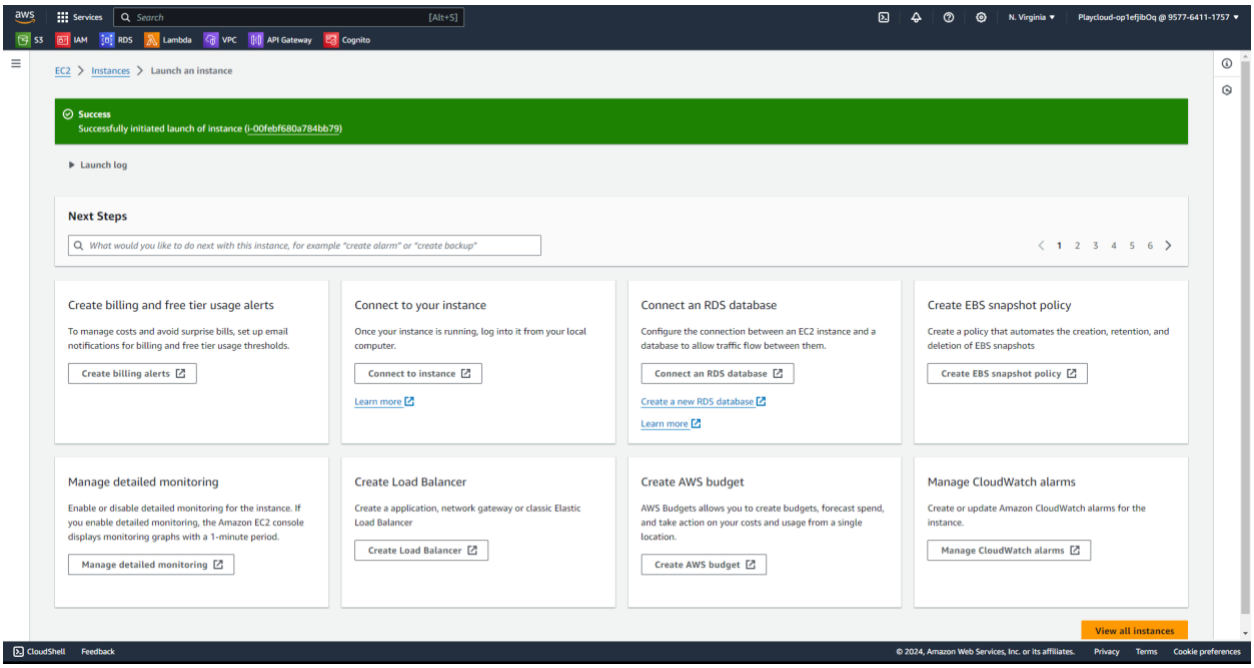
Step 1: Update the system

```
sudo yum update -y

# Step 2: Install Nginx
sudo yum install nginx -y

# Step 3: Start Nginx Service
sudo service nginx start

# Create a custom HTML page directly in the HTML directory
echo '<h1>Welcome to my web page!</h1>' | sudo tee /usr/share/nginx/html/mypage.html
> /dev/null

# Add the configuration directly
echo 'server {
   listen 80 default_server;
   server_name _;
   root /usr/share/nginx/html;

   location / {
      index mypage.html;
   }
}' | sudo tee /etc/nginx/conf.d/server.conf > /dev/null

# Reload Nginx for the changes to take effect
sudo nginx -t && sudo service nginx reload
```

Info: #!/bin/bash is called a shebang or hashbang. It's a special line at the beginning of a script that tells the system which interpreter should be used to execute the script. In this case, it specifies that the script should be interpreted and executed using the Bash shell.

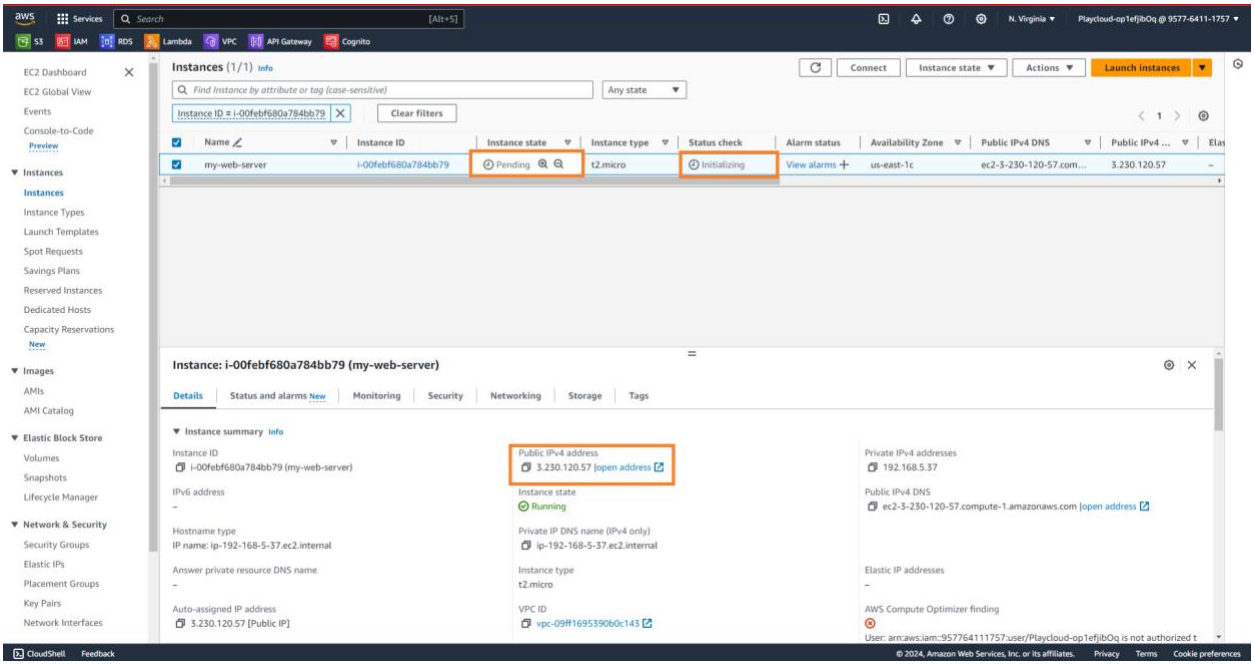
6. Ensure that all are set up correctly, then, click **Launch instance.**

7. You will be notified with the success pop-up. Click the instance id.

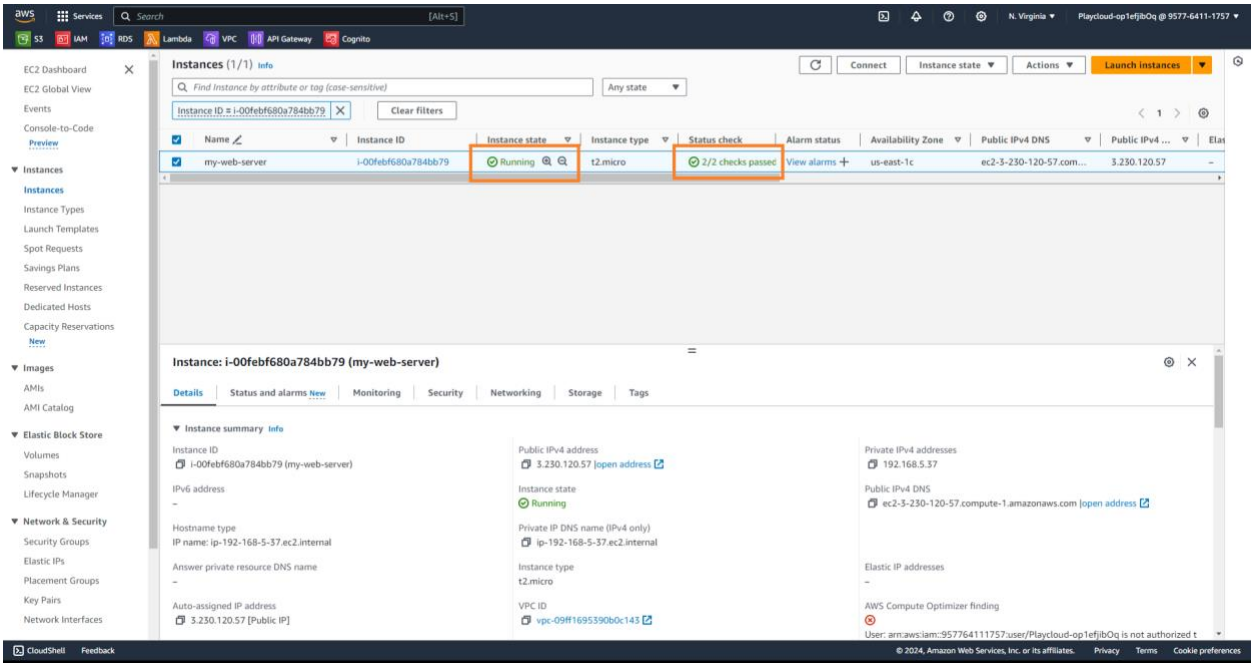


8. Wait for the "**Instance state"** to go from "**Pending"** to "**Running",** and the **"Status check"** from **"Initializing"** to **"2/2 checked passed."**

While waiting, take note and copy the "**Public IPv4 address."**



9. Now that the **"Instance state"** is in **"Running",** and **t**he **"Status check"** is in **"2/2 checked passed."** Paste the **"Public IPv4 address"** you copied earlier to a new tab of your browser and press Enter.



10. Finally, you should see a welcome page.

Welcome to my web page!

Tips: Add to your habit of cleaning up the resources after your labs (i.e. in this case, terminating your Instance). This is not only a good habit but gives you good practice in taking responsibility for your resources.