Cloud Computing Exercise #23

Launching an EC2 instance using a bash script

A. Preparation

1. Sign in to your AWS account as the non-root admin user.
2. In this lab, you will launch an EC2 instance from a bash script. First, you will launch an EC2 instance, the master instance, manually and log into it using ssh. Then, you will write and run a bash script in the master instance that will launch another EC2 instance, the slave EC2 instance. The script will retrieve various system and instance properties, store them in different variables, and output them on the screen.

B. Launch an EC2 instance using the Management Console

1. Before Launching the EC2 Instance, Make sure you have the access keys downloaded on your  
   machine. You should have this if you have followed the previous exercises. If not, go to IAM ->  
   Security Credentials -> and create access keys. Now go to EC2 and launch a new instance using  
   the “Amazon Linux 2023 AMI”, Add a tag with key field “Name” and value field “Master EC2  
   instance” and the t2.micro instance type. You can use the default 8 GB EBS volume settings and  
   the default security group settings (port 22 open).. Launch the EC2 instance, and connect to it  
   using your local SSH client/Putty (the user name is “ec2-user”). This AMI has the AWS CLI client  
   already installed, so there is no need to manually install it. Make note of the Instance ID, Key pair attached to the instance, and the availability zone where the instance was launched. These 3 things will need to be entered as a part of our script

C. Configure the AWS CLI

1. SSH into the instance, Start the CLI configuration process by entering the following command on  
   the EC2 ssh console:aws configure

Copy and paste the access key ID and the secret access key when prompted.

D. Write and run a bash script that launches an EC2 instance

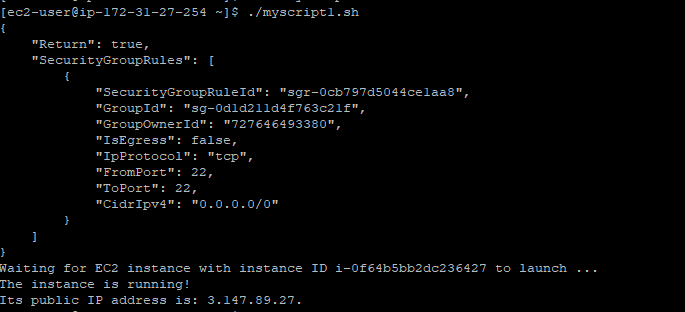
1. Start a text editor (e.g. nano) in the SSH console of the master EC2 instance, and create a bash script (e.g. named myscript.sh) that goes through the below steps. Whenever you save some output in a variable, format the AWS CLI command output as text (--output option) to avoid unwanted characters (e.g. “{“, “}” and quotes) in the output. If you want to store the output of a CLI command in a variable, use the “<variable>=$(<command>)” syntax in your script.
   * Store the AMI ID of the Amazon Linux 2 AMI in a variable called AMIID.
   * Store the ID of the default VPC in a variable called VPCID by issuing the “aws ec2 describe-vpcs” CLI command. Use server-side filtering (--filters option) to retrieve only the default VPC’s information, and use client-side filtering (--query option) to only output the VPC ID of the VPC.
   * Store the subnet ID of the default VPC’s subnet in the “us-east-1” availability zone in a variable called SUBNETID by issuing the “aws ec2 describe-subnets” CLI command. Use server-side filtering (--filters option) to only retrieve the subnet information for the subnet in the us-east-1a availability zone of the default VPC and use the value of the VPCID variable in the filtering expression. Use client-side filtering (--query option) to only output the subnet ID of the subnet.
   * Create a new security group called “My security group” and save its security group ID in a variable called SGID by issuing the “aws ec2 create-security-group” CLI command. Use the value of the VPCID variable with the --vpc-id option to specify the VPC to create the security group in.
   * Configure the security group to allow incoming SSH traffic from any source IP address by issuing the ”aws ec2 authorize-security-group-ingress” CLI command. Use the value of the SGID variable with the --group-id option to specify the security group for the firewall rule.
   * Launch a new EC2 instance and save its instance ID in a variable called INSTANCEID by issuing the CLI command “aws ec2 run-instances” with the following parameters:
     + Image ID: the value of the AMIID variable
     + Key name: the name of the key pair you want to use (e.g. mykey)
     + Instance type: t2.micro
     + Security group ID: the value of the SGID variable
     + Subnet ID: the value of the SUBNETID variable
     + Block device mapping parameters: device name: /dev/xvda, the EBS volume should be deleted on instance termination; volume size: 10 GByte,
     + Tags: Key: “Name”, Value: “Slave EC2 instance”

Use client-side filtering (--query option) to only output the instance ID of the newly launched instance.

* Print "Waiting for EC2 instance with instance ID <instance ID> to launch ..." on the screen, where <instance ID> is the value of the INSTANCEID variable. Then, wait until the launch process is finished and the new instance has transitioned to “running” state by issuing the “aws ec2 wait instance-running” CLI command. Use the value of the INSTANCEID variable with the –instance-ids option to specify the instance ID.
* Print “The instance is running!” on the screen.
* Save the public IP address of the running slave EC2 instance in a variable called PUBLICIP by issuing the “aws ec2 describe-instances” CLI command. Use the value of the INSTANCEID variable with the --instance-ids option to specify the instance ID. Filter the output (--query option) to only output the public IP address of the instance.
* Print "Its public IP address is: <public IP>." on the screen, where <public IP> is the value of the PUBLICIP variable.

**If you need help with the script, I have attached the script in the modules section. Kindly go through it.**

1. Save the script, and change the permissions on the script file so that the owning user (you) could run it as a command. Then, run the script. If it runs correctly, the script output should be similar to this:



If you go to the Management Console (EC2/Instances/Instances), you should see both the master and the slave EC2 instances in “running” state.

A screenshot of a computer

Description automatically generated

E. Clean up after yourself

1. Close the ssh window and terminate the both the slave and the master EC2 instances from the Management Console. Delete the security groups you created. Check and make sure that there are no EBS volumes that survived the instance termination.
2. Log out of AWS.