DevOps AWS Exercise for- DevOps Support Escalation Position

Jason Taylor

Overview

I have provided two code examples that accomplished the requirements for this exercise. The first written using Ansible. This example runs on Linux only. The second I scripted using Terraform and runs on both Windows and Linux.

Assumptions I have made

- When passing the AWS Credentials, I am using the AWS Access Key and the AWS Secret Key not the AWS username and password.
- I coded the examples to use the region us-east-2.
- I am not checking the Instance Type to see if it is valid. I am assuming you will enter a valid Instance Type.

URLs to the video example (URLs will expire 02/15/2021)

- Ansible: https://objectstorage.us-ashburn-1.oraclecloud.com/p/X80-7xOobLqvbo-SERXXhOnZIi5kktbCxSpON1-Ov2sIJPRCp1xJLmpXTiEHt1PZ/n/idl9lvumysr2/b/bucket-20210103-2039/o/devops_aws_exercise_ansible.mp4
- Terraform on Linux: https://objectstorage.us-ashburn-1.oraclecloud.com/p/1ogo43qPM1aUPDXs_6FxZjVIQgLPkvk3EFcoOaJJbFGH3SaDXfMYsNd5Drr_WDfX/n/idl9lvumysr2/b/bucket-20210103-2039/o/devops_aws_exercise_terraform.mp4
- Terraform on Windows: https://objectstorage.us-ashburn-1.oraclecloud.com/p/1ogo43qPM1aUPDXs_6FxZjVIQgLPkvk3EFcoOaJJbFGH3SaDXfMYsNd5Drr_WDfX/n/idl9lvumysr2/b/bucket-20210103-2039/o/devops_aws_exercise_terraform.mp4

I did not have tools to edit the videos so I could not hide the AWS Access and Secret keys that were used in the videos. The access and secret keys that were used in the video have been deleted.

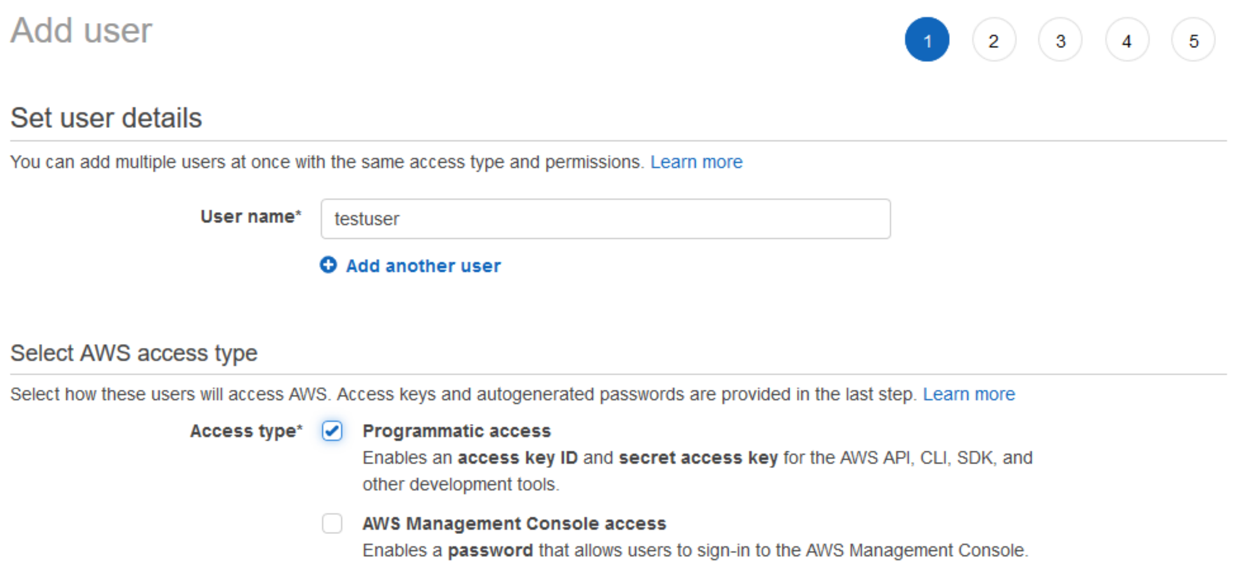
URL to the Source Code (URLs will expire 02/15/2021)

https://objectstorage.us-ashburn-1.oraclecloud.com/p/wCbD7xSw_bHQA_FnMV3MPhDPueEVqfzRvNpliStj-zK043Ad_j3B_t6eVaZ4hL_H/n/idl9lvumysr2/b/bucket-20210103-2039/o/devops_aws_exercise.zip

Setup Tasks

IAM

The first step is to create an AWS user. Login to AWS and go to the IAM service. Then go to users and click the Add User Button. Next define a username and check the Programmatic access check box under Access Type and then click Next.



Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* testuser

Add another user

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

Access type* [✓] **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

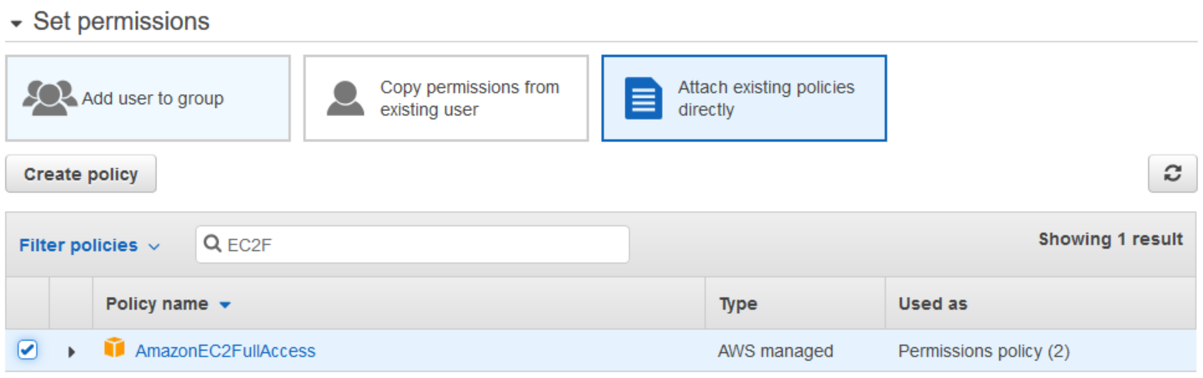
[ ] **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

Under the Set Permissions dialog select Attach existing policies directly and add the following policies.

1. AmazonEC2FullAccess
2. AmazonS3FullAccess
3. IAMFullAccess

The Ansible example only needs the AmazonEC2FullAccess policy where the terraform needs all three policies.

Once the three policies are selected click the Next button.



Set permissions

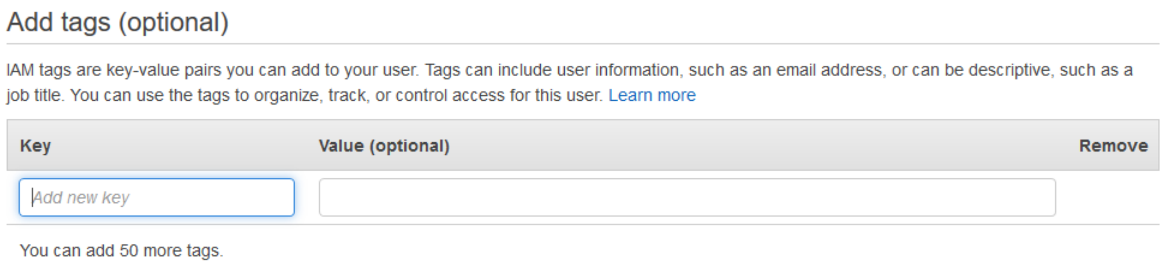
Add user to group	Copy permissions from existing user	Attach existing policies directly

Create policy

Filter policies | EC2F | Showing 1 result

		Policy name	Type	Used as
✓	▸	AmazonEC2FullAccess	AWS managed	Permissions policy (2)

Add tags if desired and click next.



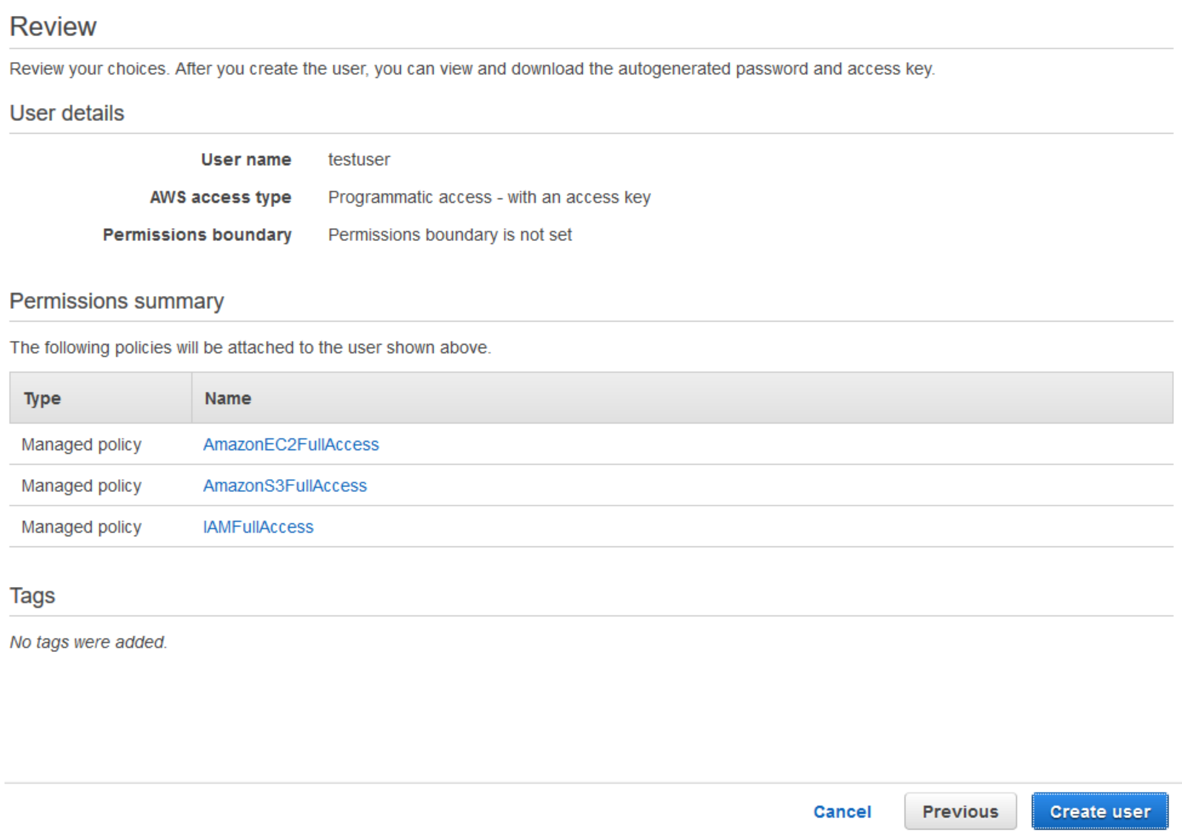
Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. Learn more

Key	Value (optional)	Remove
Add new key		

You can add 50 more tags.

Review the user setting and then click the Create User button.



Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	testuser
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Permissions summary

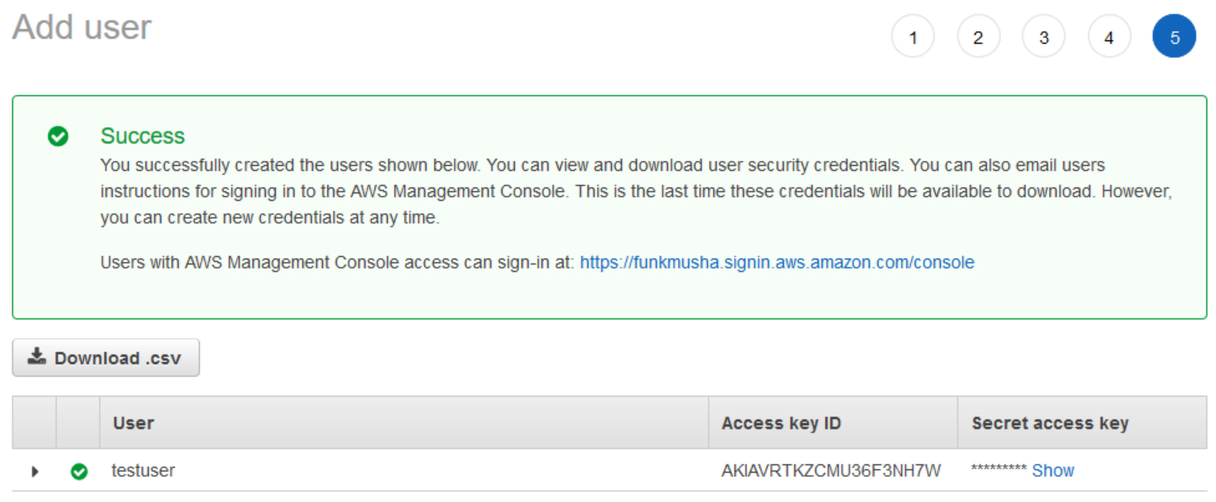
The following policies will be attached to the user shown above.

Type	Name
Managed policy	AmazonEC2FullAccess
Managed policy	AmazonS3FullAccess
Managed policy	IAMFullAccess

Tags

No tags were added.

Copy the Access key ID and Secret access key or download the csv by clicking the Download .csv button.



Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: https://funkmusha.signin.aws.amazon.com/console

		User	Access key ID	Secret access key
▸	✓	testuser	AKIAVRTKZCMU36F3NH7W	********* Show

The Access key ID and Secret access key will be used to run the two exercises.

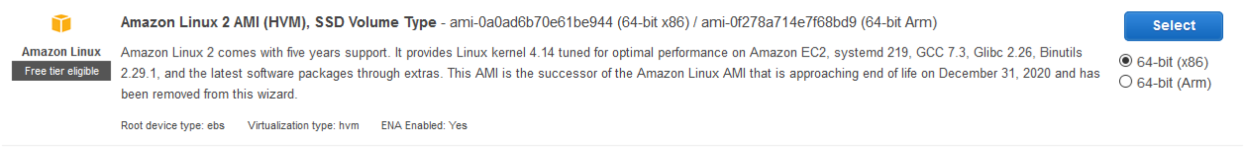
Ansible

Note: The ansible exercise only works on Linux.

Create a bastion server in AWS to run the Ansible exercise.

Login to AWS and go to the EC2 service. Next click Instances (left hand side) and click Launch Instances button (top right).

Select the AIM: ami-0a0ad6b70e61be944 (64-bit x86) and click the Select button.

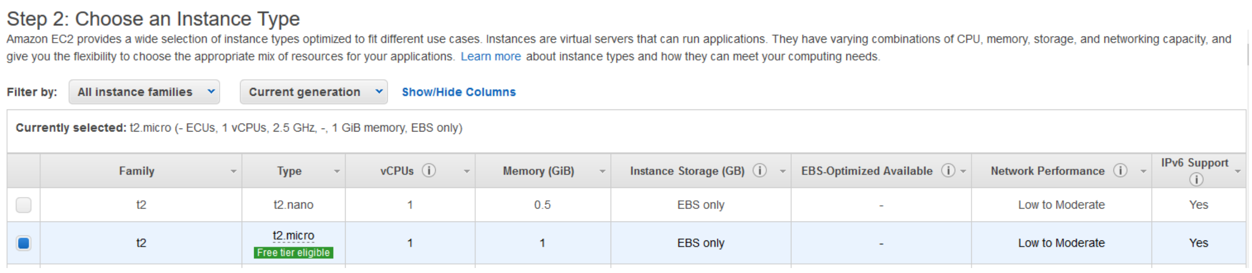


Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0a0ad6b70e61be944 (64-bit x86) / ami-0f278a714e7f68bd9 (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is approaching end of life on December 31, 2020 and has been removed from this wizard.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Select the instance type and click the Next button.



Step 2: Choose an Instance Type

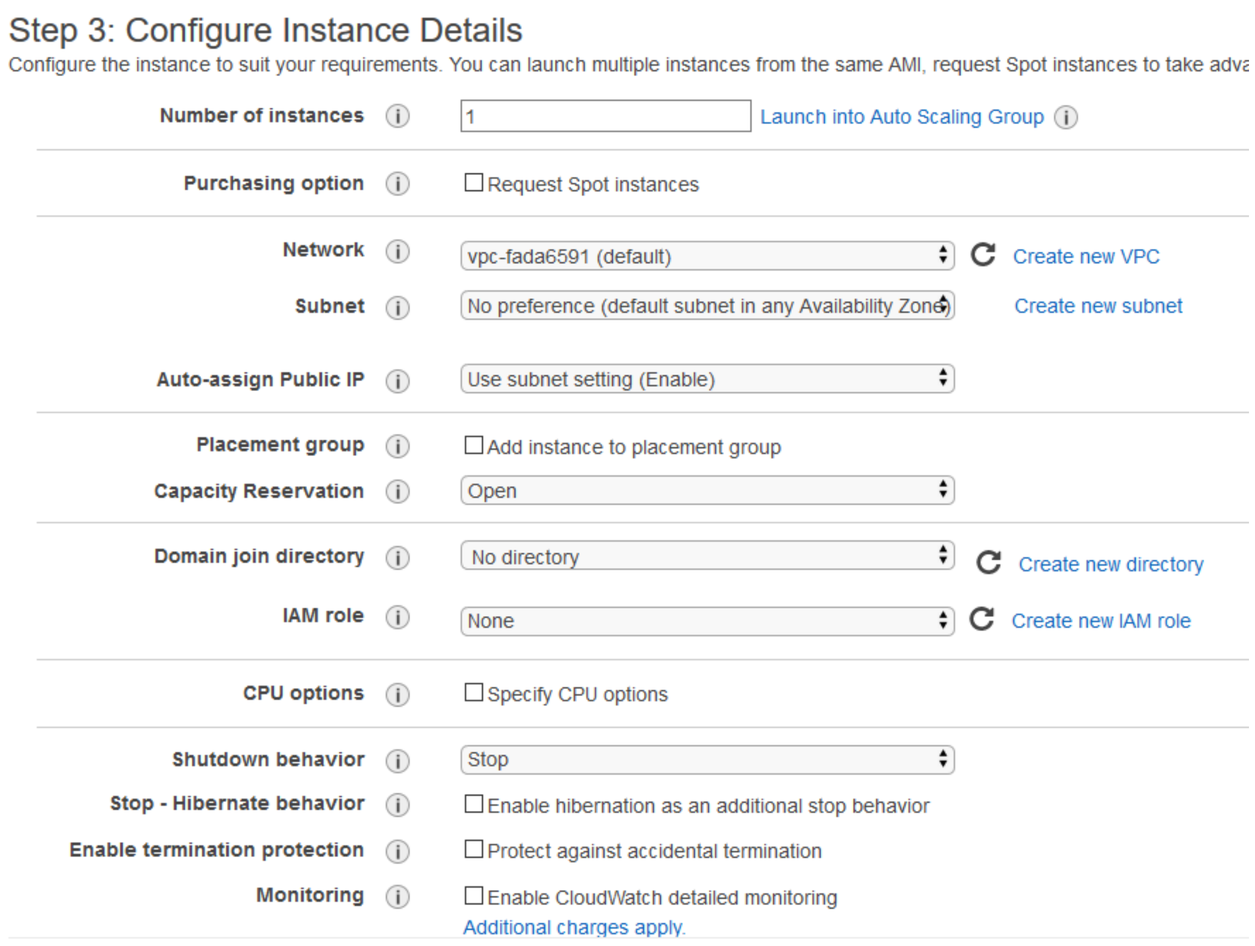
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance families · Current generation · Show/Hide Columns

Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
☐	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
■	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes

Make any changes necessary for the Step 3: Configure Instance Details dialog and click the Next button.

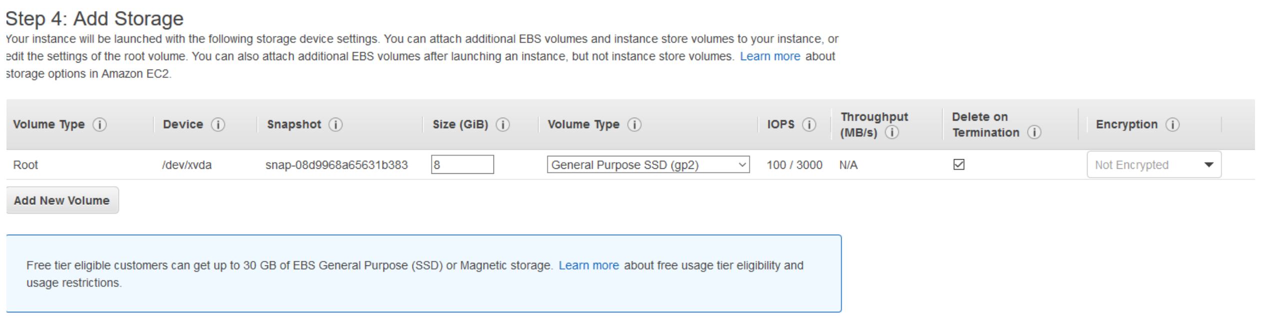


Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take adva

Number of instances	1 Launch into Auto Scaling Group
Purchasing option	☐ Request Spot instances
Network	vpc-fada6591 (default) Create new VPC
Subnet	No preference (default subnet in any Availability Zone) Create new subnet
Auto-assign Public IP	Use subnet setting (Enable)
Placement group	☐ Add instance to placement group
Capacity Reservation	Open
Domain join directory	No directory Create new directory
IAM role	None Create new IAM role
CPU options	☐ Specify CPU options
Shutdown behavior	Stop
Stop - Hibernate behavior	☐ Enable hibernation as an additional stop behavior
Enable termination protection	☐ Protect against accidental termination
Monitoring	☐ Enable CloudWatch detailed monitoring Additional charges apply.

On the Add Storage page make any necessary changes and click the Next button.



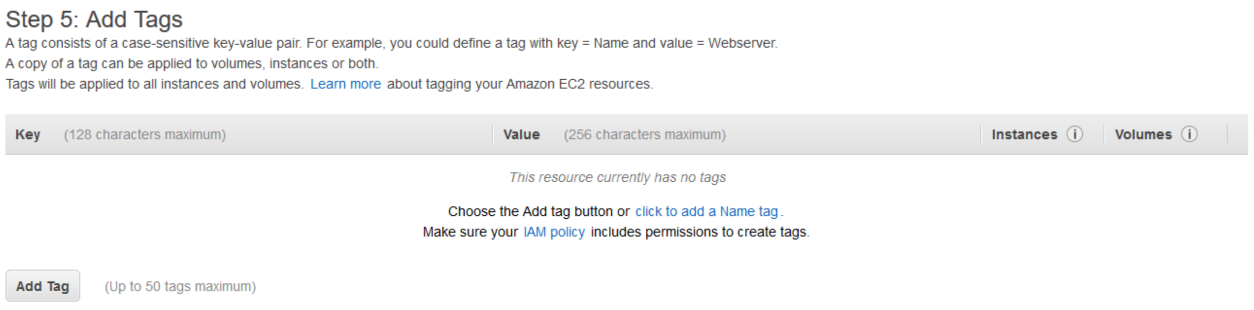
Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-08d9968a65631b383	8	General Purpose SSD (gp2)	100 / 3000	N/A	✓	Not Encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

Add tags if necessary and click the Next button.



Step 5: Add Tags

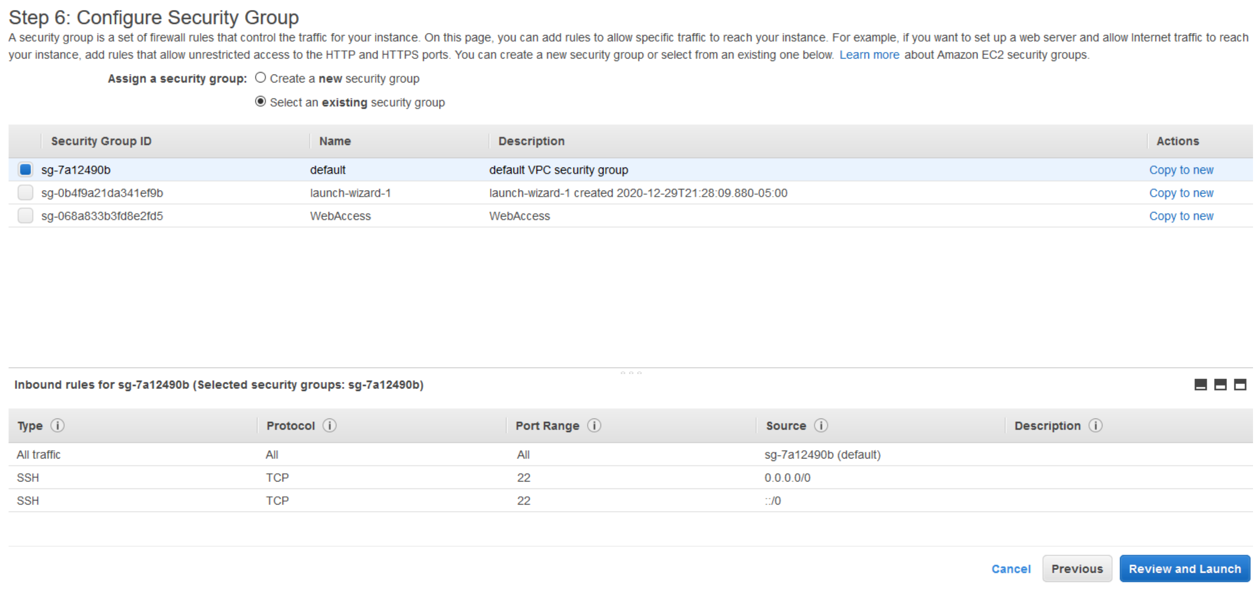
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes

This resource currently has no tags

Choose the Add tag button or click to add a Name tag.
Make sure your IAM policy includes permissions to create tags.

Create or select an existing Security Group that has SSH access setup and the click the Review and Launch button.



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: Create a **new** security group [✓ Select an **existing** security group]

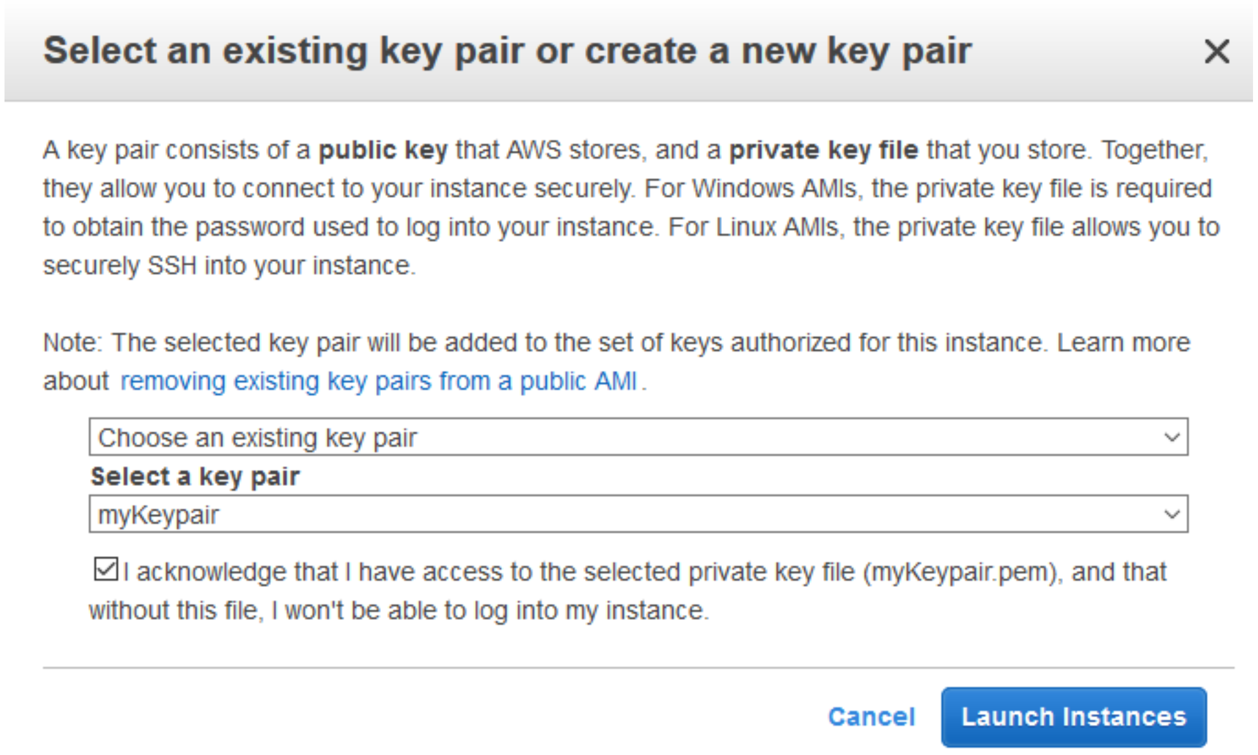
	Security Group ID	Name	Description	Actions
✓	sg-7a12490b	default	default VPC security group	Copy to new
	sg-0b4f9a21da341ef9b	launch-wizard-1	launch-wizard-1 created 2020-12-29T21:28:09.880-05:00	Copy to new
	sg-068a833b3fd8e2fd5	WebAccess	WebAccess	Copy to new

Inbound rules for sg-7a12490b (Selected security groups: sg-7a12490b)

Type	Protocol	Port Range	Source	Description
All traffic	All	All	sg-7a12490b (default)	
SSH	TCP	22	0.0.0.0/0	
SSH	TCP	22	::/0	

After Reviewing the instance details click the launch button.

Select an existing key pair or create a new key pair and click the Launch Instances button.



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Choose an existing key pair

Select a key pair

myKeypair

☑ I acknowledge that I have access to the selected private key file (myKeypair.pem), and that without this file, I won't be able to log into my instance.

Once the EC2 instance is running login.

Install the required software:

```
sudo yum install python3 -y
sudo pip3 install boto boto3 ansible
```

Terraform

Linux

Use the same EC2 Linux instance created for the Ansible exercise.

Install the required software:

```
wget https://releases.hashicorp.com/terraform/0.14.3/terraform_0.14.3_linux_amd64.zip
sudo unzip terraform_0.14.3_linux_amd64.zip -d /bin
```

Verify the Terraform installation:

```
terraform -version
```

Windows

Login to your windows environment.

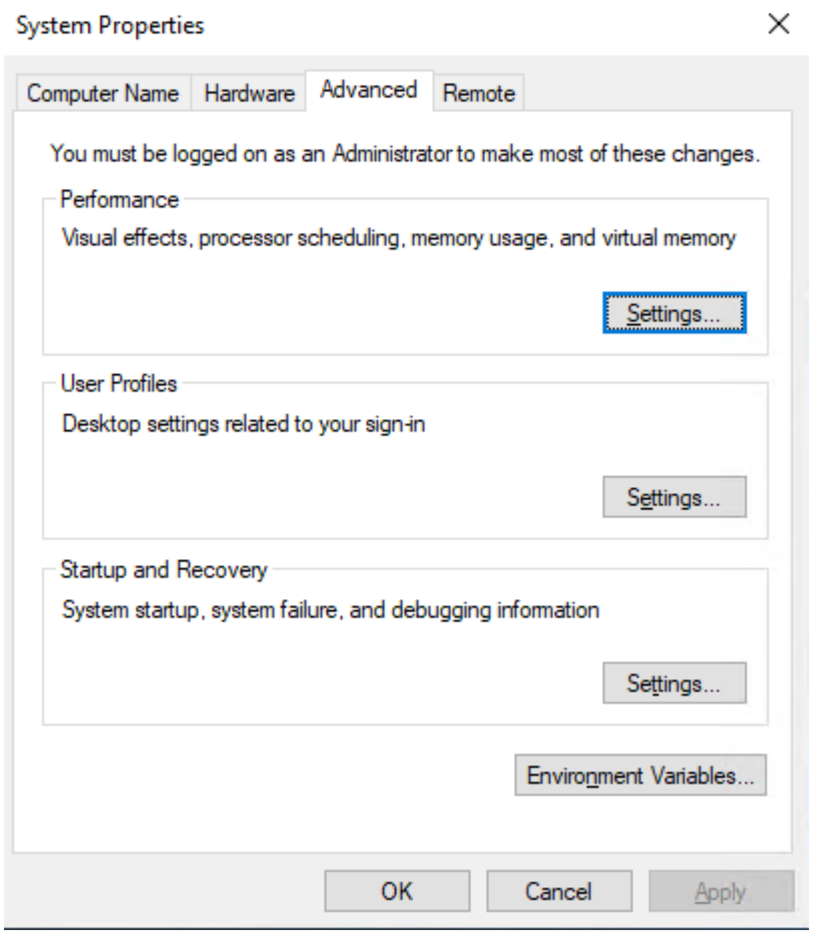
Install Terraform.

Download the following file and extract it to the local file system.

https://releases.hashicorp.com/terraform/0.14.3/terraform_0.14.3_windows_amd64.zip

Add the path to the terraform executable to the system PATH.

From System Properties click the Environment Variables button.



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

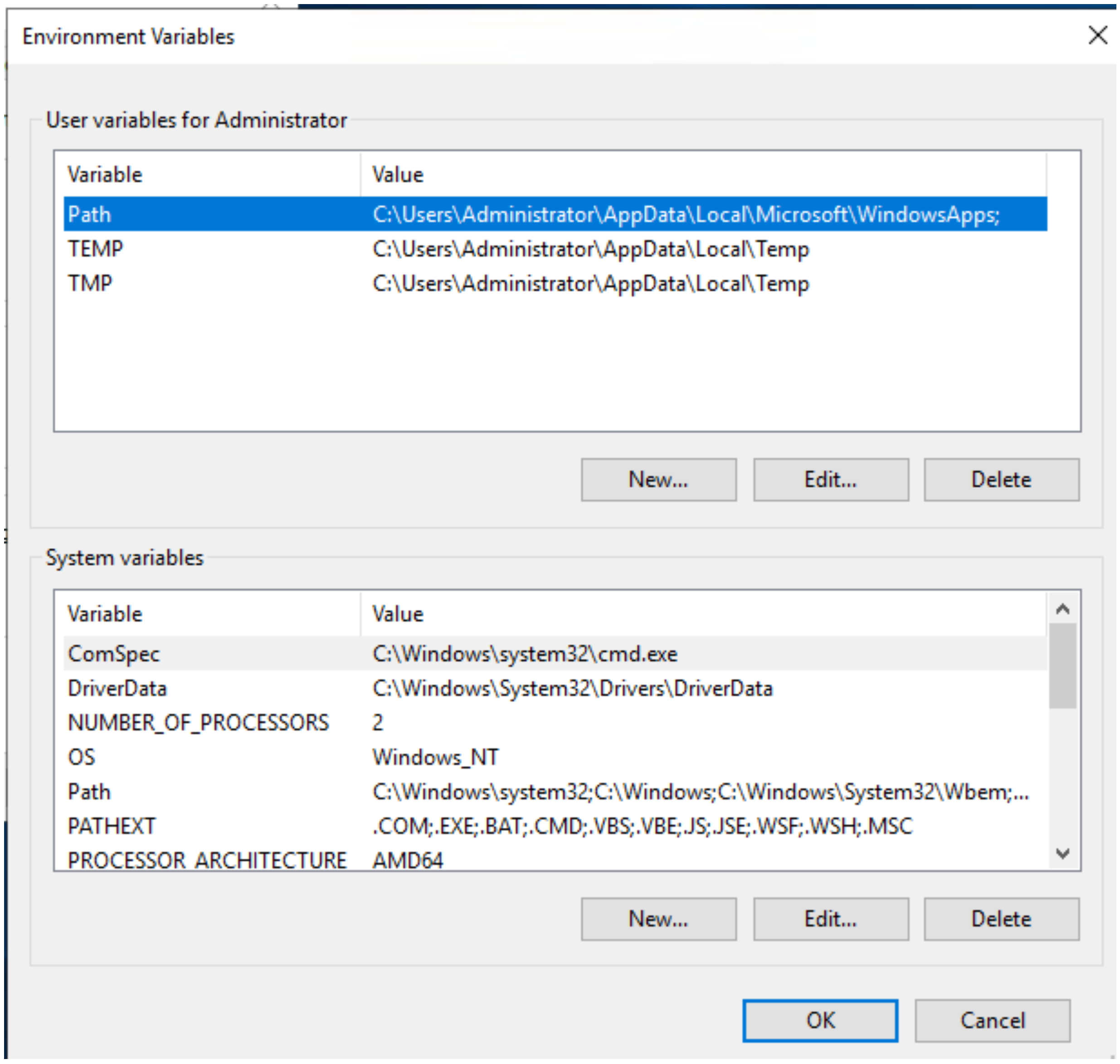
User Profiles

Desktop settings related to your sign-in

Startup and Recovery

System startup, system failure, and debugging information

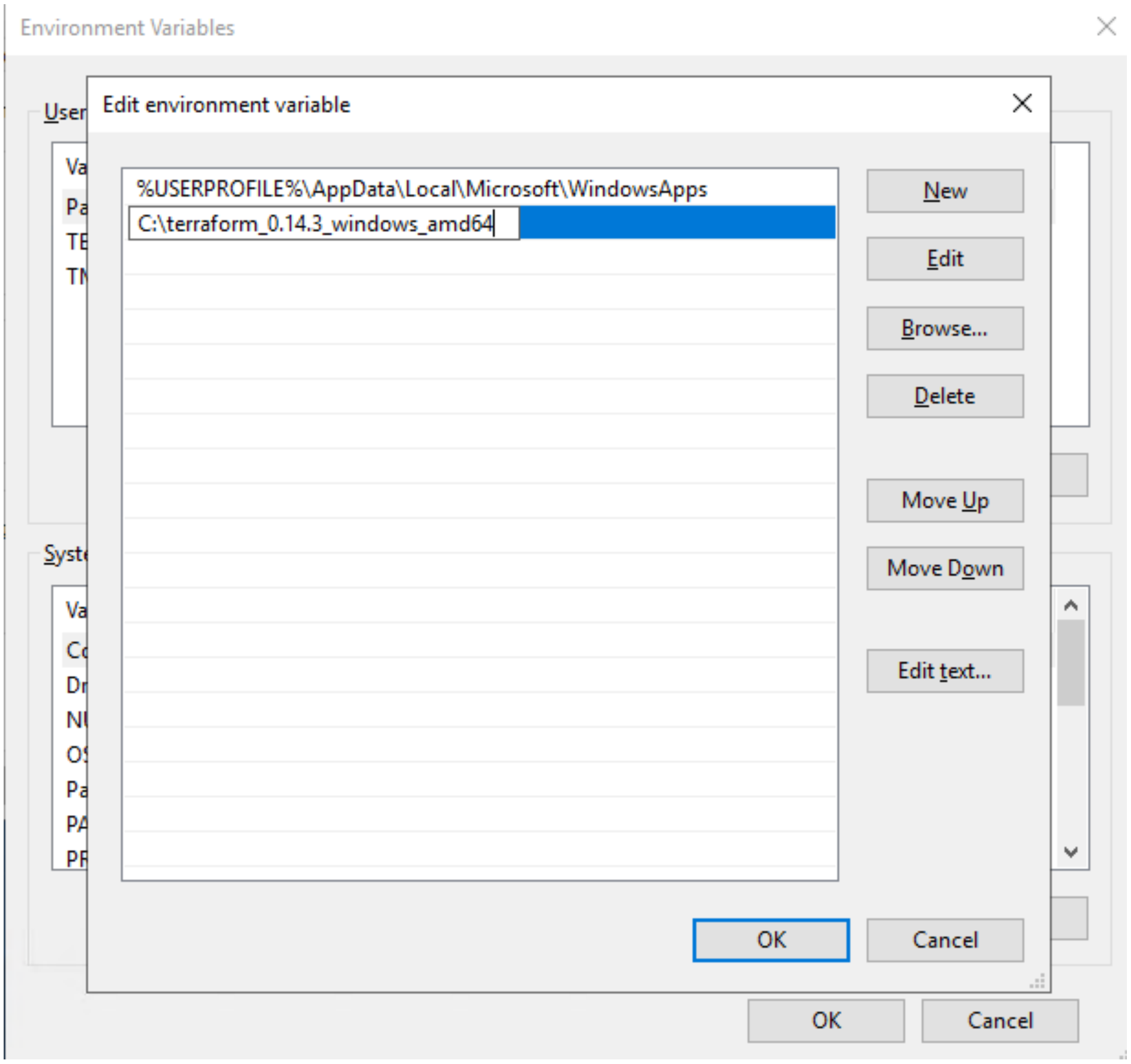
Highlight the Path User variable and click Edit.



Variable	Value
Path	C:\Users\Administrator\AppData\Local\Microsoft\WindowsApps;
TEMP	C:\Users\Administrator\AppData\Local\Temp
TMP	C:\Users\Administrator\AppData\Local\Temp

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	2
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

Add the path to the terraform executable by clicking New and pasting in the path. Then click OK.



```
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\terraform_0.14.3_windows_amd64
```

Install AWS CLI using the instruction from the following page.

https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-windows.html

Or just download the MSI from here: https://awscli.amazonaws.com/AWSCLIV2.msi

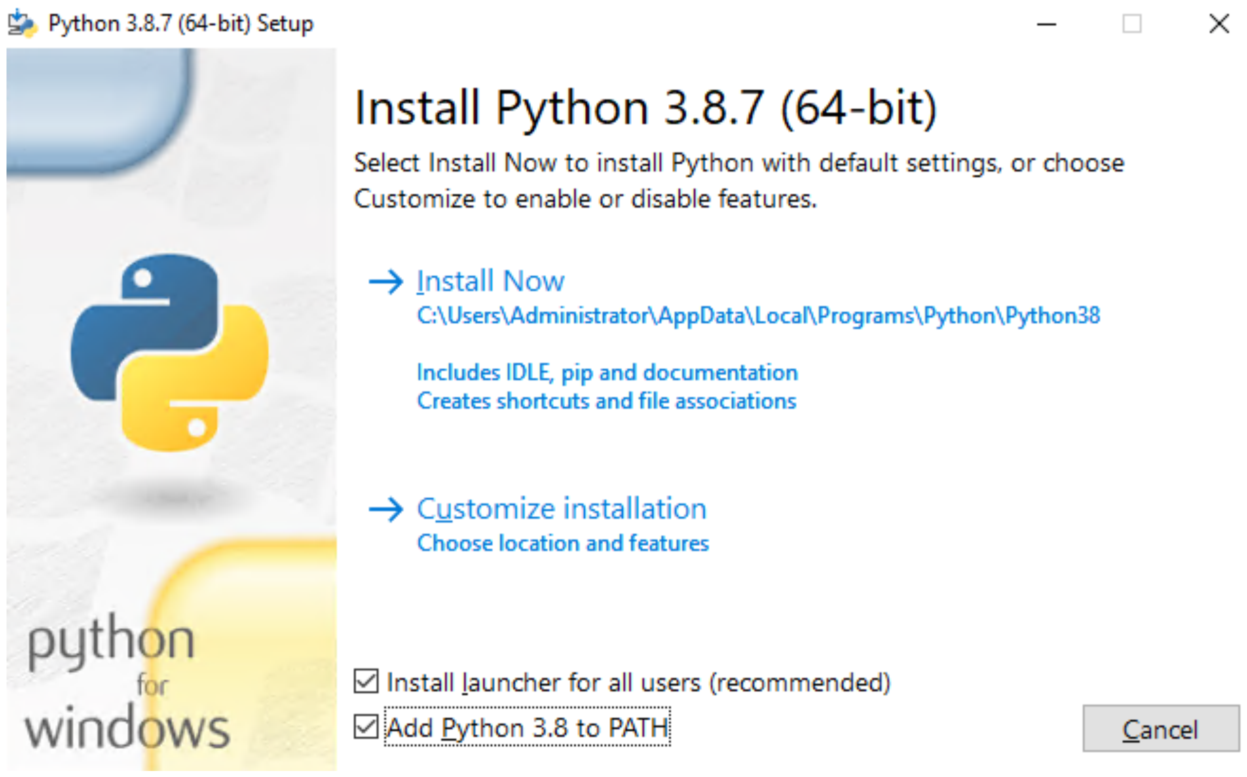
Once the AWS CLI install file has downloaded right click the file and click install. Use the default options for the installation.

Install Python 3

Download the following file

https://www.python.org/ftp/python/3.8.7/python-3.8.7-amd64.exe

Run the install as an administrator. Select Add Python to PATH and click Install Now.



Python 3.8.7 (64-bit) Setup

Install Python 3.8.7 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

→ Install Now
C:\Users\Administrator\AppData\Local\Programs\Python\Python38

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.8 to PATH

Open PowerShell as an Administrator and run:

```
pip3 install boto boto3
```

Download source code

Linux

```
wget https://objectstorage.us-ashburn-1.oraclecloud.com/p/wCbD7xSw_bHQA_FnMV3MPhDPueEVqfzRvNpliStj-zK043Ad_j3B_t6eVaZ4hL_H/n/idl9lvumysr2/b/bucket-20210103-2039/o/devops_aws_exercise.zip
```

```
unzip devops_aws_exercise.zip
```

Windows

Download the zip file from the following link and extract the files.

https://objectstorage.us-ashburn-1.oraclecloud.com/p/wCbD7xSw_bHQA_FnMV3MPhDPueEVqfzRvNpliStj-zK043Ad_j3B_t6eVaZ4hL_H/n/idl9lvumysr2/b/bucket-20210103-2039/o/devops_aws_exercise.zip

Configure the DevOps exercises

Ansible

After extracting the source code change to the devops_aws_exercise/ansible directory.

```
cd devops_aws_exercise/ansible/
chmod 755 configure.sh
chmod 755 run_aws_exercise.sh
./configure.sh
```

Terraform

Linux

After extracting the source code change to the devops_aws_exercise/terraform directory.

```
cd devops_aws_exercise/terraform/
terraform init
```

Run configure.py or create a Key Pair named devops_aws_exercise_tf and download it to the devops_aws_exercise/terraform directory.

To create the key pair using configure.py we first need to configure the aws cli.

Run:

```
aws configure
```

When prompted enter the AWS Access Key ID, AWS Secret Access Key, and Default region name. **Set the region to us-east-2.**

Once aws cli has been configured run the python script configure.py and change the permissions on the downloaded key file.

Run:

```
python3 configure.py
chmod 600 devops_aws_exercise_tf.pem
```

Windows

After extracting the source code change to the devops_aws_exercise/terraform directory.

```
cd devops_aws_exercise/terraform/
terraform init
```

Run configure.py or create a Key Pair named devops_aws_exercise_tf and download it to the devops_aws_exercise/terraform directory.

To create the key pair using configure.py we first need to configure the aws cli.

Run:

```
aws configure
```

When prompted enter the AWS Access Key ID, AWS Secret Access Key, and Default region name. **Set the region to us-east-2.**

Once aws cli has been configured run the python script configure.py and change the permissions on the downloaded key file.

Run:

```
python configure.py
```

Run the DevOps exercises

Note: Access Keys and Secret Kes in these examples have been deleted.

Ansible

Example:

```
./run_aws_exercise.sh aws_access_key_id aws_secret_access_key instance_size instance_name file
```

Run the following updating the parameters for your environment:

```
./run_aws_exercise.sh AKIXXXXXXXXXXXXD5GI RO99kDS32332frgF+Mr8R8F/jGZRG2Ym30Lw46 t2.micro aws_test1 /home/ec2-user/test.txt
```

Terraform

Linux

Change to the directory that holds the terraform code.

Run the following updating the parameters for your environment:

```
terraform apply -var 'access_key=AKIXXXXXXXXXXXXD5GI' -var 'secret_key=RO99kDS32332frgF+Mr8R8F/jGZRG2Ym30Lw46' -var 'uploadFile=/home/ec2-user/test.txt' -var 'instance_size=t2.micro' -var 'instance_name=aws_test' -auto-approve
```

Windows

Start Powershell as an administrator.

Change to the directory that holds the terraform code.

Run the following updating the parameters for your environment:

```
terraform apply -var 'access_key=AKIXXXXXXXXXXXXD5GI' -var 'secret_key=RO99kDS32332frgF+Mr8R8F/jGZRG2Ym30Lw46' -var 'uploadFile=C:\devops_aws_exercise\terraform\test.txt' -var 'instance_size=t2.micro' -var 'instance_name=aws_test' -auto-approve
```

Resource Cleanup

For the terraform exercise you can remove the EC2 instance and all the resources by running the following. Update the parameters for your environment.

```
terraform destroy -var 'access_key=AKIXXXXXXXXXXXXD5GI' -var
'secret_key=RO99kDS32332frgF+Mr8R8F/jGZRG2Ym30Lw46' -var
'uploadFile=C:\devops_aws_exercise\terraform\test.txt' -var 'instance_size=t2.micro' -var
'instance_name=aws_test' -auto-approve
```

For the Ansible exercise you will need to manually remove the EC2 instance and all the resources that were created.