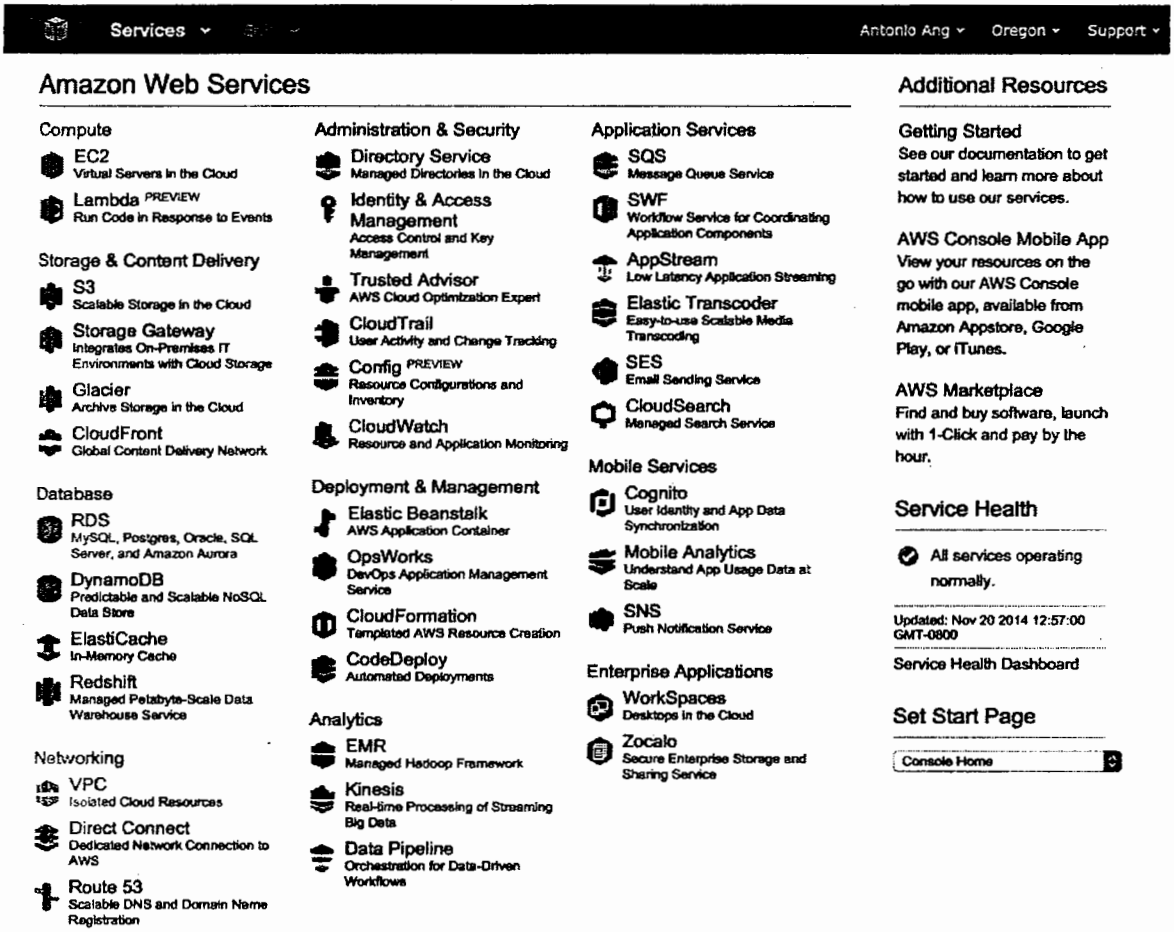
Lab 1

CREATE YOUR FIRST AMAZON EC2 INSTANCE (LINUX)

STEP 1: Log In to the Amazon Web Service Console

This laboratory experience is about Amazon Web Services and you will use the AWS Management Console in order to complete all the lab steps.



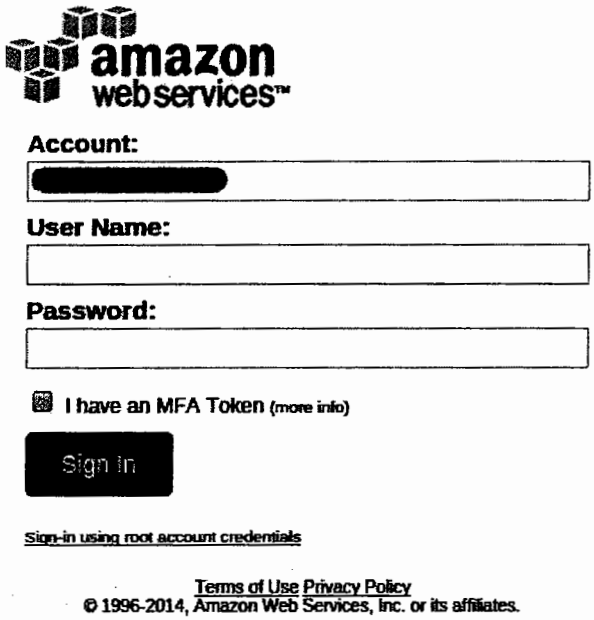
The AWS Management Console is a web control panel for managing all your AWS resources, from EC2 instances to SNS topics. The console enables cloud management for all aspects of the AWS account, including managing security credentials, or even setting up new IAM Users.

Log in to the AWS Management Console

In order to start the laboratory experience, open the Amazon Console by clicking this button:

Open AWS Console

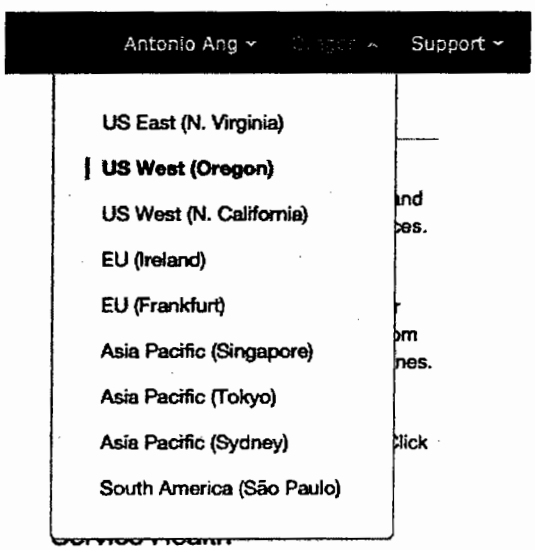
Log in with the username **xxxx** and the password **xxxx**



Select the right AWS Region

Amazon Web Services is available in different regions all over the world, and the console lets you provision resources across multiple regions. You usually choose a region that best suits your business needs to optimize your customer's experience, but you must use the region **US West (Oregon)** for this laboratory.

You can select the **US West (Oregon)** region using the upper right dropdown menu on the AWS Console page.

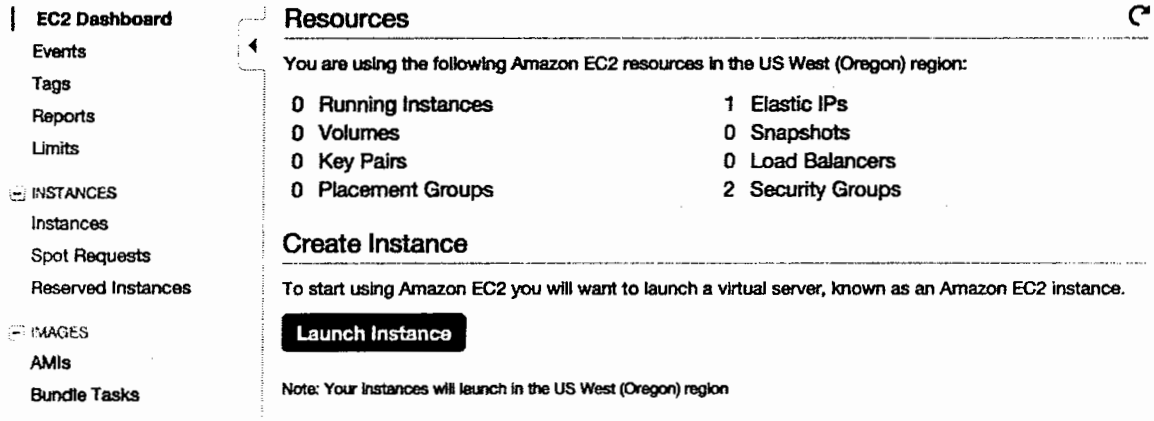


STEP 2: Create an EC2 instance

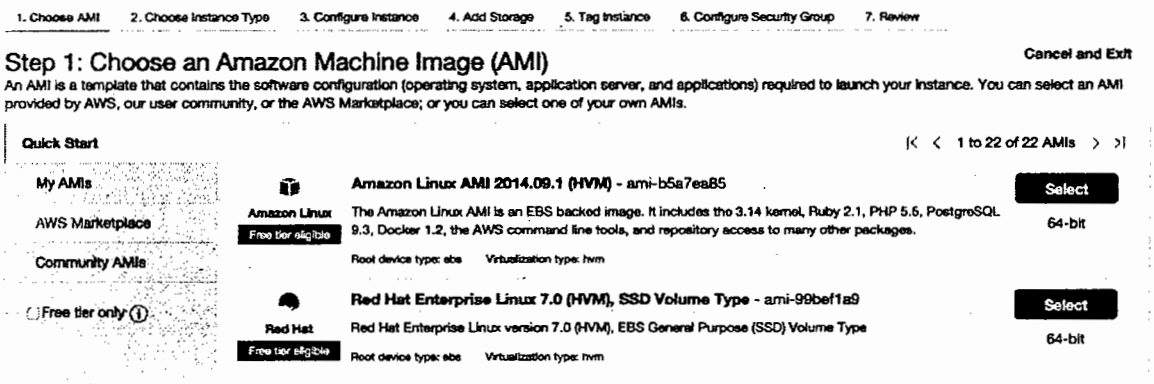
You can launch an EC2 instance using the EC2 launch wizard.

Select the EC2 service from the Management Console dashboard:

From the EC2 dashboard, click **Launch Instance**.



The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations called **Amazon Machine Images (AMIs)** that serve as templates for your instance. Select the first listed 64-bit **Amazon Linux AMI**.



On the **Choose an Instance Type** page, do **not** change any options and click **Next : Configure Instance Details.**

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Tag Instance | 6. Configure Security Group | 7. Review

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 types | Current generation | Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
■	General purpose	t2.micro (Free tier eligible)	1	1	EBS only	-	Low to Moderate
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate

Cancel | Previous | Review and Launch | Next: Configure Instance Details

On the **Configure Instance Details** tab, check the selected **Network (VPC)** and **Subnet**. Change them, if needed, and then click **Next : Add Storage**.

1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Tag Instance | 6. Configure Security Group | 7. Review

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 advantage of the lower pricing, assign an access management role to the instance, and more.

Number of Instances: 1

Purchasing option: ☐ Request Spot Instances

Network: vpc-e144e4c4 (10.0.0.0/16) | Create new VPC

Subnet: subnet-5931ea2e(10.0.0.0/24) | Public-A | us-west-2a | Create new subnet

251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Disable)

IAM role: None

Shutdown behavior: Stop

Enable termination protection: ☐ Protect against accidental termination

Monitoring: ☐ Enable CloudWatch detailed monitoring

Additional charges apply.

Tenancy: Shared tenancy (multi-tenant hardware)

Additional charges will apply for dedicated tenancy.

▾ Network interfaces

Device	Network interface	Subnet	Primary IP	Secondary IP addresses
eth0	New network interface	subnet-5931ea2e (F	Auto-assign	Add IP

Add Device

Cancel | Previous | Review and Launch | Next: Add Storage

On the **Add Storage** tab, do **not** change any options, and click the **Review and Launch** button.

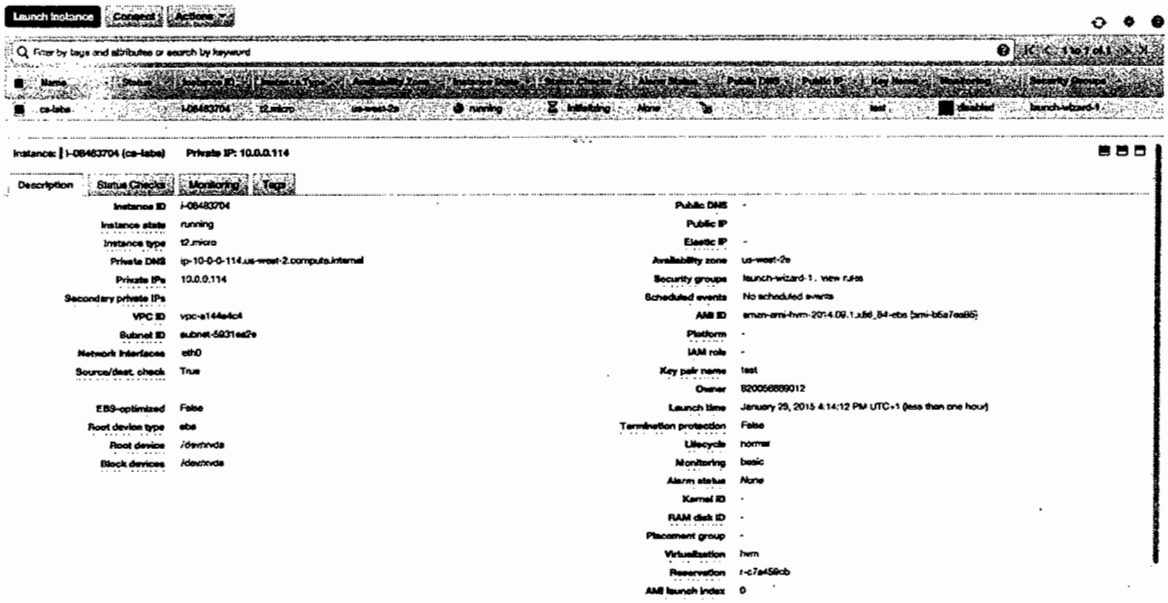
On the Review Instance Launch page, click **Launch**.

In the **Select an existing key pair or create a new key pair** dialog box, select **Create a new key pair**, then type a KeyPair name (e.g., "TestKeys") and download it.

Select the acknowledgment checkbox, and then click **Launch Instances**.

A confirmation page will let you know that your instance is launching. Click **View Instances** to close the confirmation page and return to the console.

On the Instances Screen, you can view the status of your instance. It will take a short time for your instance to be launched. When you launch an instance, its initial state defaults to *pending*. After the instance starts, its Instance State changes to *running*, and it receives a public DNS name.



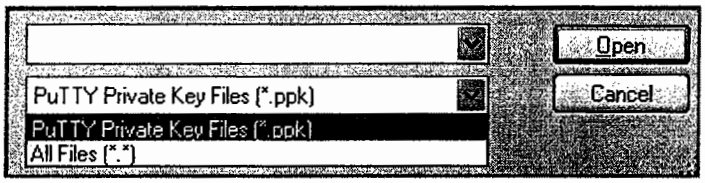
STEP 3: Convert a PEM key to a PPK key

If you are a Windows user, you will probably use **PuTTY** for connecting to the remote instance. PuTTY is a great SSH client, but it does not natively support the PEM key format. Fortunately, PuTTY has a tool called **PuTTYgen**, which can convert keys to the required PPK format.

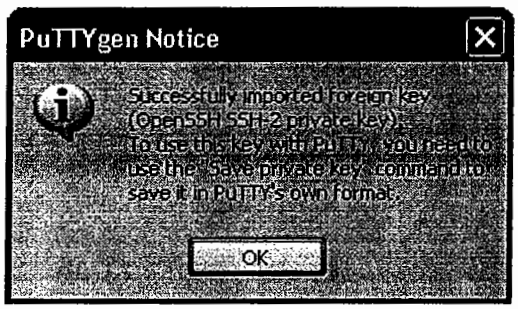
Converting a PEM key is easy and fast:

- ✓ If you do not already have it, download the PuTTYgen executable from its main website: **PuTTYgen**
- ✓ Start PuTTYgen (no installation required).
- ✓ Click **Load** and browse to the location of the private key file that you want to convert (e.g. ec2key.pem). By default, PuTTYgen displays only files with extension .ppk. You'll

need to change that default to display files of all types in order to see your .pem key file.zy



- ✓ Select your .pem key file and click **Open**. PuTTYgen displays the following message.



When you click **OK**, PuTTYgen displays a dialog box with information about the key you loaded, such as the public key and the fingerprint.

- ✓ Click **Save private key** to save the key in PuTTY's format.
- ✓ Do NOT select a passphrase and save your private key somewhere secure.

Now you are ready to use PuTTY for connecting to the previously created instance!

STEP 4: Connect to a remote shell using an SSH connection

In order to manage a remote Linux server, you must employ an **SSH Client**. Secure Shell (SSH) is a cryptographic network protocol for securing data communication. It establishes a secure channel over an insecure network. Common applications include remote command-line login and remote command execution.

Connect using Linux / Mac OS

Linux distributions and Mac OS are shipped with a fully working SSH client that accepts standard PEM Keys.

Starting a remote SSH session is easy:

- ✓ Open your **Terminal** application
- ✓ Write and run the following command: **ssh -i /path/to/your/keypair.pem**

  user@server-ip . **server-ip** is the Public IP of your server, you can find it in the EC2

instance details **user** is the remote system user that will be used for the remote authentication

Amazon Linux AMIs typically use **ec2-user** as username.

Ubuntu AMIs login user is **ubuntu**, Debian AMIs use **admin** instead.

Assuming that you selected the Amazon Linux AMI, your assigned public IP is 123.123.123.123, and your keypair (named "keypair.pem") is stored in /home/youruser/keypair.pem, the right command to run is: **ssh -i /home/youruser/keypair.pem ec2-user@123.123.123.123**

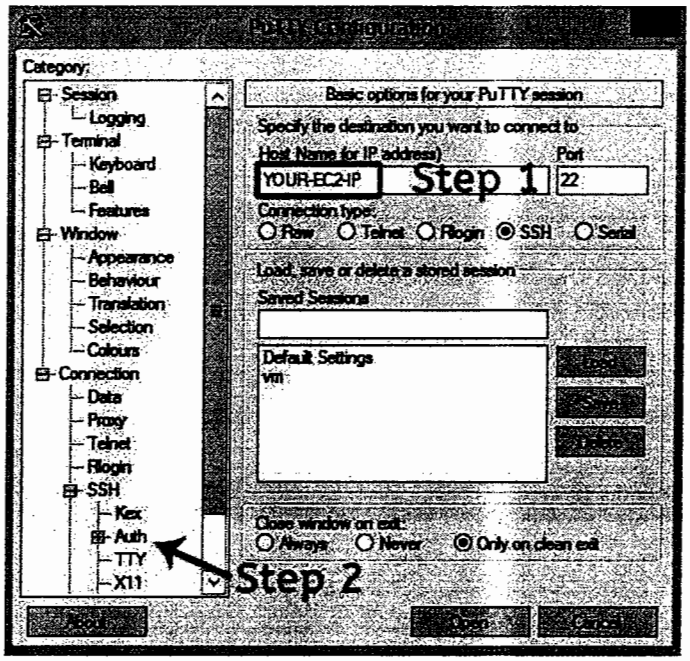
Note: your SSH Client may refuse to start the connection, warning that the key file is unprotected. You should deny the file access to any other system user by changing its permissions. Issue the following command and then try again:

chmod 600 /home/youruser/keypair.pem

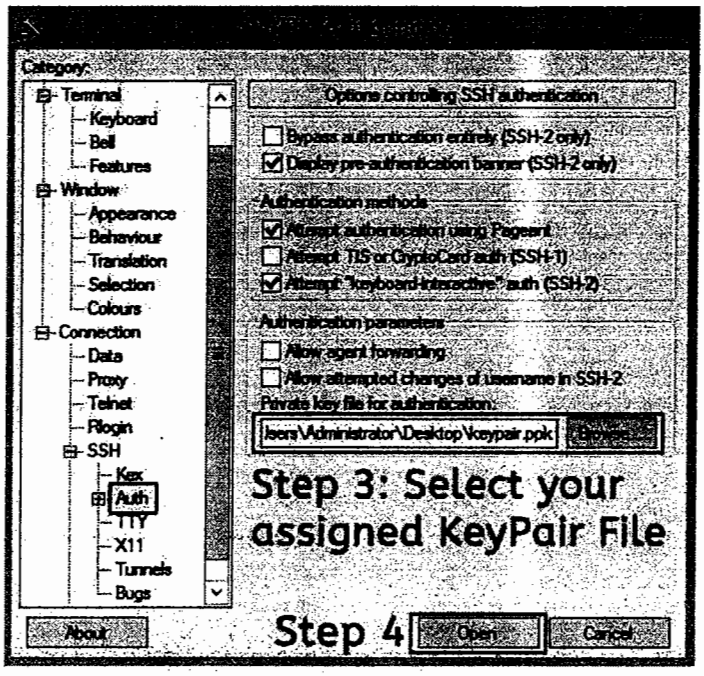
Connect using Windows

Windows has no SSH client, so you must use PuTTY and convert the PEM key to PPK using PuTTYgen. Starting a remote SSH session using PuTTY is easy:

- ✓ Open PuTTY and insert the EC2 instance IP Address in the Host Name field.



Select **Connection > SSH > Auth** section and then select the downloaded Keypair that you previously converted to PPK format.



After some seconds, you will see the authentication form. **Login as ec2-user** and you will see the EC2 server welcome banner.

STEP 5: Get the EC2 instance metadata

Now you are ready to send the first commands to your EC2 linux instance. Let's check the EC2 instance metadata by hitting a specific AWS node only available from within the instance itself.

Instance metadata is data about your instance that you can use to configure or manage the running instance. You can list all instance metadata by issuing the following command:

curl -w "\n" http://169.254.169.254/latest/meta-data/

```
       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2014.09-release-notes/
[ec2-user@ip-172-31-1-167 ~]$ curl -w "\n" http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
```

You can easily check the list of security groups attached to the instance, its ID, the hostname, or the ID of the used AMI. These commands are extremely useful when you want to automate the setup of new instances:

curl -w "\n" http://169.254.169.254/latest/meta-data/security-groups

curl -w "\n" http://169.254.169.254/latest/meta-data/ami-id

curl -w "\n" http://169.254.169.254/latest/meta-data/hostname

curl -w "\n" http://169.254.169.254/latest/meta-data/instance-id

curl -w "\n" http://169.254.169.254/latest/meta-data/instance-type

```
[ec2-user@ip-172-31-1-167 ~]$ curl -w "\n" http://169.254.169.254/latest/meta-data/services

[ec2-user@ip-172-31-1-167 ~]$ curl -w "\n" http://169.254.169.254/latest/meta-data/security-groups
launch-wizard-1
[ec2-user@ip-172-31-1-167 ~]$ curl -w "\n" http://169.254.169.254/latest/meta-data/ami-id
ami-0fc5b0ef
[ec2-user@ip-172-31-1-167 ~]$ curl -w "\n" http://169.254.169.254/latest/meta-data/hostnam
ip-172-31-1-167.us-west-2.compute.internal
[ec2-user@ip-172-31-1-167 ~]$ curl -w "\n" http://169.254.169.254/latest/meta-data/instance-id
i-532c1e5f
[ec2-user@ip-172-31-1-167 ~]$ curl -w "\n" http://169.254.169.254/latest/meta-data/instance-type
t2.micro
```

Finally, you can also get the public key of the attached Keypair using the public-keys metadata:

curl -w "\n" http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key

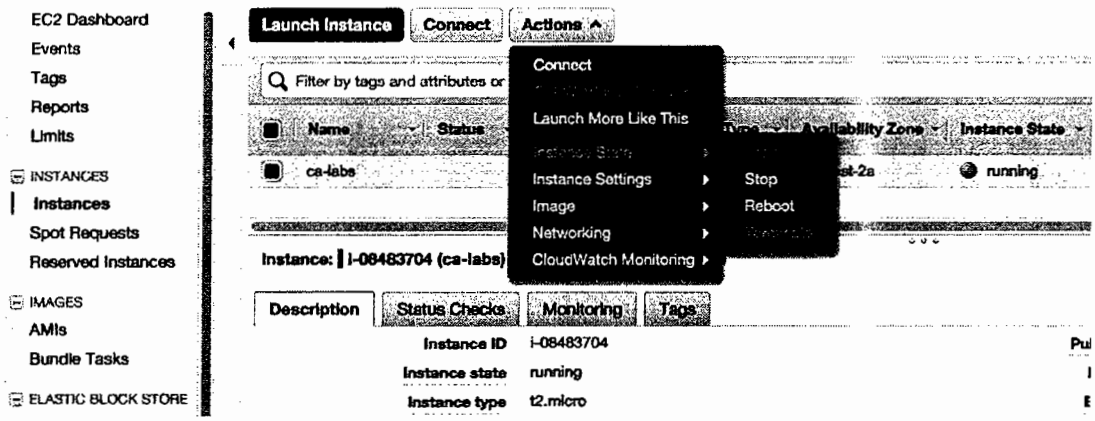
```
[ec2-user@ip-172-31-1-167 ~]$ curl -w "\n" http://169.254.169.254/latest/meta-data/public-keys/0/openssh-key
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQC[illegible]
```

STEP 6: Terminate an EC2 instance

When you've decided that you no longer need an instance, you can terminate it.

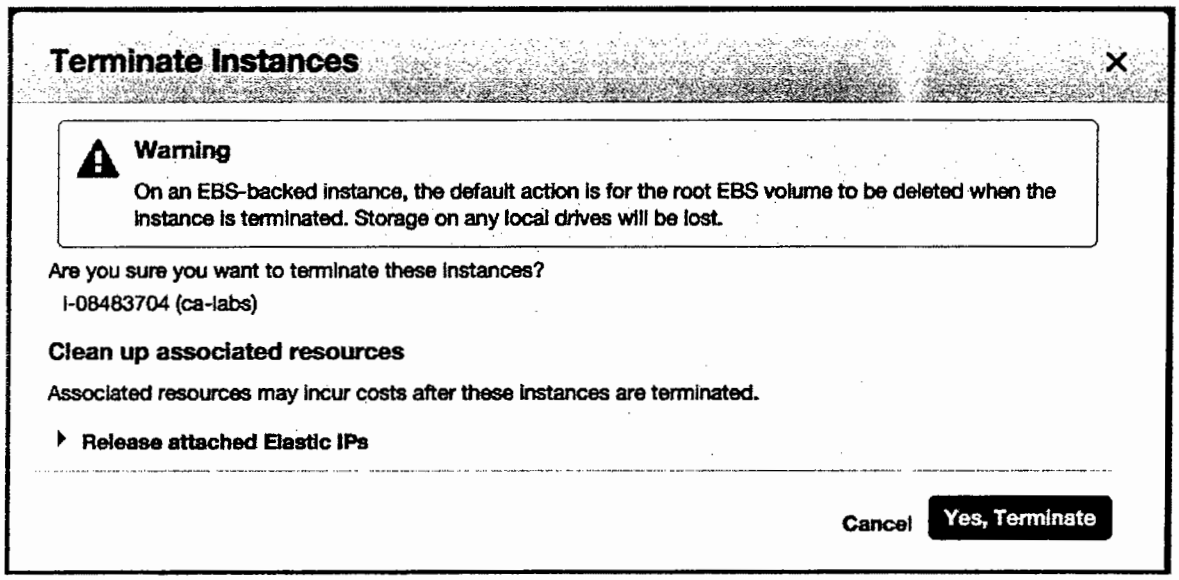
Select the EC2 service from the Management Console dashboard:

In the navigation pane, click **Instances**.



Select the instance ec2instance, click **Actions**, select **Instance State**, and then click **Terminate**.

Click **Yes, Terminate** when prompted for confirmation.



Now your instance is completely destroyed.