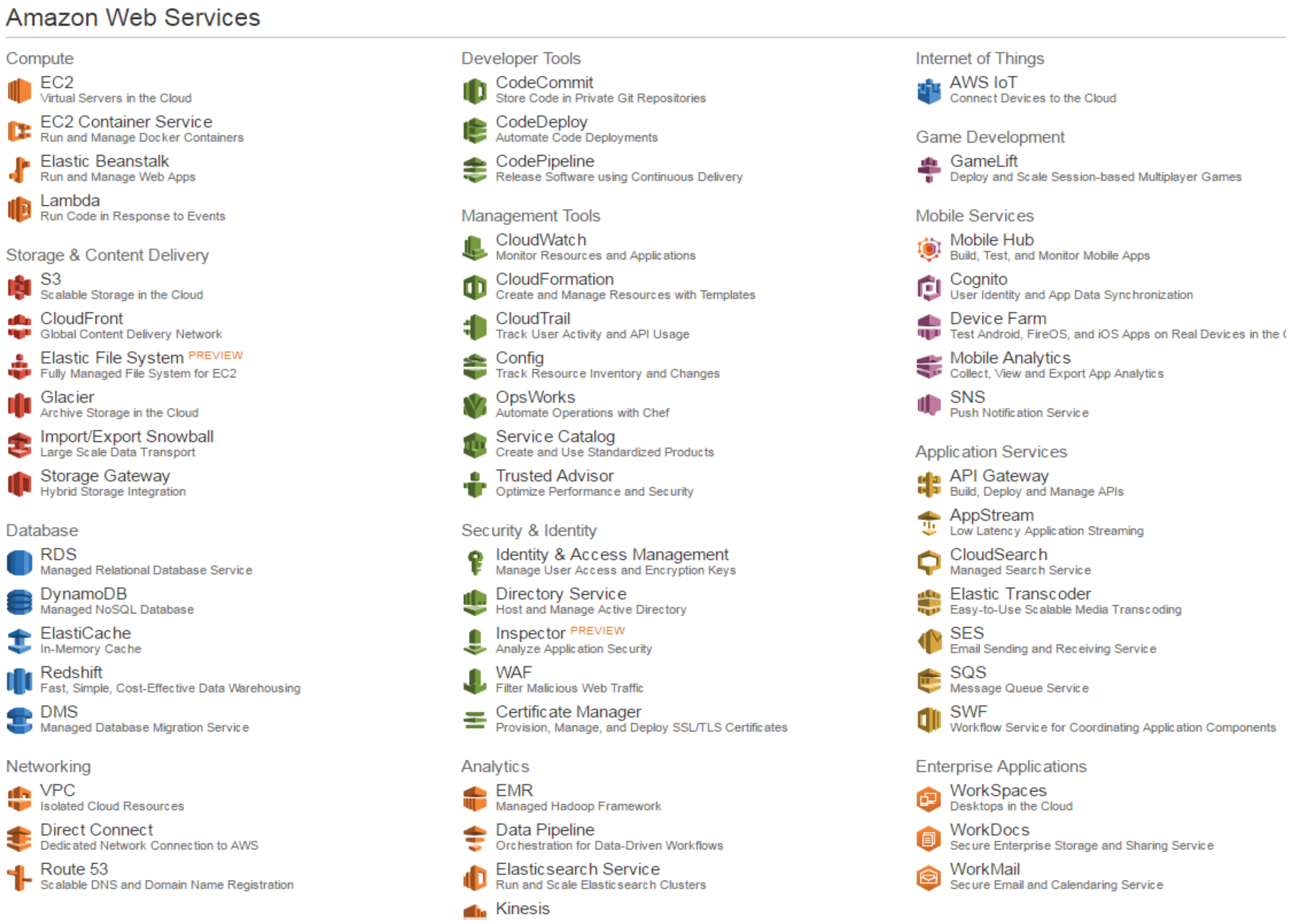
Create Your First EC2 Instance (Linux) by *kvreddi*

Create EC2 Instance (Linux)

STEP#1: Login to Amazon Web Service Console



Amazon Web Services

Compute

EC2
Virtual Servers in the Cloud

EC2 Container Service
Run and Manage Docker Containers

Elastic Beanstalk
Run and Manage Web Apps

Lambda
Run Code in Response to Events

Storage & Content Delivery

S3
Scalable Storage in the Cloud

CloudFront
Global Content Delivery Network

Elastic File System PREVIEW
Fully Managed File System for EC2

Glacier
Archive Storage in the Cloud

Import/Export Snowball
Large Scale Data Transport

Storage Gateway
Hybrid Storage Integration

Database

RDS
Managed Relational Database Service

DynamoDB
Managed NoSQL Database

ElastiCache
In-Memory Cache

Redshift
Fast, Simple, Cost-Effective Data Warehousing

DMS
Managed Database Migration Service

Networking

VPC
Isolated Cloud Resources

Direct Connect
Dedicated Network Connection to AWS

Route 53
Scalable DNS and Domain Name Registration

Developer Tools

CodeCommit
Store Code in Private Git Repositories

CodeDeploy
Automate Code Deployments

CodePipeline
Release Software using Continuous Delivery

Management Tools

CloudWatch
Monitor Resources and Applications

CloudFormation
Create and Manage Resources with Templates

CloudTrail
Track User Activity and API Usage

Config
Track Resource Inventory and Changes

OpsWorks
Automate Operations with Chef

Service Catalog
Create and Use Standardized Products

Trusted Advisor
Optimize Performance and Security

Security & Identity

Identity & Access Management
Manage User Access and Encryption Keys

Directory Service
Host and Manage Active Directory

Inspector PREVIEW
Analyze Application Security

WAF
Filter Malicious Web Traffic

Certificate Manager
Provision, Manage, and Deploy SSL/TLS Certificates

Analytics

EMR
Managed Hadoop Framework

Data Pipeline
Orchestration for Data-Driven Workflows

Elasticsearch Service
Run and Scale Elasticsearch Clusters

Kinesis

Internet of Things

AWS IoT
Connect Devices to the Cloud

Game Development

GameLift
Deploy and Scale Session-based Multiplayer Games

Mobile Services

Mobile Hub
Build, Test, and Monitor Mobile Apps

Cognito
User Identity and App Data Synchronization

Device Farm
Test Android, FireOS, and iOS Apps on Real Devices in the (

Mobile Analytics
Collect, View and Export App Analytics

SNS
Push Notification Service

Application Services

API Gateway
Build, Deploy and Manage APIs

AppStream
Low Latency Application Streaming

CloudSearch
Managed Search Service

Elastic Transcoder
Easy-to-Use Scalable Media Transcoding

SES
Email Sending and Receiving Service

SQS
Message Queue Service

SWF
Workflow Service for Coordinating Application Components

Enterprise Applications

WorkSpaces
Desktops in the Cloud

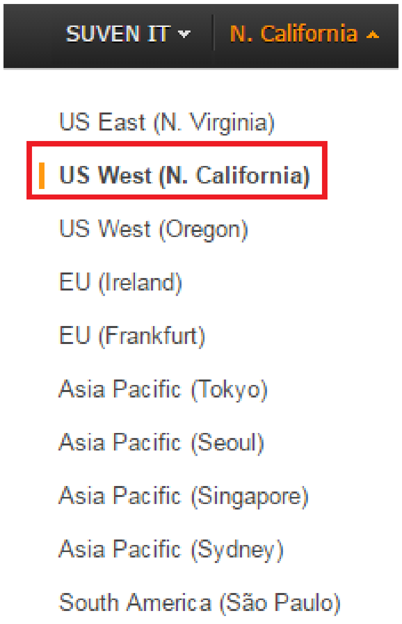
WorkDocs
Secure Enterprise Storage and Sharing Service

WorkMail
Secure Email and Calendaring Service

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

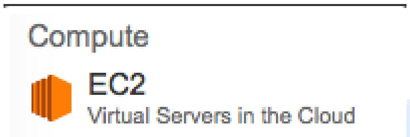
STEP#2: Select the right AWS Region

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 those best suits your business needs to optimize your customer’s experience

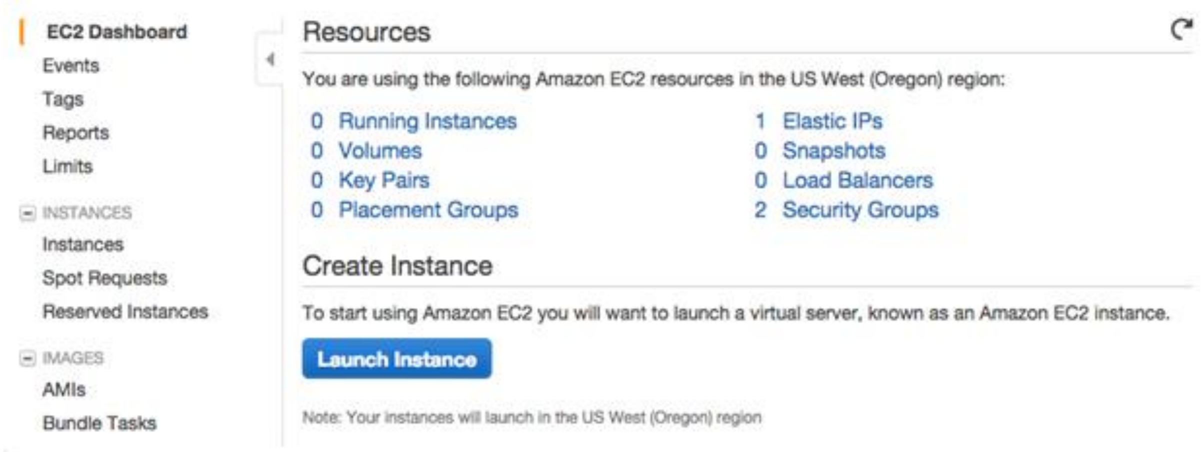


STEP#3: Create a Linux EC2 instance

You can launch an EC2 instance using the EC2 launch wizard.
Select the EC2 service from the Management Console dashboard:



From the dashboard, click Launch Instance.



You are using the following Amazon EC2 resources in the US West (Oregon) region:

0	Running Instances	1	Elastic IPs
0	Volumes	0	Snapshots
0	Key Pairs	0	Load Balancers
0	Placement Groups	2	Security Groups

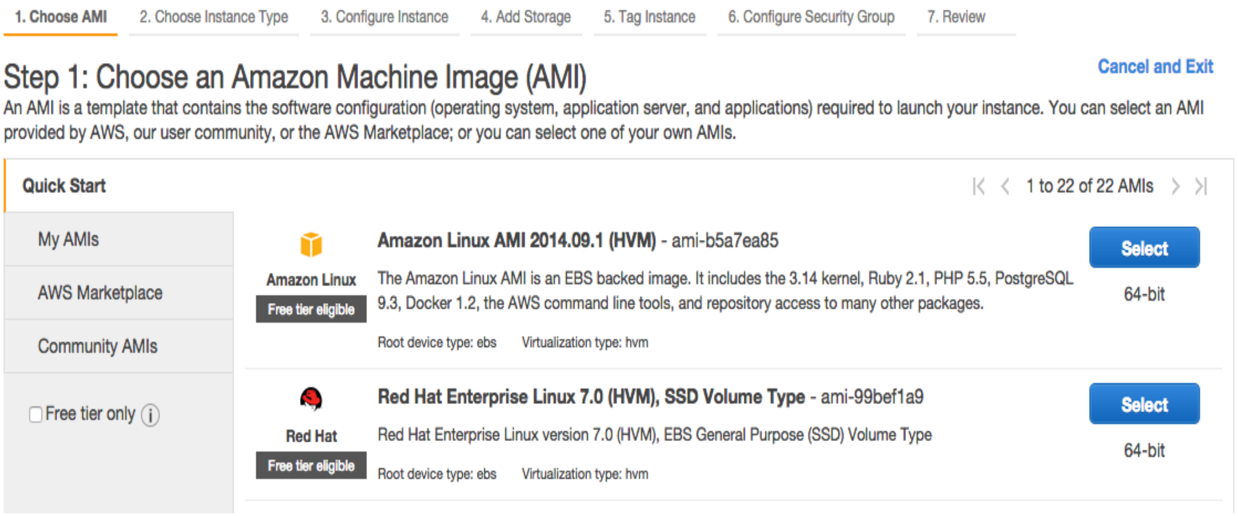
Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US West (Oregon) region

STEP#4: Select the AMI Linux EC2 instance

The Select 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 Redhat Linux Base AMI.



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Amazon Linux — Free tier eligible

Amazon Linux AMI 2014.09.1 (HVM) - ami-b5a7ea85

The Amazon Linux AMI is an EBS backed image. It includes the 3.14 kernel, Ruby 2.1, PHP 5.5, PostgreSQL 9.3, Docker 1.2, the AWS command line tools, and repository access to many other packages.

Root device type: ebs   Virtualization type: hvm

64-bit

Red Hat — Free tier eligible

Red Hat Enterprise Linux 7.0 (HVM), SSD Volume Type - ami-99bef1a9

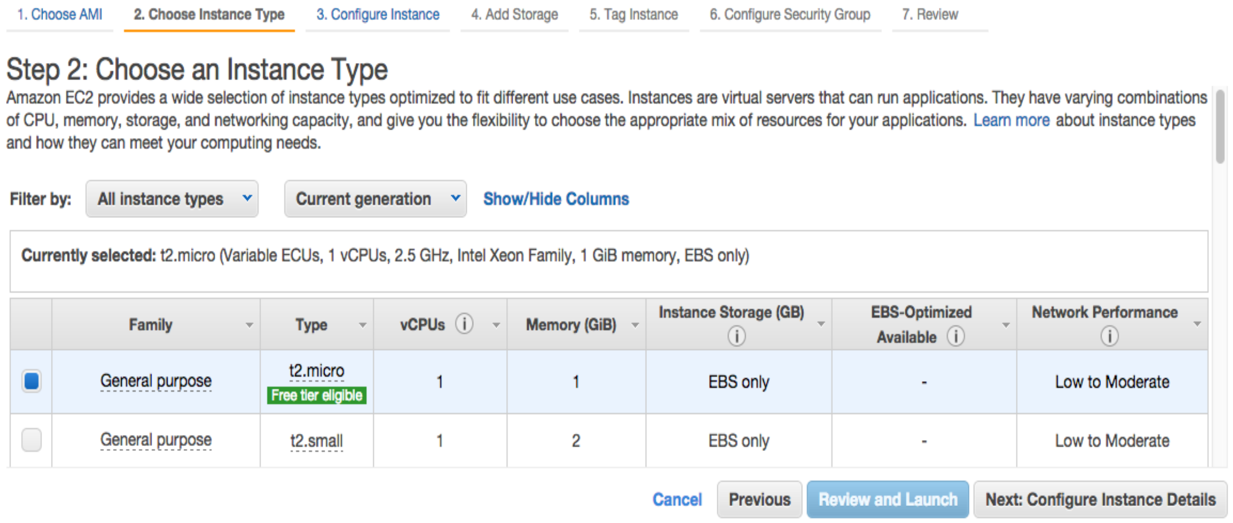
Red Hat Enterprise Linux version 7.0 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs   Virtualization type: hvm

64-bit

STEP#5: Choose an Instance type for Linux EC2 instance

On the **Select an Instance Type** page, do not change any option and click on **Next, Configure Instance Details.**



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.

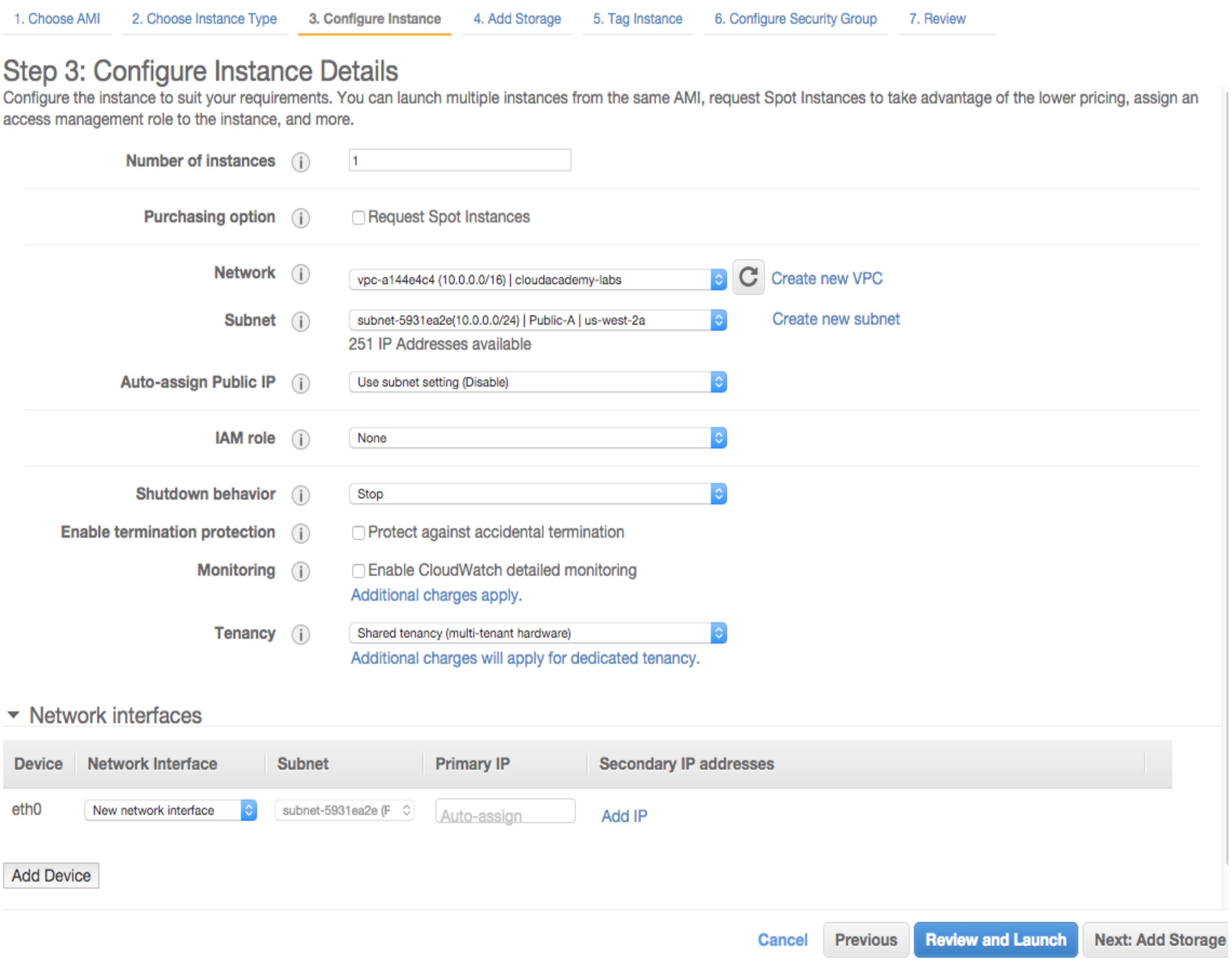
Filter by: All instance types | Current generation

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

STEP#6 -Configure Instance

Check the selected Network (VPC) and Subnet. Change them if needed and then clickNext, Add Storage.



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-a144e4c4 (10.0.0.0/16) | cloudacademy-labs

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

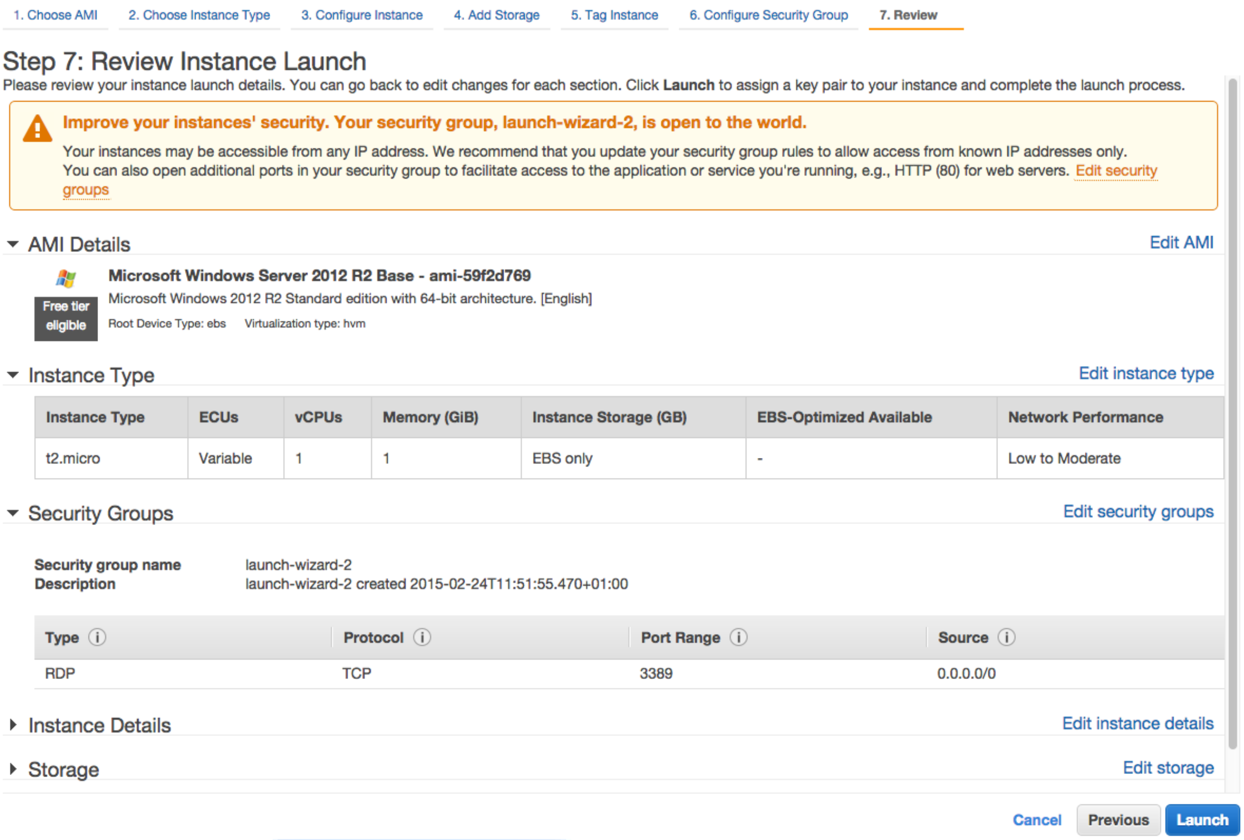
Tenancy: Shared tenancy (multi-tenant hardware)
Additional charges will apply for dedicated tenancy.

Network interfaces

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

STEP#4 -Add Storage

Do not change any option and click "Review and Launch" button.



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

Improve your instances' security. Your security group, launch-wizard-2, is open to the world.

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. Edit security groups

▼ AMI Details — Edit AMI

Microsoft Windows Server 2012 R2 Base - ami-59f2d769
Microsoft Windows 2012 R2 Standard edition with 64-bit architecture. [English]
Free tier eligible
Root Device Type: ebs   Virtualization type: hvm

▼ Instance Type — Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

▼ Security Groups — Edit security groups

Security group name launch-wizard-2
Description launch-wizard-2 created 2015-02-24T11:51:55.470+01:00

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
RDP	TCP	3389	0.0.0.0/0

▶ Instance Details — Edit instance details

▶ Storage — Edit storage

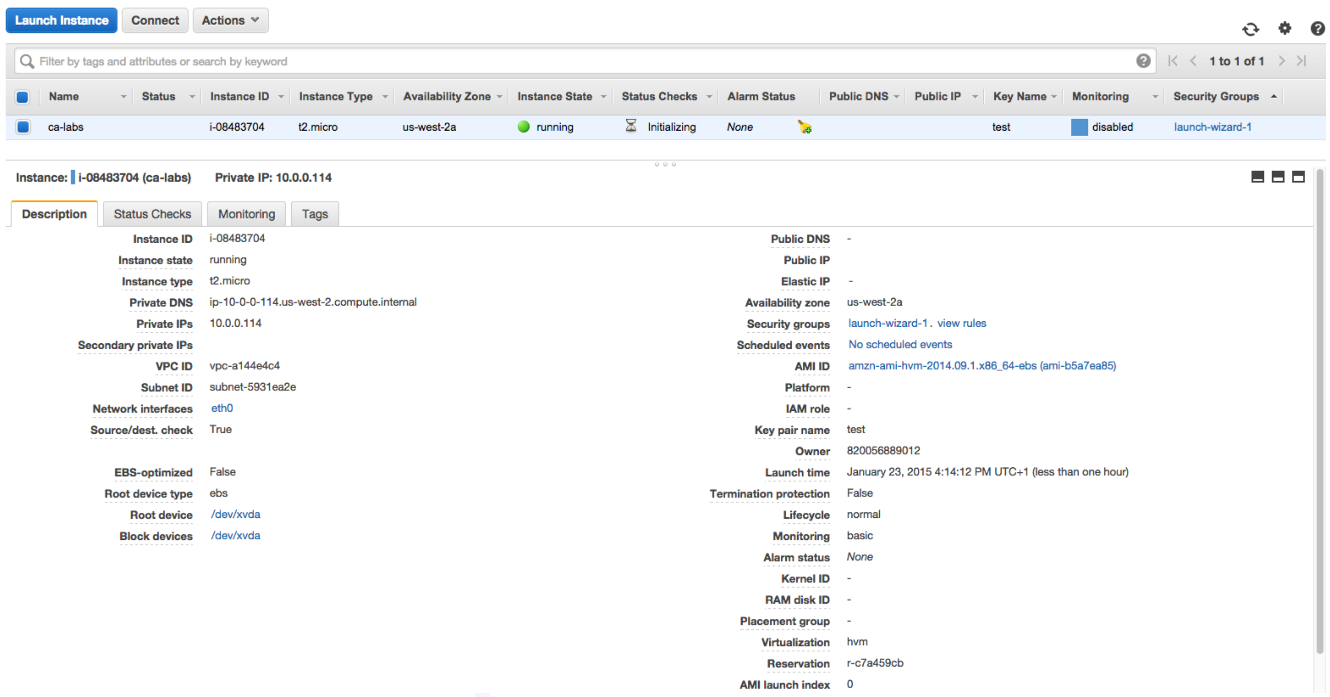
On the Review Instance Launch page, click **Launch**.

In the **Select an existing key pair or create a new key pair** dialog box, select **Create a new key pair**, then choose a KeyPair name and download it.

Select the acknowledgment check box, 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 is **pending**. After the instance starts, its state changes to **running**, and it receives a public DNS name.

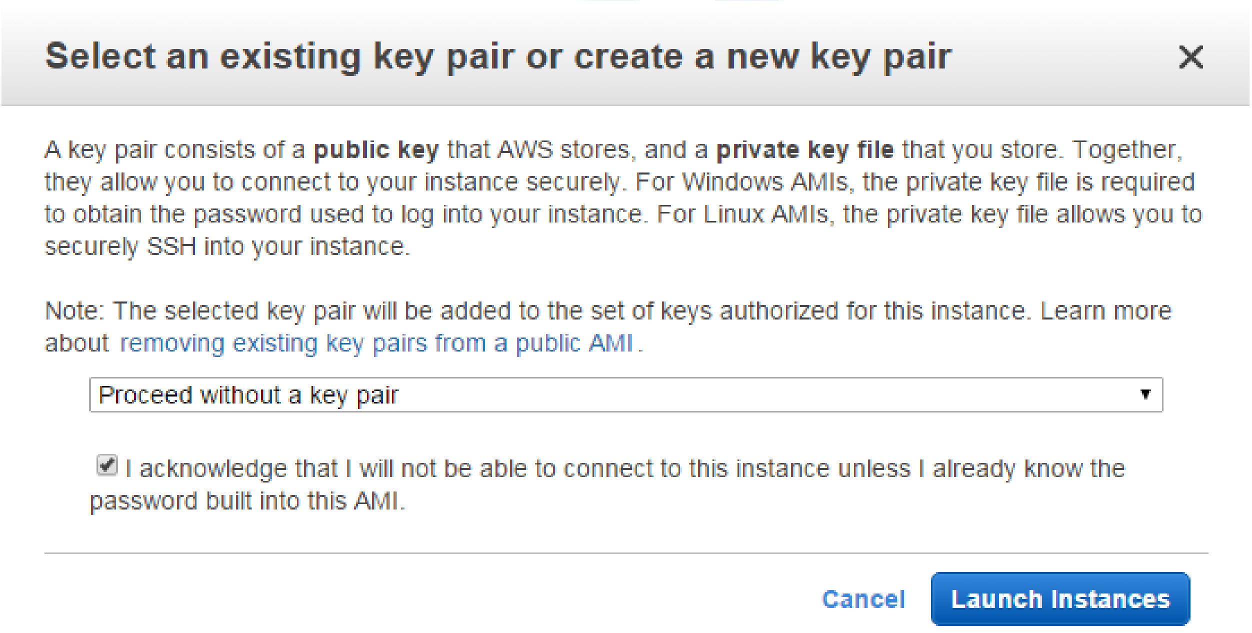


Name	Status	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring	Security Groups
ca-labs		i-08483704	t2.micro	us-west-2a	● running	Initializing	None			test	disabled	launch-wizard-1

Instance: i-08483704 (ca-labs) Private IP: 10.0.0.114

Instance ID	i-08483704	Public DNS	-
Instance state	running	Public IP	
Instance type	t2.micro	Elastic IP	-
Private DNS	ip-10-0-0-114.us-west-2.compute.internal	Availability zone	us-west-2a
Private IPs	10.0.0.114	Security groups	launch-wizard-1 . view rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-a144e4c4	AMI ID	amzn-ami-hvm-2014.09.1.x86_64-ebs (ami-b5a7ea85)
Subnet ID	subnet-5931ea2e	Platform	-
Network interfaces	eth0	IAM role	-
Source/dest. check	True	Key pair name	test
		Owner	820056889012
EBS-optimized	False	Launch time	January 23, 2015 4:14:12 PM UTC+1 (less than one hour)
Root device type	ebs	Termination protection	False
Root device	/dev/xvda	Lifecycle	normal
Block devices	/dev/xvda	Monitoring	basic
		Alarm status	None
		Kernel ID	-
		RAM disk ID	-
		Placement group	-
		Virtualization	hvm
		Reservation	r-c7a459cb
		AMI launch index	0

STEP#7: Next, it will ask for Key.



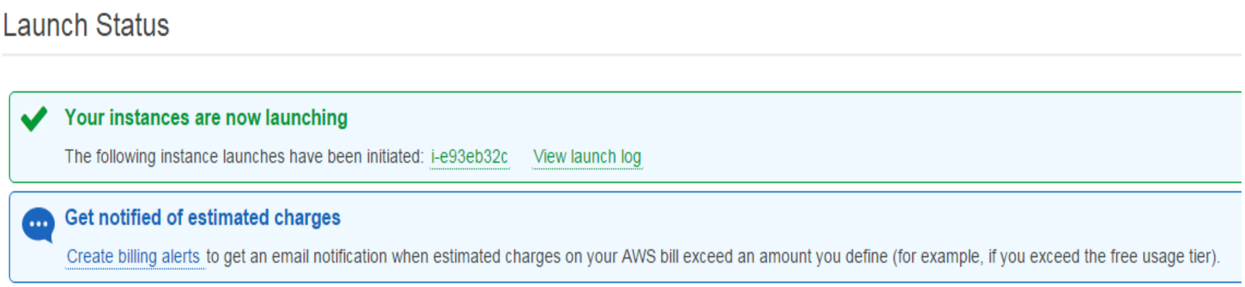
Select an existing key pair or create a new key pair

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 .

Proceed without a key pair ▼

☑ I acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI.



Launch Status

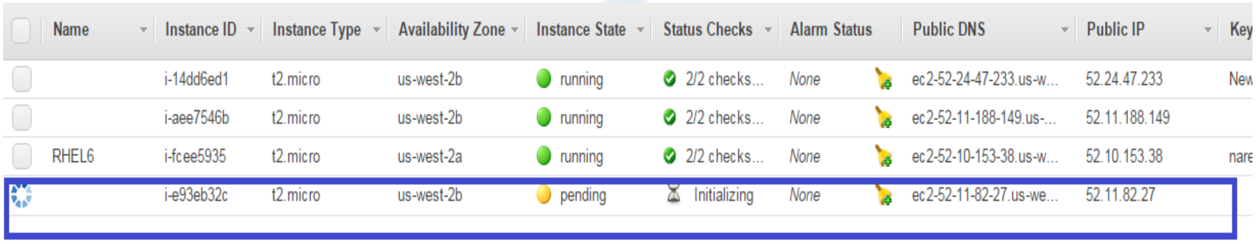
✓ **Your instances are now launching**

The following instance launches have been initiated: i-e93eb32c View launch log

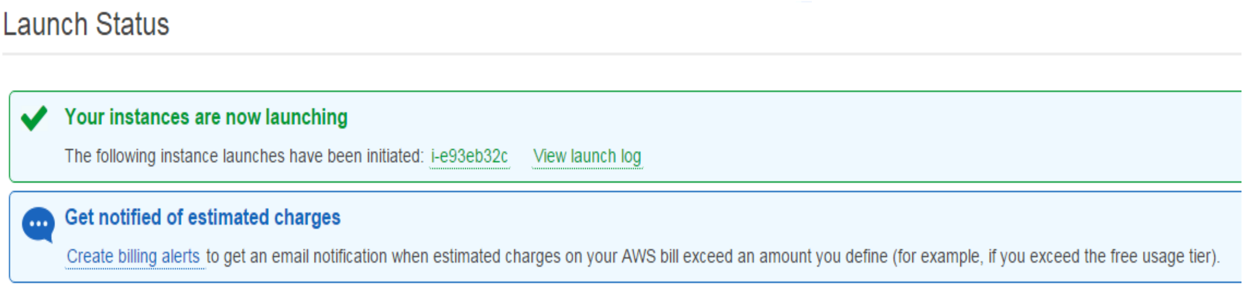
Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

Initializing the server



□	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status		Public DNS	Public IP	Key
□		i-14dd6ed1	t2.micro	us-west-2b	● running	✓ 2/2 checks...	None		ec2-52-24-47-233.us-w...	52.24.47.233	New
□		i-aee7546b	t2.micro	us-west-2b	● running	✓ 2/2 checks...	None		ec2-52-11-188-149.us-...	52.11.188.149	
□	RHEL6	i-fcee5935	t2.micro	us-west-2a	● running	✓ 2/2 checks...	None		ec2-52-10-153-38.us-w...	52.10.153.38	nare
		i-e93eb32c	t2.micro	us-west-2b	● pending	⌛ Initializing	None		ec2-52-11-82-27.us-we...	52.11.82.27	



Launch Status

✓ **Your instances are now launching**

The following instance launches have been initiated: i-e93eb32c View launch log

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

Finally, the server status becomes running.

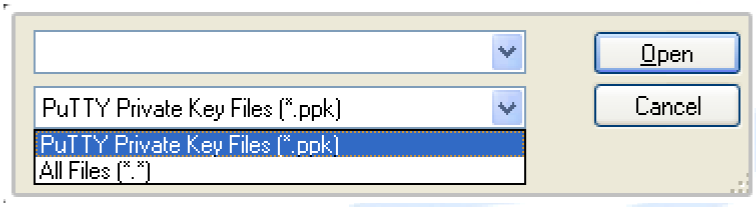
Connect to a remote shell using a SSH connection

Convert a PEM key to a PPK key to access the server.

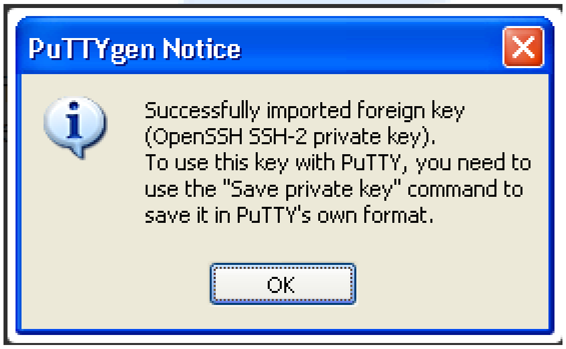
If you are a Windows user, you are probably using **Putty** for connecting to the remote instance. Putty is a great SSH client, but it does not natively support the PEM key format. Fortunately, PuTTY has a tool called **PuTTYgen**, which can convert keys to the required PuTTY format.

Converting a PEM key is an easy and fast operation:

1. Download the PuTTYgen executable from its main website: **PuTTYgen**
2. Start PuTTYgen (no installation required).
3. 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 to display files of all types in order to see your .pem key file.


Open
Cancel
PuTTY Private Key Files (*.ppk)
PuTTY Private Key Files (*.ppk)
All Files (*.*)

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


PuTTYgen Notice

Successfully imported foreign key (OpenSSH SSH-2 private key).
To use this key with PuTTY, you need to use the "Save private key" command to save it in PuTTY's own format.

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

5. Click **Save private key** to save the key in PuTTY's format.
6. 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!

In order to manage a remote Linux server, you need to use an **SSH Client**. Secure SHell (SSH) is a cryptographic network protocol for securing data communication. It establishes a secure channel over an insecure network and common applications include remote command-line login, 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 quite 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.

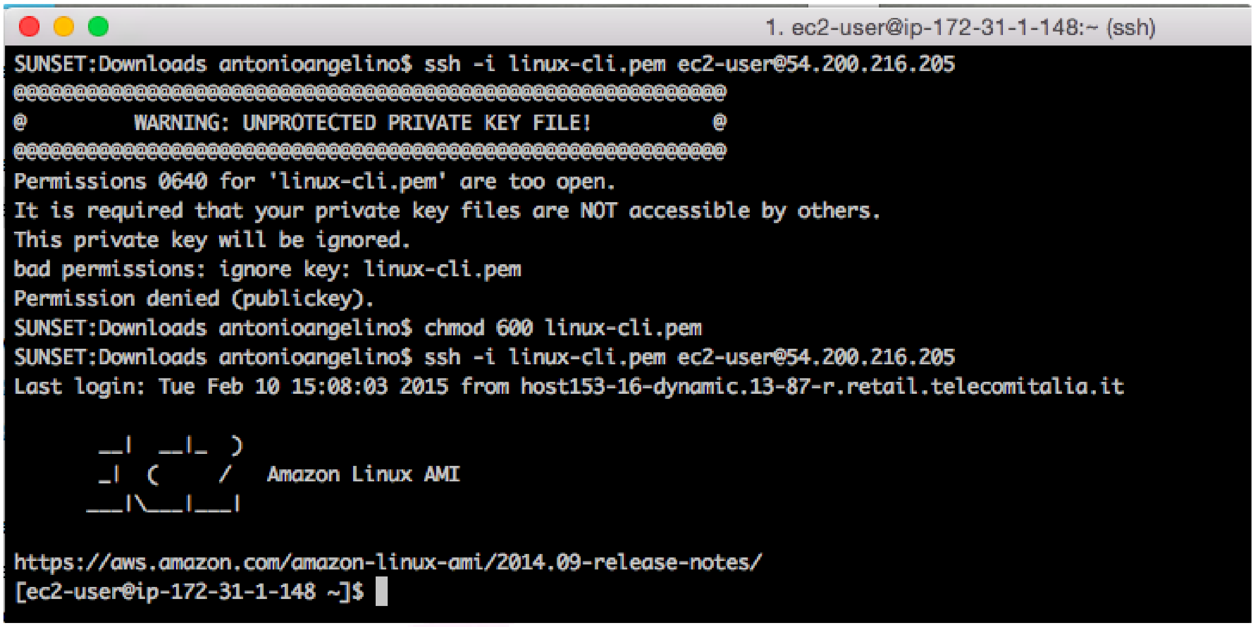
Amazon Linux AMIs usually use `ec2-user` as username.

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 need to 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
```



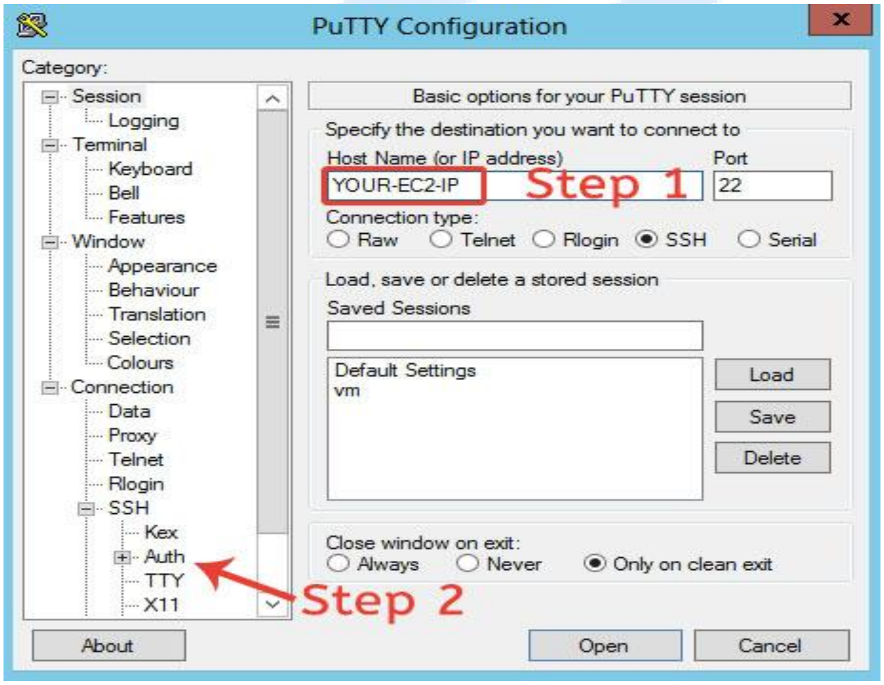
1. ec2-user@ip-172-31-1-148:~ (ssh)

```
SUNSET:Downloads antonioangelino$ ssh -i linux-cli.pem ec2-user@54.200.216.205
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0640 for 'linux-cli.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
bad permissions: ignore key: linux-cli.pem
Permission denied (publickey).
SUNSET:Downloads antonioangelino$ chmod 600 linux-cli.pem
SUNSET:Downloads antonioangelino$ ssh -i linux-cli.pem ec2-user@54.200.216.205
Last login: Tue Feb 10 15:08:03 2015 from host153-16-dynamic.13-87-r.retail.telecomitalia.it

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

https://aws.amazon.com/amazon-linux-ami/2014.09-release-notes/
[ec2-user@ip-172-31-1-148 ~]$ █
```

Connect using Windows

Windows hasn't any SSH client, so you need to 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.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): YOUR-EC2-IP   Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

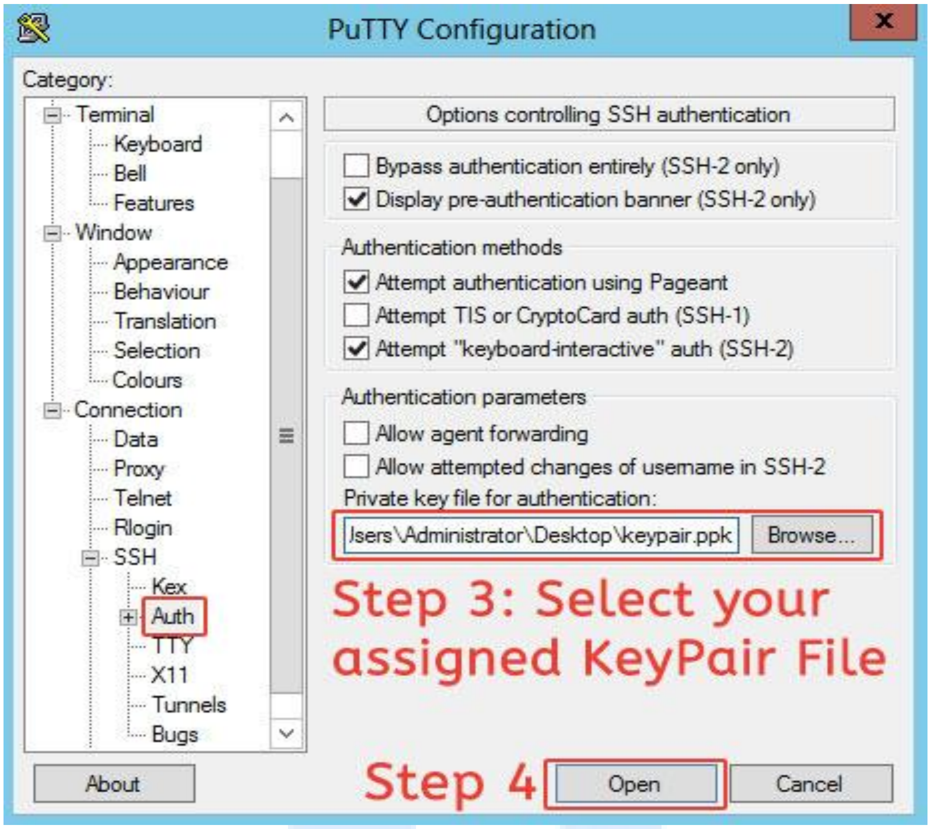
Default Settings
vm

Close window on exit: ○ Always ○ Never ● Only on clean exit

Step 1

Step 2

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



Options controlling SSH authentication

☐ Bypass authentication entirely (SSH-2 only)
☑ Display pre-authentication banner (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant
☐ Attempt TIS or CryptoCard auth (SSH-1)
☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

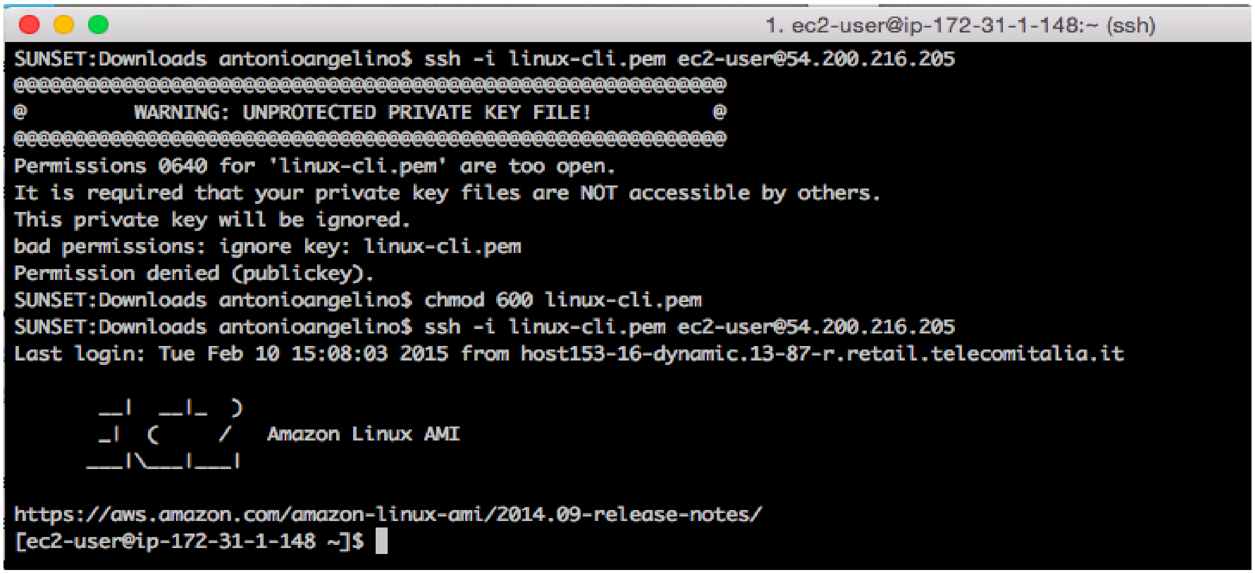
☐ Allow agent forwarding
☐ Allow attempted changes of username in SSH-2

Private key file for authentication: Jsers\Administrator\Desktop\keypair.ppk

Step 3: Select your assigned KeyPair File

Step 4

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



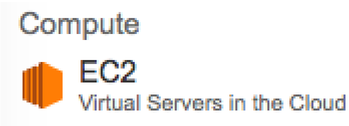
1. ec2-user@ip-172-31-1-148:~ (ssh)

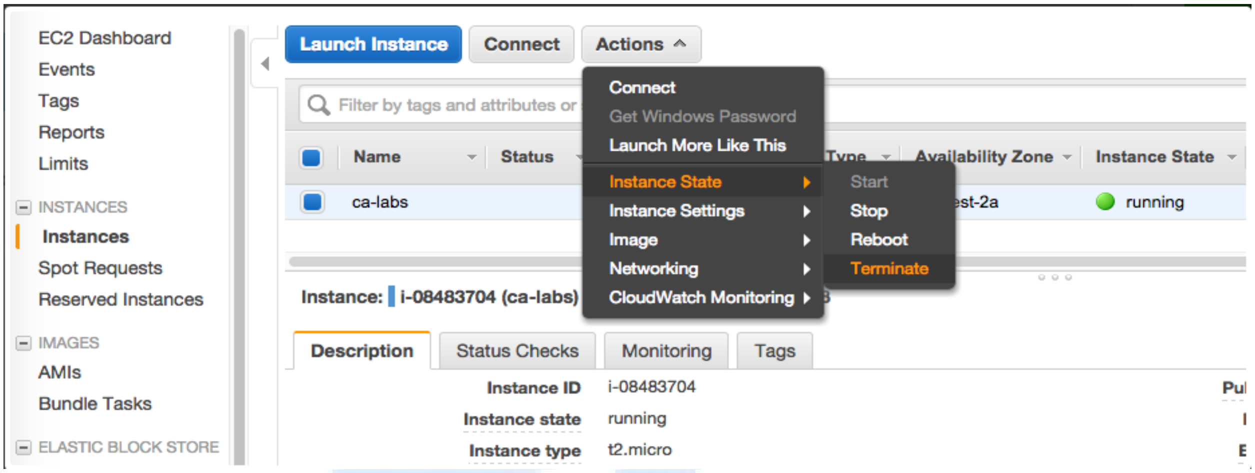
```
SUNSET:Downloads antonioangelino$ ssh -i linux-cli.pem ec2-user@54.200.216.205
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0640 for 'linux-cli.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
bad permissions: ignore key: linux-cli.pem
Permission denied (publickey).
SUNSET:Downloads antonioangelino$ chmod 600 linux-cli.pem
SUNSET:Downloads antonioangelino$ ssh -i linux-cli.pem ec2-user@54.200.216.205
Last login: Tue Feb 10 15:08:03 2015 from host153-16-dynamic.13-87-r.retail.telecomitalia.it

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

https://aws.amazon.com/amazon-linux-ami/2014.09-release-notes/
[ec2-user@ip-172-31-1-148 ~]$ 
```

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:





Instance: i-08483704 (ca-labs)

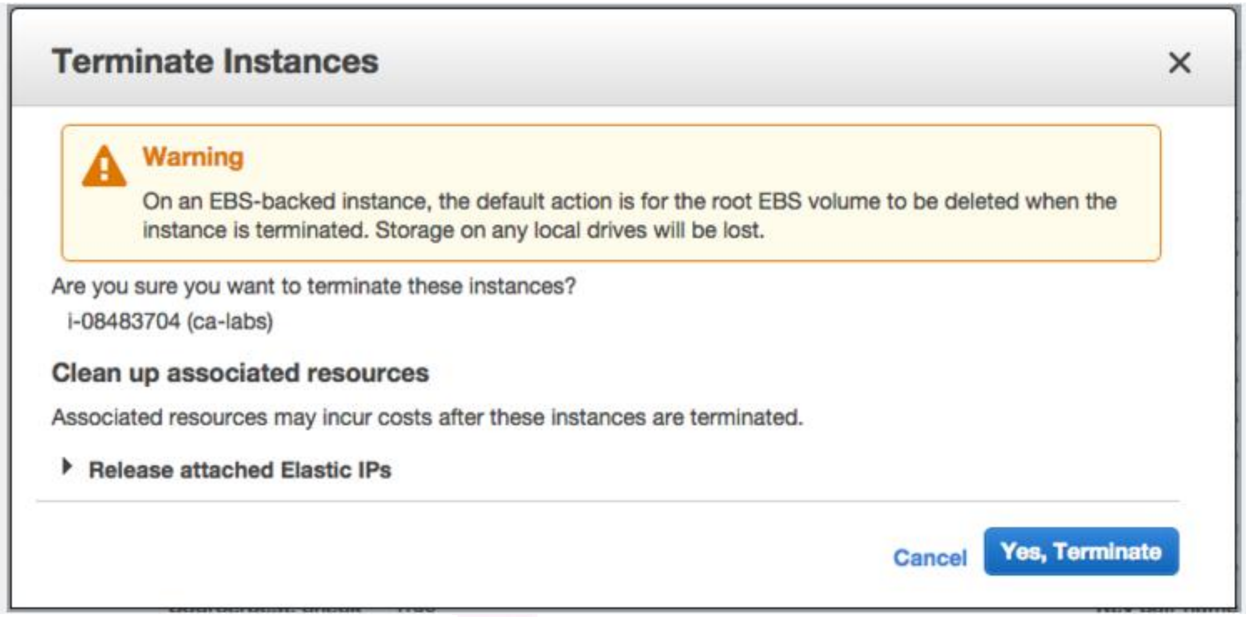
Instance ID i-08483704

Instance state running

Instance type t2.micro

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

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



Terminate Instances

Warning

On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Are you sure you want to terminate these instances?

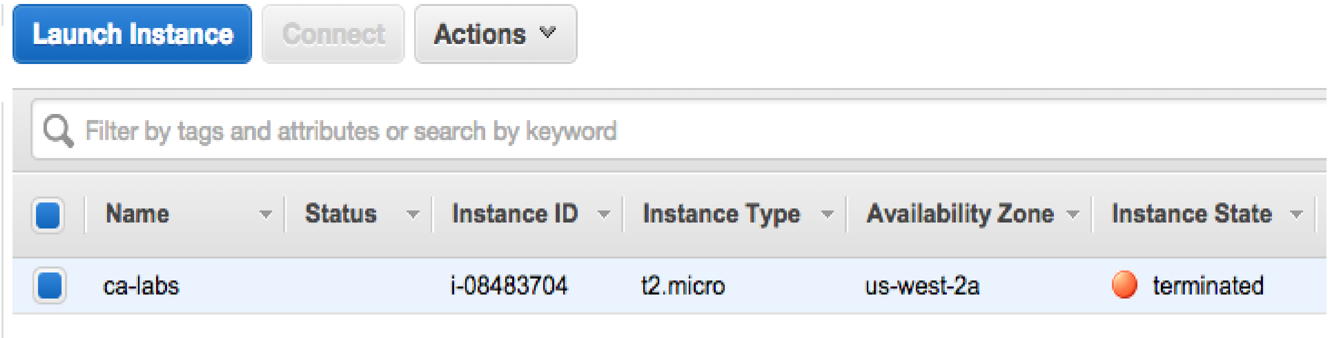
i-08483704 (ca-labs)

Clean up associated resources

Associated resources may incur costs after these instances are terminated.

Now your instance is completely destroyed.

Check the status of the server.



	Name	Status	Instance ID	Instance Type	Availability Zone	Instance State
■	ca-labs		i-08483704	t2.micro	us-west-2a	● terminated

Note: We can’t restive the server after terminating

SUVEN IT

About us

***SUVEN IT** established in 01-Jan--2010 by **Mr. kvreddi** having 20 years teaching and 17 years of real time work experience across USA & India, We are recognized as a leader in all IT training Courses to supply quality IT Professionals to Industry. SUVEN IT committed to provide high quality service with elevated level of student's satisfaction and provides the high end industry training and real time knowledge to students.*

We trained and placed 3000+ Students in top MNC's within 6 Years (Most of them are selected in first interview)

By Kvreddi