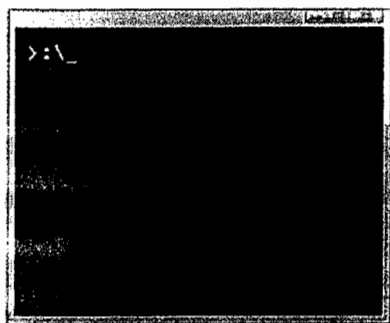


Lab 2

FIRST STEPS INTO THE LINUX CONSOLE

STEP 1: Introduction to the Linux Command Line Interface (CLI)



GNU/Linux is a modern Operating System and it's widely used on servers. Server-like Linux distributions are usually setup without any Graphical User Interface (GUI) for saving computational power and because you can do almost anything using the command-line interface (CLI). In the old days, CLI was the only user interface available on a Unix-like system such as Linux.

The **command-line interface** is a tool into which you can type text commands to perform specific tasks. Since you can directly control the computer by typing, many tasks can be performed more quickly, and some **tasks can be automated** with special commands that loop through and perform the same action on many files—saving you a lot of time.

The application or user interface that accepts your typed responses and displays the data on the screen is called a **shell**, and there are many different varieties that you can choose from (ksh, tcsh, zsh...), but the most common these days is the **Bash** shell, which is the default on Linux and Mac systems. Bash stands for Bourne Again SHell, and it is an enhanced version of the original Unix shell program, **sh**, written by Steve Bourne.

You can use a local or remote shell using a program called a **terminal emulator**. This is a program that opens a window and lets you interact with the shell.

There are a bunch of different terminal emulators you can use:

PuTTY is a terminal client suitable for Windows users.

Terminal is embedded in all Mac OS versions.

iTerm is an enhanced terminal application for MacOS.

The next lab steps will show you how to connect to a remote EC2 instance using the SSH protocol and get familiar with the basic shell commands.

Go to the next step and start by creating a basic EC2 Linux instance.

STEP 2: 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 screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with 'Services' and a search bar. The main content area is titled 'Amazon Web Services' and is divided into several columns of service categories. On the right, there are sections for 'Additional Resources' (Getting Started, AWS Console Mobile App, AWS Marketplace) and 'Service Health' (All services operating normally, updated Nov 20 2014 12:57:00 GMT-0800). At the bottom right, there's a 'Set Start Page' section with a 'Console Home' button.

Amazon Web Services

- Compute**
 - EC2: Virtual Servers in the Cloud
 - Lambda PREVIEW: Run Code in Response to Events
- Storage & Content Delivery**
 - S3: Scalable Storage in the Cloud
 - Storage Gateway: Integrates On-Premises IT Environments with Cloud Storage
 - Glacier: Archive Storage in the Cloud
 - CloudFront: Global Content Delivery Network
- Database**
 - RDS: MySQL, PostgreSQL, Oracle, SQL Server, and Amazon Aurora
 - DynamoDB: Predictable and Scalable NoSQL Data Store
 - ElastiCache: In-Memory Cache
 - Redshift: Managed Petabyte-Scale Data Warehouse Service
- Networking**
 - VPC: Isolated Cloud Resources
 - Direct Connect: Dedicated Network Connection to AWS
 - Route 53: Scalable DNS and Domain Name Registration
- Administration & Security**
 - Directory Service: Managed Directories in the Cloud
 - Identity & Access Management: Access Control and Key Management
 - Trusted Advisor: AWS Cloud Optimization Expert
 - CloudTrail: User Activity and Change Tracking
 - Config PREVIEW: Resource Configurations and Inventory
 - CloudWatch: Resource and Application Monitoring
- Deployment & Management**
 - Elastic Beanstalk: AWS Application Container
 - OpsWorks: DevOps Application Management Service
 - CloudFormation: Templated AWS Resource Creation
 - CodeDeploy: Automated Deployments
- Analytics**
 - EMR: Managed Hadoop Framework
 - Kinesis: Real-time Processing of Streaming Big Data
 - Data Pipeline: Orchestration for Data-Driven Workflows
- Application Services**
 - SQS: Message Queue Service
 - SWF: Workflow Service for Coordinating Application Components
 - AppStream: Low Latency Application Streaming
 - Elastic Transcoder: Easy-to-use Scalable Media Transcoding
 - SES: Email Sending Service
 - CloudSearch: Managed Search Service
- Mobile Services**
 - Cognito: User Identity and App Data Synchronization
 - Mobile Analytics: Understand App Usage Data at Scale
 - SNS: Push Notification Service
- Enterprise Applications**
 - WorkSpaces: Desktops in the Cloud
 - Zocalo: Secure Enterprise Storage and Sharing Service

Additional Resources

- Getting Started**
See our documentation to get started and learn more about how to use our services.
- AWS Console Mobile App**
View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.
- AWS Marketplace**
Find and buy software, launch with 1-Click and pay by the hour.

Service Health

- All services operating normally.
- Updated: Nov 20 2014 12:57:00 GMT-0800
- Service Health Dashboard

Set Start Page

Console Home

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**.



Account:

User Name:

Password:

☐ I have an MFA Token ([more info](#))

Sign In

[Sign-in using root account credentials](#)

[Terms of Use](#) [Privacy Policy](#)

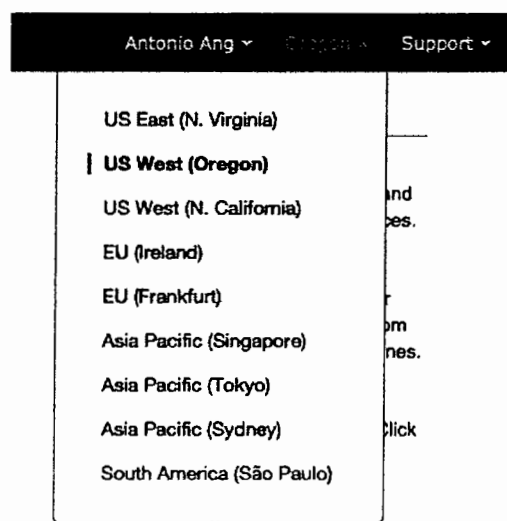
© 1996-2014, Amazon Web Services, Inc. or its affiliates.

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 3: Create an EC2 instance

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

Select the EC2 service from the Management Console dashboard:

Compute



EC2

Virtual Servers in the Cloud

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

EC2 Dashboard

Events
Tags
Reports
Limits

☒ INSTANCES

Instances
Spot Requests
Reserved Instances

☒ IMAGES

AMIs
Bundle Tasks

Resources

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.

Launch Instance

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

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**.

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

Step 1: Choose an Amazon Machine Image (AMI)

Cancel and Exit

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.

Quick Start

My AMIs

AWS Marketplace

Community AMIs

☐ Free tier only ⓘ

Amazon Linux AMI 2014.09.1 (HVM) - ami-b5a7ae85

Free tier eligible

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

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

Free tier eligible

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

Root device type: ebs Virtualization type: hvm

64-bit

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 (GiB) | EBS-Optimized Available | Network Performance |
|-------------------------------------|-----------------|--------------------------------|-------|--------------|------------------------|-------------------------|---------------------|
| <input checked="" type="checkbox"/> | General purpose | t2.micro Free tier eligible | 1 | 1 | EBS only | - | Low to Moderate |
| <input type="checkbox"/> | 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 1

Purchasing option ① ☐ Request Spot Instances

Network ① vpc-a144e40f (10.0.0.0/16) **Create new VPC**

Subnet ① subnet-6931ea2e (10.0.0.0/24) | Public-A | us-west-2a **Create new subnet**
251 IP Addresses available

Auto-assign Public IP ① Use subnet setting (Disable)

IAM role ① None

Shutdown behavior ① Stop

Enable termination protection ① ☐ Protect against accidental termination

Monitoring ① ☐ Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy ① 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-6931ea2e | 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.

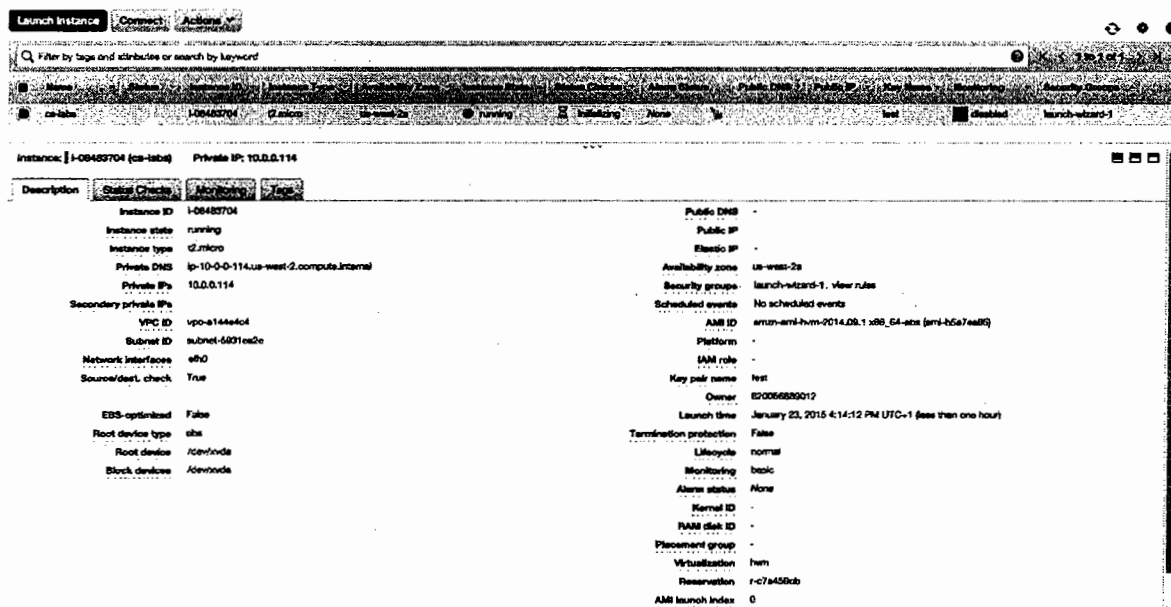
On the Review Instance Launch page, click **Launch**.

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 4: Convert a PEM key to a PPK key

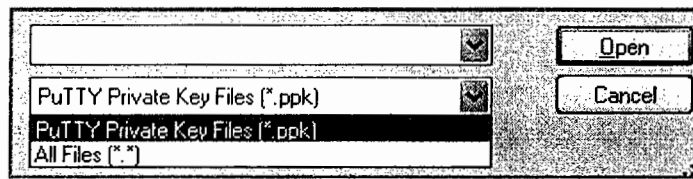
If you are a Windows user, you will probably use **PuTTY** for connecting to the remote instance. PuTTY is a great SSH client, but it does not natively support the PEM key format. Fortunately, PuTTY has a tool called **PuTTYgen**, which can convert keys to the required PPK format.

Converting a PEM key is easy and fast:

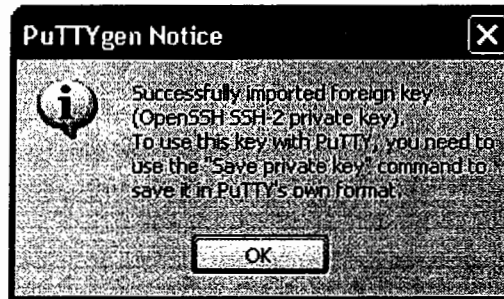
If you do not already have it, download the PuTTYgen executable from its main website: **PuTTYgen**

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.



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 5: 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

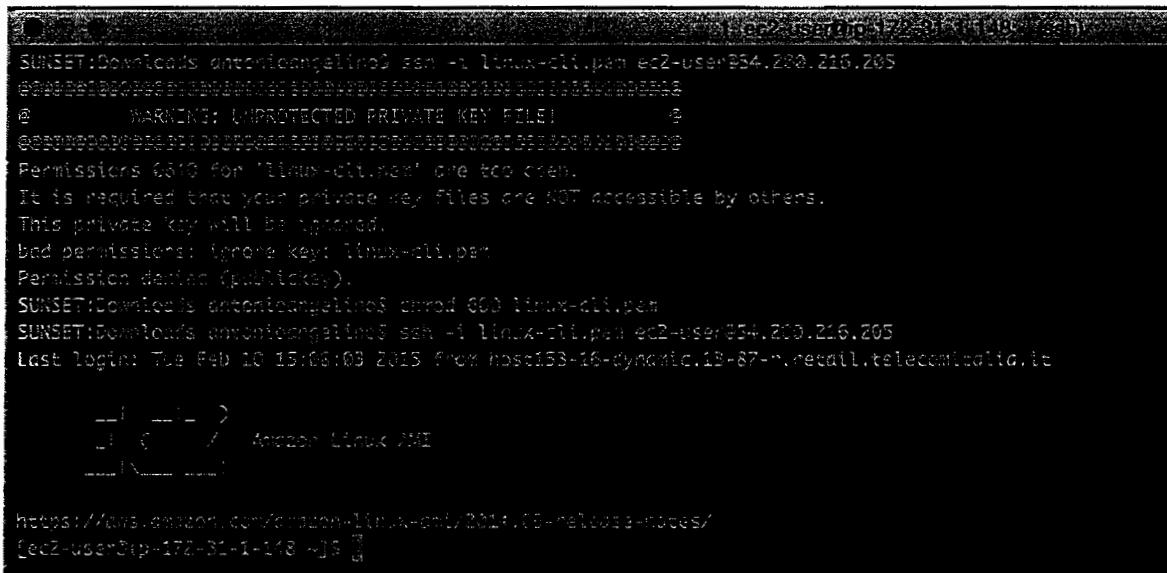
Amazon Linux AMIs typically use `ec2-user` as username.

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`



```
SUNSET:Downloads antonioangelino$ ssh -i linux-ali.pem ec2-user@54.230.216.205
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@           WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0640 for 'linux-ali.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-ali.pem
Permission denied (publickey).
SUNSET:Downloads antonioangelino$ chmod 600 linux-ali.pem
SUNSET:Downloads antonioangelino$ ssh -i linux-ali.pem ec2-user@54.230.216.205
Last login: Tue Feb 10 15:08:03 2015 from host153-16-dynamic.19-87-n.netcell.td.td.it

  ____  _
 / ___|| | | |
| |___| |_| |
 \___|_____|_|

Amazon Linux AMI

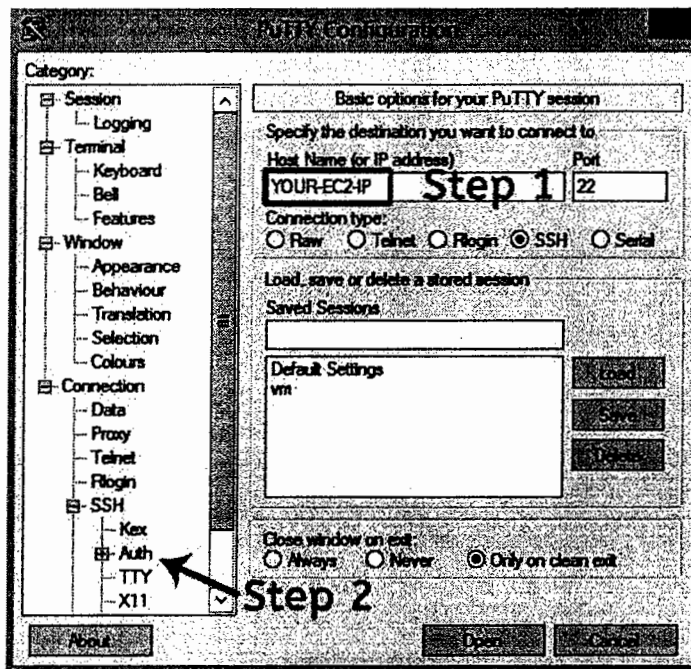
https://aws.amazon.com/amazon-linux-ami/2014.03-release-notes/
[ec2-user@ip-172-31-1-148 ~]$
```

Connect using Windows

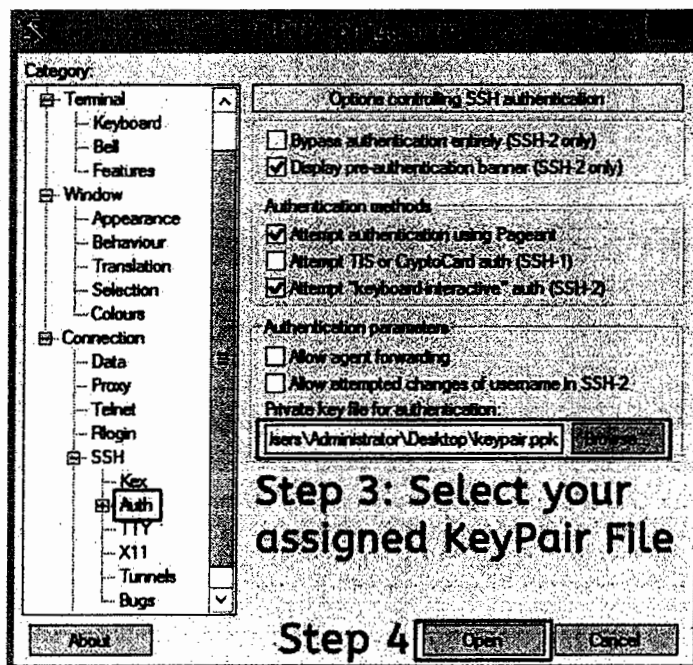
Windows has no SSH client, so you must use PuTTY and convert the PEM key to PPK using PuTTYgen.

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 6: Move between directories and list file content

Now you are ready to send the first commands to your Linux instance. Let's start by using some basic commands for browsing the file-system.

The **pwd** command will allow you to know in which directory you're located (**pwd** stands for "print working directory").

The **ls** command will show you the files in your current directory. Used with certain options (**ls -larth**), you can see sizes of files in a human readable version, when files were made, and permissions of files.

The **cd** command will allow you to change directories. When you open a terminal you will be in your home directory. To move around the file system you will use **cd**.

Examples:

To navigate into the root directory, use "**cd /**"

To navigate to your home directory, use "**cd**" or "**cd ~**". (The tilde (~) symbol stands for your home directory).

To navigate up one directory level, use "**cd ..**"

To navigate to the previous directory (or back), use "**cd -**"

To navigate through multiple levels of directory at once, specify the full directory path that you want to go to. For example, use, "**cd /var/www**" to go directly to the /www subdirectory of /var/.

```
[ec2-user@ip-172-31-1-148 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-1-148 ~]$ ls
[ec2-user@ip-172-31-1-148 ~]$ cd ..
[ec2-user@ip-172-31-1-148 ~]$ ls
.
[ec2-user@ip-172-31-1-148 ~]$ cd ..
[ec2-user@ip-172-31-1-148 ~]$ ls
[ec2-user@ip-172-31-1-148 ~]$ pwd
/
[ec2-user@ip-172-31-1-148 ~]$ cd /var/log
[ec2-user@ip-172-31-1-148 log]$ ls
.  boot  cloud-init-output.log  cronlog  lastlog  maillog  spooler  wtmp
boot.log  cloud-init.log  cron  cronlog  messages  secure  tallylog  yum.log
[ec2-user@ip-172-31-1-148 log]$ cd -
/
```

The **cat** command will output the contents of a specific file and can be used to concatenate and list files. The name is an abbreviation of *catenate*, a synonym of concatenate.

The **tail** command prints the last 10 lines of each file to standard output.

Here you can see a demo file composed by 20+ lines printed using tail and cat.

STEP 7: Manage files and their permissions

Here there are a series of useful command for copying, moving, renaming and removing files and directories.

The **cp** command will make a copy of a file for you.

Example: "**cp file foo**" will make an exact copy of "file" and name it "foo", but the file "file" will still be there.

If you are copying a directory, you must use "**cp -r directory foo**" (copy recursively).

The **mv** command will move a file to a different location or will rename a file. Examples:

"**mv file foo**" will rename the file "file" to "foo".

"**mv foo /tmp**" will move the file "foo" to the root temp directory, but it will not rename it. You must specify a new file name to rename a file.

The **rm** command will remove or delete a file in your directory.

The **rmdir** command will delete an empty directory. To delete a directory and all of its contents recursively, use **rm -r** instead.

The **mkdir** command will allow you to create directories. Example: "**mkdir music**" will create a directory called "music".

```
[ec2-user@ip-172-31-14-169 ~]$ cp test-file copied-file
[ec2-user@ip-172-31-14-169 ~]$ ls
copied-file  test-file
[ec2-user@ip-172-31-14-169 ~]$ mkdir new-dir
[ec2-user@ip-172-31-14-169 ~]$ ls
copied-file  test-file
[ec2-user@ip-172-31-14-169 ~]$ mv copied-file new-dir/
[ec2-user@ip-172-31-14-169 ~]$ ls new-dir/
copied-file
[ec2-user@ip-172-31-14-169 ~]$ ls
test-file
[ec2-user@ip-172-31-14-169 ~]$ rm new-dir/
rm: impossible to remove 'new-dir/': è una directory
[ec2-user@ip-172-31-14-169 ~]$ rm -r new-dir/
[ec2-user@ip-172-31-14-169 ~]$ ls
test-file
[ec2-user@ip-172-31-14-169 ~]$
```

Permission Management

*nix like Operating Systems (GNU/Linux, *BSD, MacOS...) manage resource access control using users and groups. Each file and directory has an ACL set to control who can read, write, and execute itself.

Two very important commands, **chmod** and **chown**, deal with permissions and ownership (respectively).

The chmod command allows you to change permissions on a file. The basic usage is: **chmod**

PERMISSIONS FILE

Where **PERMISSIONS** is either the numeric or the alpha equivalent of the permissions you want to assign and **FILE** is the list of file (or folder) you want to effect.

File permissions are in the form: OWNER | GROUP | OTHERS

Each of those sections includes: READ | WRITE | EXECUTE

Each permission (read, write, execute) is represented with the binary representation of the initial letter:

r - 4

w - 2

x - 1

To get the numeric permission you add which permissions you want to use together. It you want:

r+w you get 6.

r+w+x you get 7.

r+x you get 5.

r you get 4.

You need to set the permissions for three different "users" (Owner, Group, All Others). If you want Owner and Group to have full permissions (read, write and execute) and All Others to only have r permissions, you should use "774" as permission mask.

To change the permission of a particular file to 774 you should issue the command: **chmod**

774 FILENAME

```
SUNSET:test antonioangelino$ chmod 660 test.pem
SUNSET:test antonioangelino$ ls -l
total 8
-rw-r-----@ 1 antonioangelino  staff  1692 23 Jan 16:14 test.pem
SUNSET:test antonioangelino$ chmod 660 test.pem
SUNSET:test antonioangelino$ ls -l
total 8
-rw-rw----@ 1 antonioangelino  staff  1692 23 Jan 16:14 test.pem
SUNSET:test antonioangelino$ chmod 747 test.pem
SUNSET:test antonioangelino$ ls -l
total 8
-rwxr--rwx@ 1 antonioangelino  staff  1692 23 Jan 16:14
SUNSET:test antonioangelino$
```

If you want to change the ownership of a file or folder you need the **chown** command. Its basic usage is: **chown USER.GROUP FILE**

Where **USER** and **GROUP** are the new user and group that you want to assign to **FILE**. If you want to leave the **FILE** belonging to the original group, you can use **chown** without specifying the Owner **GROUP**.

STEP 8: Monitor and manage processes

There are some useful system commands that can help you to understand what is going on in your instance.

The **free** command displays the amount of free and used memory in the system. "**free -m**" will give the information using megabytes, which is probably most useful nowadays.

```
[ec2-user@ip-172-31-14-169 ~]$ free
              total        used        free      shared    buffers     cached
Mem:          1020196      319880      700316           0         9168      262016
-/+ buffers/cache:        48676      971520
Swap:           0           0           0
[ec2-user@ip-172-31-14-169 ~]$ free -m
              total        used        free      shared    buffers     cached
Mem:           986         312         674           0           8        255
-/+ buffers/cache:          47         942
Swap:           0           0           0
```

The **top** ('table of processes') command displays information on your Linux system, running processes and system resources, including CPU, RAM & swap usage and total number of tasks being run. To exit **top**, press "**q**".

If you want to order the processes per used memory, just press "M".

top also allows you to kill a process using its PID (Process ID). Press "k", insert the PID and then press "ENTER" twice.

The **df** command reports the amount of available disk space being used by file systems. Using **"df -h"** you can see all values converted in GB.

```
[ec2-user@ip-172-31-14-169 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1      7.8G  1.1G  6.7G  14% /
devtmpfs        491M   56K  491M   1% /dev
tmpfs           499M    0   499M   0% /dev/shm
```

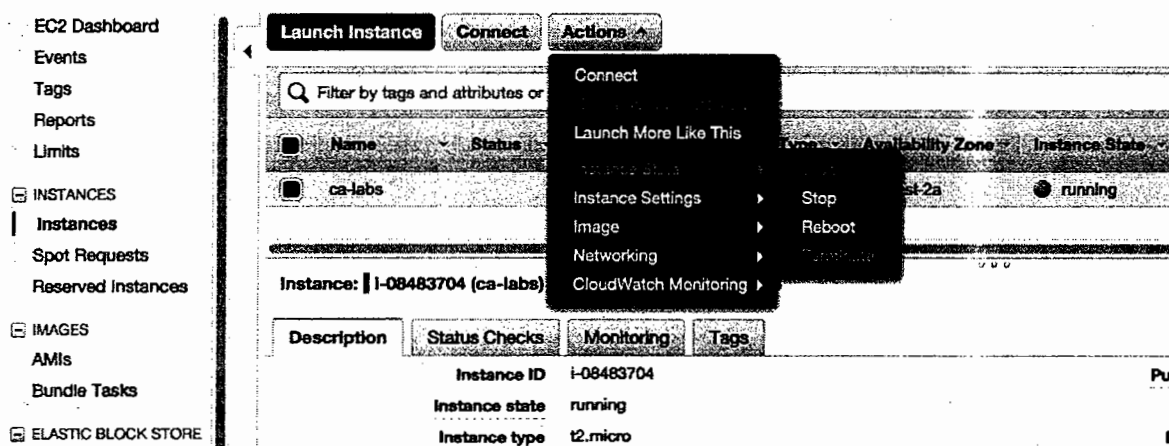
STEP 9: Terminate an EC2 instance

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

Select the EC2 service from the Management Console dashboard:

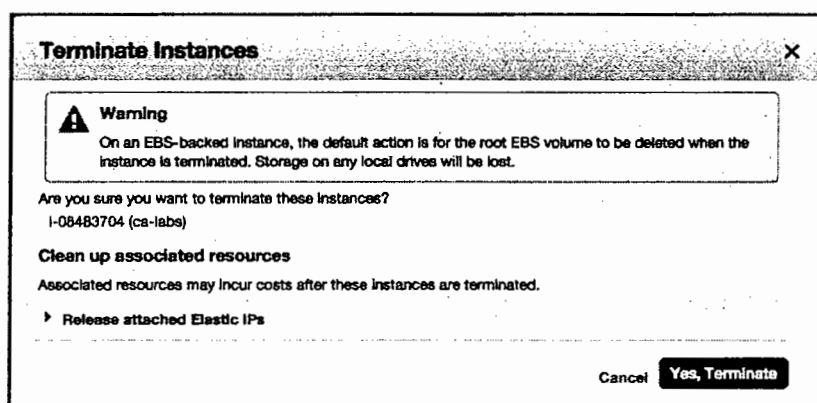


In the navigation pane, click **Instances**.



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

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



Now your instance is completely destroyed.

Launch Instance

Connect

Actions ▾

| <input type="checkbox"/> | Name ▾ | Status ▾ | Instance ID ▾ | Instance Type ▾ | Availability Zone ▾ | Instance State ▾ |
|-------------------------------------|---------|----------|---------------|-----------------|---------------------|------------------|
| <input checked="" type="checkbox"/> | ca-labs | | i-08483704 | t2.micro | us-west-2a | terminated |