Virtualized data center…

**1.) Cloud** is like On-demand service, where you can acquire resources easily and quickly….like

Demand for service like IT.

2.) **Ex:** Uber /Ola ---Taxi services

Amazon, Flipkart---e-commerce website.

Before this, we used to go to shopping mall, and buy which incurred us more expenses.

Now with e-commerce technology, flip kart etc…We log into website, demand for service …get it.

3.) Similarly for IT:

**EX-1:** As end user... Asking for 20GB of storage (Ask admin. Through servicenow req. and concerned team provision it for me) ---🡪Manual intervention required.

Now, with cloud coming into market, process is completely automated. Users are given with self-sign in portal where they can order for resources.

4.) **EX-2:**

Being a solution architect/developer/implementer..,Customer comes to IT team and ask that he want to build website like flipkart…Plz come with requirement..

I will say: I need to have DB server with master replication, APP server,10 VMS, 100 GB storage.

How long does it takes to complete infra????...

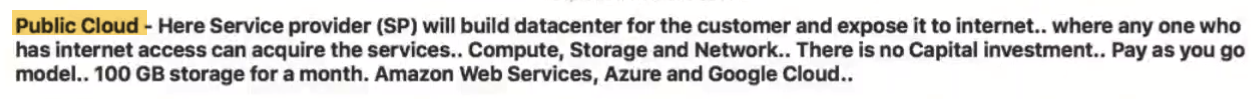
How long for 10 VMs with all OS, packages, DB installs etc…???---2-3 week’s min.

Is customer willing to wait for 3 days JUST for requirement to get implemented?

3 weeks for just infra only..???...No progress☹

This is where cloud comes into picture.We can provision 1000 vms in couple of clicks...

Cloud types:



**Advantage:** There is no capital investment. Initial investment is zero.

Let’s say we want to build s/w, app: there are 2 types of investments.

1. Investment on infra.
2. Investment on actual business,

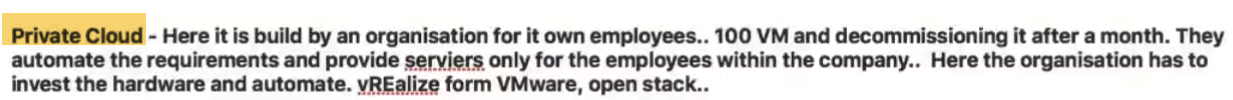
where it is in actual business, we get return of investment.

Any investor wants to reduce infra. Cost…invests more on business.. So he can get more business

b.) Need to purchase any hardware, storage.. Pay as you go.

Ex; If using 100 gb storage/ month. Pay only for that.

If using 10 instances. Pay only for that only…Not worried about hardware, S/W or where they are running.



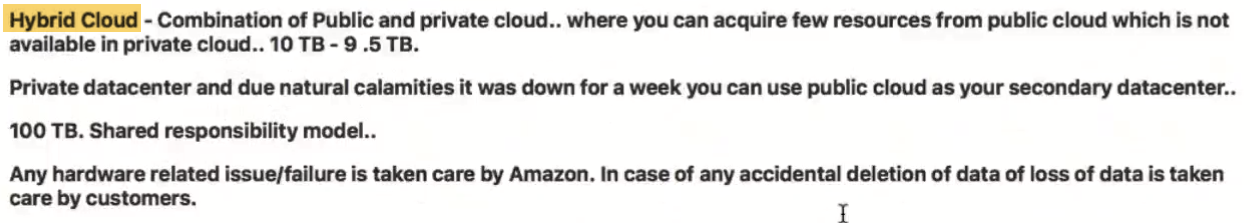
Provision 100 vms and decommission after a month…manually.

Restricted for particular organization…Their employees can only access.

Org. needs to invest on hardware.

You will be given with self-service portal, where you can order for services.

Lets say if we want 10 vms, 10 gb storage, 20 ip address.. We can order on catalog. No need to wait for admin.., and completely automated.



Ex-1: if we have private cloud, and in Nov month total, we have 10 TB of space and 9.5 TB already filled.

In Dec..Forcasted needs 3 GB minimum. At that point of time, procuring hardware, procuring S/W, attaching storage will take time. So we can go for public cloud, and acquire storage.

EX-2:We have private data center/private cloud, and due to some natural calamities down.Customer is not already to invest on other datacenter. Then we can go for public cloud for secondary data center.

When primary data center goes down, we can bring 2nd one which is on public cloud.

At this point of time, we need not pay on secondary data center.

Ex: When primary data center goes down, we can bring 2nd one which is on public cloud for 1 week ---pay for 1week.Not for whole year

**What Is Amazon EC2?**

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as **many or as few virtual servers as you need, configure security and networking, and manage storage**. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

## Features of Amazon EC2

* Virtual computing environments, known as instances
* Preconfigured templates for your instances, known as Amazon Machine Images (AMIs), that package the bits you need for your server (including the operating system and additional software)
* Various configurations of CPU, memory, storage, and networking capacity for your instances, known as instance types
* Secure login information for your instances using key pairs (AWS stores the public key, and you store the private key in a secure place)
* Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as instance store volumes
* Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as Amazon EBS volumes
* Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as regions and Availability Zones
* A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using security groups
* Static IPv4 addresses for dynamic cloud computing, known as Elastic IP addresses
* Metadata, known as tags, that you can create and assign to your Amazon EC2 resources
* Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as virtual private clouds (VPCs).

**How to Get Started with Amazon EC2**

**Get Up and Running**

* [Setting Up with Amazon EC2](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html)
* [Getting Started with Amazon EC2 Linux Instances](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html)

**Basics**

* [Instances and AMIs](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instances-and-amis.html)
* [Regions and Availability Zones](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html)
* [Instance Types](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html)
* [Tags](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html)

**Networking and Security**

* [Amazon EC2 Key Pairs](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html)
* [Security Groups](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html)
* [Elastic IP Addresses](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html)
* [Amazon EC2 and Amazon VPC](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-vpc.html)

**Storage**

* [Amazon EBS](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEBS.html)
* [Instance Store](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html)

**Working with Linux Instances**

* [Remote Management (Run Command)](http://docs.aws.amazon.com/systems-manager/latest/userguide/execute-remote-commands.html)
* [Tutorial: Installing a LAMP Web Server on Amazon Linux](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-LAMP.html)
* [Tutorial: Configure Apache Web Server on Amazon Linux to Use SSL/TLS](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/SSL-on-an-instance.html)
* [Getting Started with AWS: Hosting a Web App for Linux](http://docs.aws.amazon.com/gettingstarted/latest/wah-linux/)

**Setting Up with Amazon EC2**

You can open the Amazon EC2 console, choose **Launch Instance**, and follow the steps in the launch wizard to launch your first instance.

If you haven't signed up for AWS yet, or if you need assistance launching your first instance, complete the following tasks to get set up to use Amazon EC2:

1. [Sign Up for AWS](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html#sign-up-for-aws)
2. [Create an IAM User](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html#create-an-iam-user)
3. [Create a Key Pair](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html#create-a-key-pair)
4. [Create a Virtual Private Cloud (VPC)](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html#create-a-vpc)
5. [Create a Security Group](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html#create-a-base-security-group)

**Create an IAM User**

Services in AWS, such as Amazon EC2, require that you provide credentials when you access them, so that the service can determine whether you have permission to access its resources.

The console requires your password. You can create access keys for your AWS account to access the command line interface or API.

However, we don't recommend that you access AWS using the credentials for your AWS account; we recommend that you use AWS Identity and Access Management (IAM) instead. Create an IAM user, and then add the user to an IAM group with administrative permissions or grant this user administrative permissions. You can then access AWS using a special URL and the credentials for the IAM user.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM console. If you aren't familiar with using the console, see[Working with the AWS Management Console](http://docs.aws.amazon.com/awsconsolehelpdocs/latest/gsg/getting-started.html) for an overview.

**To create an IAM user for yourself and add the user to an Administrators group**

1. Use your AWS account email address and password to sign in to the [AWS Management Console](https://console.aws.amazon.com/) as the [*AWS account root user*](http://docs.aws.amazon.com/IAM/latest/UserGuide/id_root-user.html).
2. In the navigation pane of the console, choose **Users**, and then choose **Add user**.
3. For **User name**, type **Administrator**.
4. Select the check box next to **AWS Management Console access**, select **Custom password**, and then type the new user's password in the text box. You can optionally select **Require password reset** to force the user to select a new password the next time the user signs in.
5. Choose **Next: Permissions**.
6. On the **Set permissions for user** page, choose **Add user to group**.
7. Choose **Create group**.
8. In the **Create group** dialog box, type **Administrators**.
9. For **Filter**, choose **Job function**.
10. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.
11. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
12. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

You can use this same process to create more groups and users, and to give your users access to your AWS account resources. To learn about using policies to restrict users' permissions to specific AWS resources, go to [Access Management](http://docs.aws.amazon.com/IAM/latest/UserGuide/access.html) and [Example Policies](http://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_examples.html).

To sign in as this new IAM user, sign out of the AWS console, then use the following URL, where *your\_aws\_account\_id* is your AWS account number without the hyphens (for example, if your AWS account number is 1234-5678-9012, your AWS account ID is 123456789012):

https://*your\_aws\_account\_id*.signin.aws.amazon.com/console/

Enter the IAM user name (not your email address) and password that you just created. When you're signed in, the navigation bar displays "*your\_user\_name* @*your\_aws\_account\_id*".

If you don't want the URL for your sign-in page to contain your AWS account ID, you can create an account alias. From the IAM console, choose **Dashboard** in the navigation pane. From the dashboard, choose **Customize** and enter an alias such as your company name. To sign in after you create an account alias, use the following URL:

https://*your\_account\_alias*.signin.aws.amazon.com/console/

To verify the sign-in link for IAM users for your account, open the IAM console and check under **IAM users sign-in link** on the dashboard.

For more information about IAM, see [IAM and Amazon EC2](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/UsingIAM.html#intro-to-iam).

**Create a Key Pair**

AWS uses public-key cryptography to secure the login information for your instance. A Linux instance has no password; you use a key pair to log in to your instance securely. You specify the name of the key pair when you launch your instance, then provide the private key when you log in using SSH.

If you haven't created a key pair already, you can create one using the Amazon EC2 console. Note that if you plan to launch instances in multiple regions, you'll need to create a key pair in each region. For more information about regions, see [Regions and Availability Zones](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html).

**To create a key pair**

1. Sign in to AWS using the URL that you created in the previous section.
2. From the AWS dashboard, choose **EC2** to open the Amazon EC2 console.
3. From the navigation bar, select a region for the key pair. You can select any region that's available to you, regardless of your location. However, key pairs are specific to a region; for example, if you plan to launch an instance in the US East (Ohio) Region, you must create a key pair for the instance in the US East (Ohio) Region.


      Select a region
     

1. In the navigation pane, under **NETWORK & SECURITY**, choose **Key Pairs**.

**Tip**

The navigation pane is on the left side of the console. If you do not see the pane, it might be minimized; choose the arrow to expand the pane. You may have to scroll down to see the **Key Pairs** link.


      Open the key pairs page
     

1. Choose **Create Key Pair**.
2. Enter a name for the new key pair in the **Key pair name** field of the **Create Key Pair** dialog box, and then choose **Create**. Use a name that is easy for you to remember, such as your IAM user name, followed by -key-pair, plus the region name. For example, *me*-key-pair-*useast2*.
3. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is .pem. Save the private key file in a safe place.

**Important**

This is the only chance for you to save the private key file. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

1. If you will use an SSH client on a Mac or Linux computer to connect to your Linux instance, use the following command to set the permissions of your private key file so that only you can read it.

**Copy**

**chmod 400 *your\_user\_name*-key-pair-*region\_name*.pem**

For more information, see [Amazon EC2 Key Pairs](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html).

**To connect to your instance using your key pair**

To connect to your Linux instance from a computer running Mac or Linux, you'll specify the .pem file to your SSH client with the -i option and the path to your private key. To connect to your Linux instance from a computer running Windows, you can use either MindTerm or PuTTY. If you plan to use PuTTY, you'll need to install it and use the following procedure to convert the .pem file to a .ppk file.

**(Optional) To prepare to connect to a Linux instance from Windows using PuTTY**

1. Download and install PuTTY from <http://www.chiark.greenend.org.uk/~sgtatham/putty/>. Be sure to install the entire suite.
2. Start PuTTYgen (for example, from the **Start** menu, choose **All Programs > PuTTY > PuTTYgen**).
3. Under **Type of key to generate**, choose **RSA**.


      SSH-2 RSA key in PuTTYgen
     

1. Choose **Load**. By default, PuTTYgen displays only files with the extension .ppk. To locate your .pem file, select the option to display files of all types.


      Select all file types
     

1. Select the private key file that you created in the previous procedure and choose **Open**. Choose **OK** to dismiss the confirmation dialog box.
2. Choose **Save private key**. PuTTYgen displays a warning about saving the key without a passphrase. Choose **Yes**.
3. Specify the same name for the key that you used for the key pair. PuTTY automatically adds the .ppk file extension.

**Create a Virtual Private Cloud (VPC)**

Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. If you have a default VPC, you can skip this section and move to the next task, [Create a Security Group](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html#create-a-base-security-group). To determine whether you have a default VPC, see [Supported Platforms in the Amazon EC2 Console](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-supported-platforms.html#console-updates). Otherwise, you can create a nondefault VPC in your account using the steps below.

**Important**

If your account supports EC2-Classic in a region, then you do not have a default VPC in that region. T2 instances must be launched into a VPC.

**To create a nondefault VPC**

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. From the navigation bar, select a region for the VPC. VPCs are specific to a region, so you should select the same region in which you created your key pair.
3. On the VPC dashboard, choose **Start VPC Wizard**.
4. On the **Step 1: Select a VPC Configuration** page, ensure that **VPC with a Single Public Subnet** is selected, and choose **Select**.
5. On the **Step 2: VPC with a Single Public Subnet** page, enter a friendly name for your VPC in the **VPC name** field. Leave the other default configuration settings, and choose **Create VPC**. On the confirmation page, choose **OK**.

For more information about Amazon VPC, see [What is Amazon VPC?](http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/) in the *Amazon VPC User Guide*.

**Create a Security Group**

Security groups act as a firewall for associated instances, controlling both inbound and outbound traffic at the instance level. You must add rules to a security group that enable you to connect to your instance from your IP address using SSH. You can also add rules that allow inbound and outbound HTTP and HTTPS access from anywhere.

Note that if you plan to launch instances in multiple regions, you'll need to create a security group in each region. For more information about regions, see [Regions and Availability Zones](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html).

**Prerequisites**

You'll need the public IPv4 address of your local computer. The security group editor in the Amazon EC2 console can automatically detect the public IPv4 address for you. Alternatively, you can use the search phrase "what is my IP address" in an Internet browser, or use the following service: [Check IP](http://checkip.amazonaws.com/). If you are connecting through an Internet service provider (ISP) or from behind a firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

**To create a security group with least privilege**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

**Tip**

Alternatively, you can use the Amazon VPC console to create a security group. However, the instructions in this procedure don't match the Amazon VPC console. Therefore, if you switched to the Amazon VPC console in the previous section, either switch back to the Amazon EC2 console and use these instructions, or use the instructions in [Set Up a Security Group for Your VPC](http://docs.aws.amazon.com/AmazonVPC/latest/GettingStartedGuide/SecurityGroup.html) in the *Amazon VPC Getting Started Guide*.

1. From the navigation bar, select a region for the security group. Security groups are specific to a region, so you should select the same region in which you created your key pair.


      Select a region
     

1. Choose **Security Groups** in the navigation pane.
2. Choose **Create Security Group**.
3. Enter a name for the new security group and a description. Use a name that is easy for you to remember, such as your IAM user name, followed by \_SG\_, plus the region name. For example, *me*\_SG\_*uswest2*.
4. In the **VPC** list, select your VPC. If you have a default VPC, it's the one that is marked with an asterisk (\*).

**Note**

If your account supports EC2-Classic, select the VPC that you created in the previous task.

1. On the **Inbound** tab, create the following rules (choose **Add Rule** for each new rule), and then choose **Create**:
   * Choose **HTTP** from the **Type** list, and make sure that **Source** is set to **Anywhere** (0.0.0.0/0).
   * Choose **HTTPS** from the **Type** list, and make sure that **Source** is set to **Anywhere** (0.0.0.0/0).
   * Choose **SSH** from the **Type** list. In the **Source** box, choose **My IP** to automatically populate the field with the public IPv4 address of your local computer. Alternatively, choose **Custom** and specify the public IPv4 address of your computer or network in CIDR notation. To specify an individual IP address in CIDR notation, add the routing suffix /32, for example, 203.0.113.25/32. If your company allocates addresses from a range, specify the entire range, such as 203.0.113.0/24.

**Warning**

For security reasons, we don't recommend that you allow SSH access from all IPv4 addresses (0.0.0.0/0) to your instance, except for testing purposes and only for a short time.

# Getting Started with Amazon EC2 Linux InstancesAn instance is a virtual server in the AWS cloud. With Amazon EC2, you can set up and configure the operating system and applications that run on your instance.

# You’ll incur the standard Amazon EC2 usage fees from the time that you launch the instance until you terminate the instance (which is the final task of this tutorial), even if it remains idle.

**Contents**

* [Overview](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html#ec2-get-started-overview)
* [Prerequisites](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html#ec2-getstarted-prereqs)
* [Step 1: Launch an Instance](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html#ec2-launch-instance)
* [Step 2: Connect to Your Instance](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html#ec2-connect-to-instance-linux)
* [Step 3: Clean Up Your Instance](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html#ec2-clean-up-your-instance)
* [Next Steps](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html#ec2-next-steps)

## Overview

The instance is an Amazon EBS-backed instance (meaning that the root volume is an EBS volume). You can either specify the Availability Zone in which your instance runs, or let Amazon EC2 select an Availability Zone for you. When you launch your instance, you secure it by specifying a key pair and security group. When you connect to your instance, you must specify the private key of the key pair that you specified when launching your instance.


     An Amazon EBS-backed instance with an additional Amazon Elastic Block Store (EBS) volume
    

**Tasks**

To complete this tutorial, perform the following tasks:

1. [Launch an Instance](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html#ec2-launch-instance)
2. [Connect to Your Instance](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html#ec2-connect-to-instance-linux)
3. [Clean Up Your Instance](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html#ec2-clean-up-your-instance)

**Related Tutorials**

* If you'd prefer to launch a Windows instance, see this tutorial in the Amazon EC2 User Guide for Windows Instances: [Getting Started with Amazon EC2 Windows Instances](http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/EC2_GetStarted.html).
* If you'd prefer to use the command line, see this tutorial in the AWS Command Line Interface User Guide: [Using Amazon EC2 through the AWS CLI](http://docs.aws.amazon.com/cli/latest/userguide/cli-using-ec2.html).

## Prerequisites

Before you begin, be sure that you've completed the steps in [Setting Up with Amazon EC2](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html).

## Step 1: Launch an Instance

You can launch a Linux instance using the AWS Management Console as described in the following procedure. This tutorial is intended to help you launch your first instance quickly, so it doesn't cover all possible options. For more information about the advanced options, see [Launching an Instance](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/launching-instance.html).

**To launch an instance**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the console dashboard, choose **Launch Instance**.
3. 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 HVM edition of the Amazon Linux AMI. Notice that this AMI is marked "Free tier eligible."
4. On the **Choose an Instance Type** page, you can select the hardware configuration of your instance. Select the t2.micro type, which is selected by default. Notice that this instance type is eligible for the free tier.

**Note**

[T2 instances](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/t2-instances.html), such as t2.micro, must be launched into a VPC. If your AWS account supports EC2-Classic and you do not have a VPC in the selected region, the launch wizard creates a VPC for you and you can continue to the next step. Otherwise, the **Review and Launch** button is disabled and you must choose **Next: Configure Instance Details** and follow the directions to select a subnet.

1. Choose **Review and Launch** to let the wizard complete the other configuration settings for you.
2. On the **Review Instance Launch** page, under **Security Groups**, you'll see that the wizard created and selected a security group for you. You can use this security group, or alternatively you can select the security group that you created when getting set up using the following steps:
   1. Choose **Edit security groups**.
   2. On the **Configure Security Group** page, ensure that **Select an existing security group** is selected.
   3. Select your security group from the list of existing security groups, and then choose **Review and Launch**.
3. On the **Review Instance Launch** page, choose **Launch**.
4. When prompted for a key pair, select **Choose an existing key pair**, then select the key pair that you created when getting set up.

Alternatively, you can create a new key pair. Select **Create a new key pair**, enter a name for the key pair, and then choose **Download Key Pair**. This is the only chance for you to save the private key file, so be sure to download it. Save the private key file in a safe place. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

**Warning**

Don't select the **Proceed without a key pair** option. If you launch your instance without a key pair, then you can't connect to it.

When you are ready, select the acknowledgement check box, and then choose **Launch Instances**.

1. A confirmation page lets you know that your instance is launching. Choose **View Instances** to close the confirmation page and return to the console.
2. On the **Instances** screen, you can view the status of the launch. It takes a short time for an instance to launch. When you launch an instance, its initial state is pending. After the instance starts, its state changes to runningand it receives a public DNS name. (If the **Public DNS (IPv4)** column is hidden, choose **Show/Hide Columns** (the gear-shaped icon) in the top right corner of the page and then select **Public DNS (IPv4)**.)
3. It can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks; you can view this information in the **Status Checks** column.

## Step 2: Connect to Your Instance

There are several ways to connect to a Linux instance. In this procedure, you'll connect using your browser. Alternatively, you can connect using PuTTY or an SSH client. It's also assumed that you followed the steps earlier and launched an instance from an Amazon Linux AMI, which has a specific user name. Other Linux distributions may use a different user name. For more information, see [Connecting to Your Linux Instance from Windows Using PuTTY](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html) or [Connecting to Your Linux Instance Using SSH](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html).

**Important**

You can't connect to your instance unless you launched it with a key pair for which you have the .pem file and you launched it with a security group that allows SSH access. If you can't connect to your instance, see[Troubleshooting Connecting to Your Instance](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/TroubleshootingInstancesConnecting.html) for assistance.

**To connect to your Linux instance using a web browser**

1. You must have Java installed and enabled in the browser. If you don't have Java already, you can contact your system administrator to get it installed, or follow the steps outlined in the following pages: [Install Java](http://java.com/en/download/help/index_installing.xml) and [Enable Java in your web browser](http://java.com/en/download/help/enable_browser.xml).
2. From the Amazon EC2 console, choose **Instances** in the navigation pane.
3. Select the instance, and then choose **Connect**.
4. Choose **A Java SSH client directly from my browser (Java required)**.
5. Amazon EC2 automatically detects the public DNS name of your instance and populates **Public DNS** for you. It also detects the key pair that you specified when you launched the instance. Complete the following, and then choose **Launch SSH Client**.
   1. In **User name**, enter ec2-user.
   2. In **Private key path**, enter the fully qualified path to your private key (.pem) file, including the key pair name.
   3. (Optional) Choose **Store in browser cache** to store the location of the private key in your browser cache. This enables Amazon EC2 to detect the location of the private key in subsequent browser sessions, until you clear your browser's cache.
6. If necessary, choose **Yes** to trust the certificate, and choose **Run** to run the MindTerm client.
7. If this is your first time running MindTerm, a series of dialog boxes asks you to accept the license agreement, confirm setup for your home directory, and confirm setup of the known hosts directory. Confirm these settings.
8. A dialog prompts you to add the host to your set of known hosts. If you do not want to store the host key information on your local computer, choose **No**.

A window opens and you are connected to your instance.

**Note**

If you chose **No** in the previous step, you'll see the following message, which is expected:

Verification of server key disabled in this session.

## Step 3: Clean Up Your Instance

After you've finished with the instance that you created for this tutorial, you should clean up by terminating the instance. If you want to do more with this instance before you clean up, see [Next Steps](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html#ec2-next-steps).

**Important**

Terminating an instance effectively deletes it; you can't reconnect to an instance after you've terminated it.

If you launched an instance that is not within the [AWS Free Tier](http://aws.amazon.com/free/), you'll stop incurring charges for that instance as soon as the instance status changes to shutting down or terminated. If you'd like to keep your instance for later, but not incur charges, you can stop the instance now and then start it again later. For more information, see [Stopping Instances](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Stop_Start.html).

**To terminate your instance**

1. In the navigation pane, choose **Instances**. In the list of instances, select the instance.
2. Choose **Actions**, **Instance State**, **Terminate**.
3. Choose **Yes, Terminate** when prompted for confirmation.

Amazon EC2 shuts down and terminates your instance. After your instance is terminated, it remains visible on the console for a short while, and then the entry is deleted.