**Hands on lab for VM migration**

This Lab is to migrate a Web Servers Running on your premises to AWS Cloud

On Premise Architecture



**AWS Architecture**

**Availability Zone**

**Availability Zone**

**Auto Scaling group**

**region**

**security group**



**Amazon EC2**



**Amazon EC2**

**Web Server AMI**



**Amazon EC2**



**Amazon EC2**

**Migration Bucket**



**Data Volume Snapshot**



**Data volume**



**Data volume**



**Data volume**



**Data volume**

**Elastic Load Balancing**



**Image bucket**



**Amazon CloudFront**



**Amazon  
Route 53**

Step 1: Import On Premise Image to AWS



1. Install AWS CLI and EC2 CLI

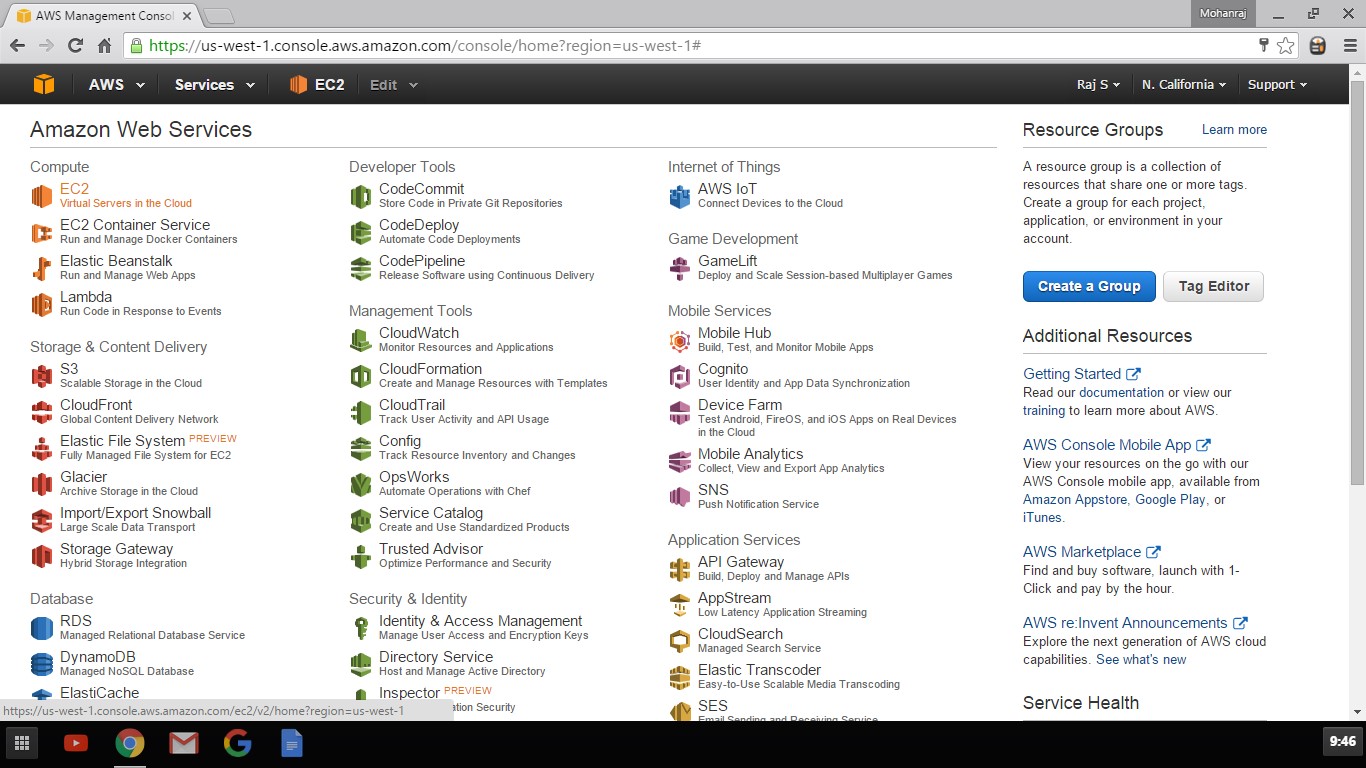
Refer the Document Install AWS Command line document to install locally, In this lab we will use AWS build server to do the job which will have AWS cli preinstalled

1. Start a AWS Linux Build Instance in AWS.

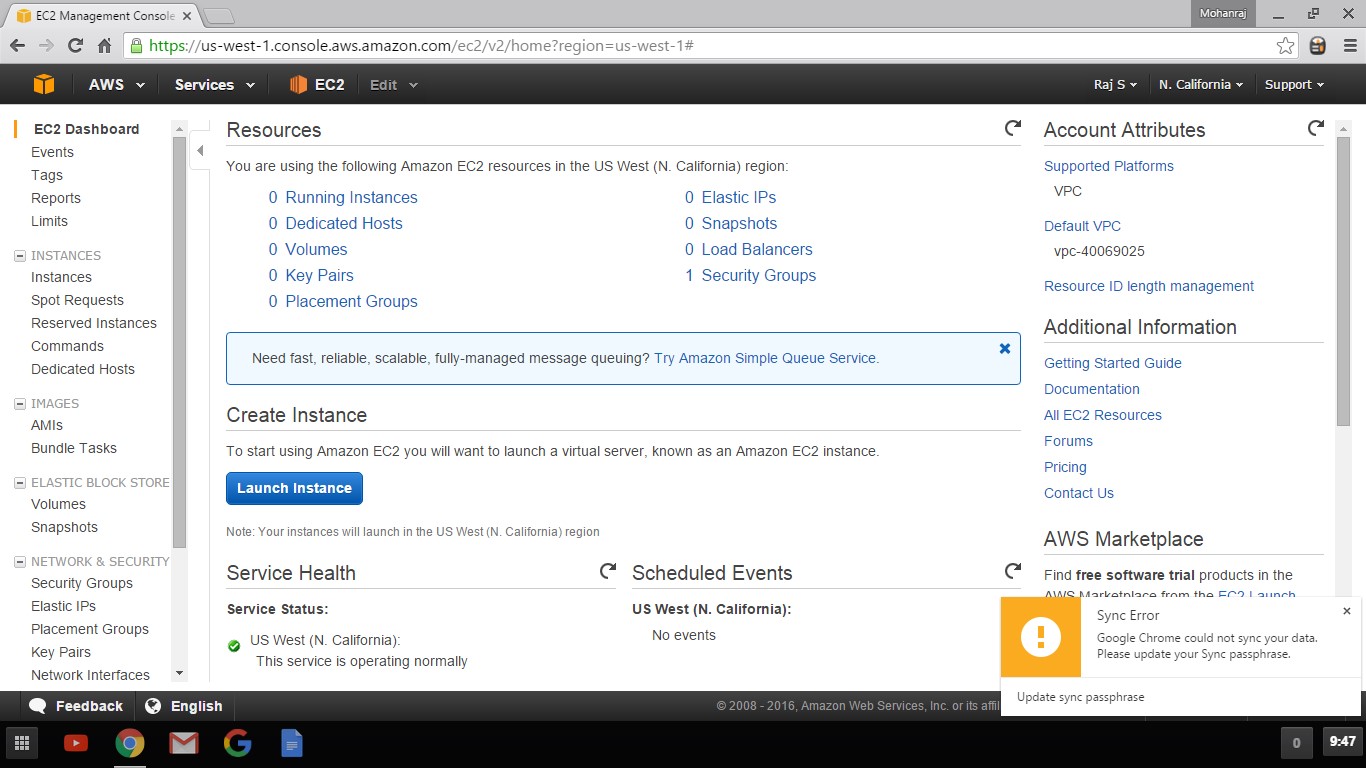
Open Console

<https://console.aws.amazon.com/console/home>

Click EC2 to create Instance or VM



Click Launch Instance



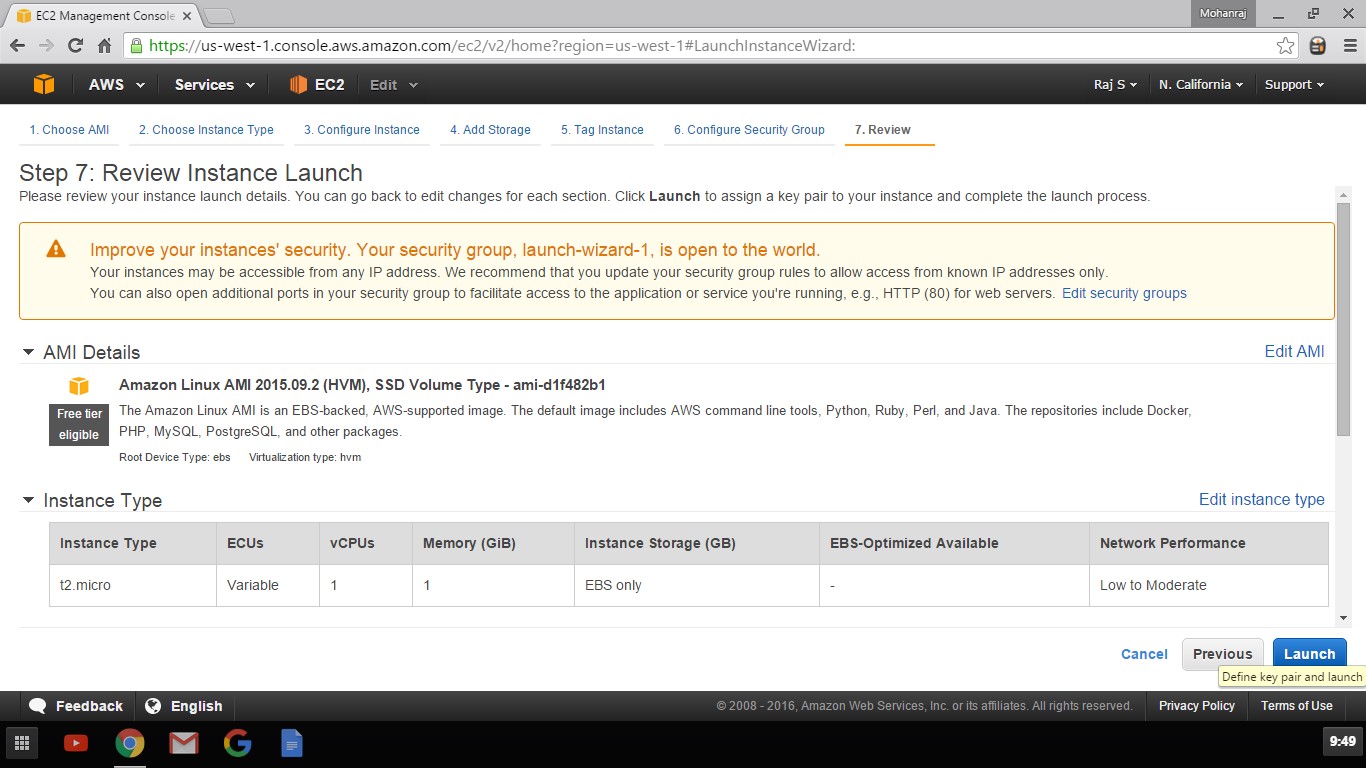
Select Amazon Linux AMI



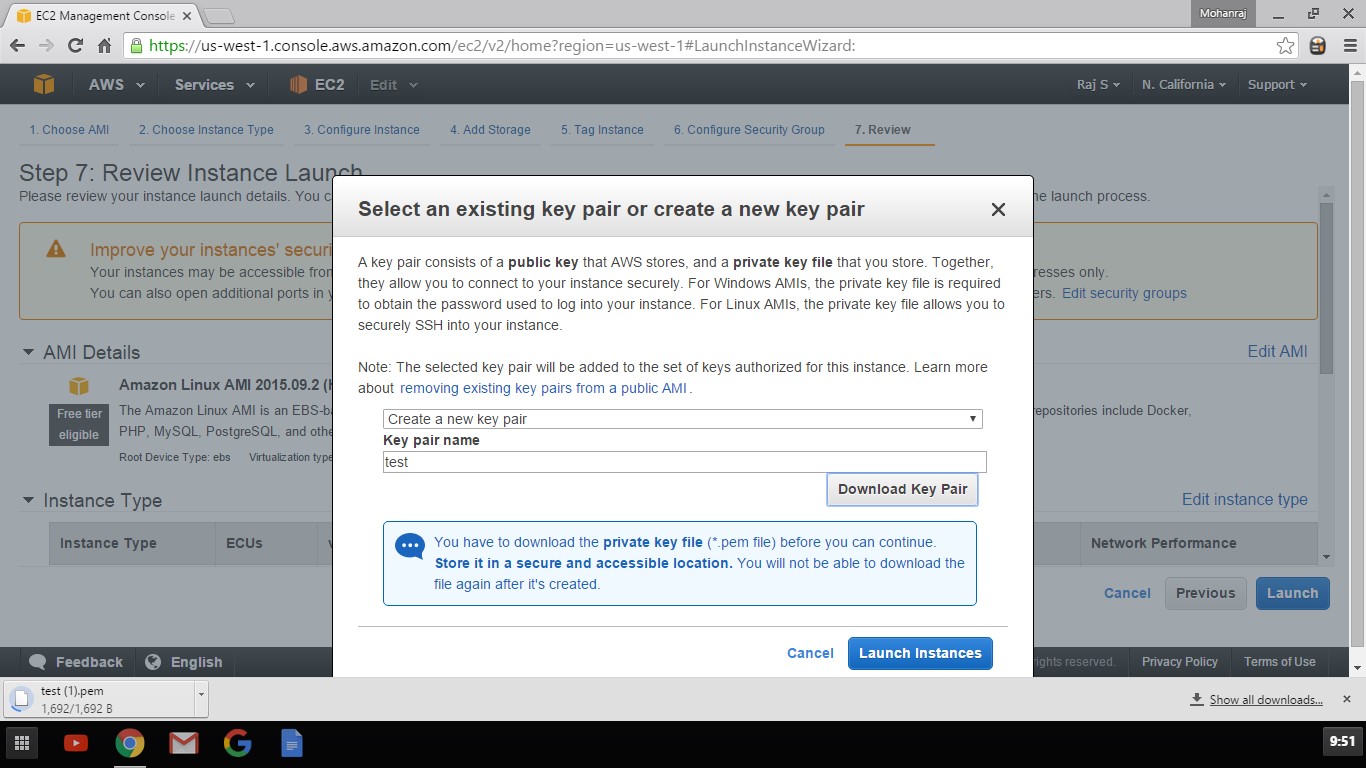
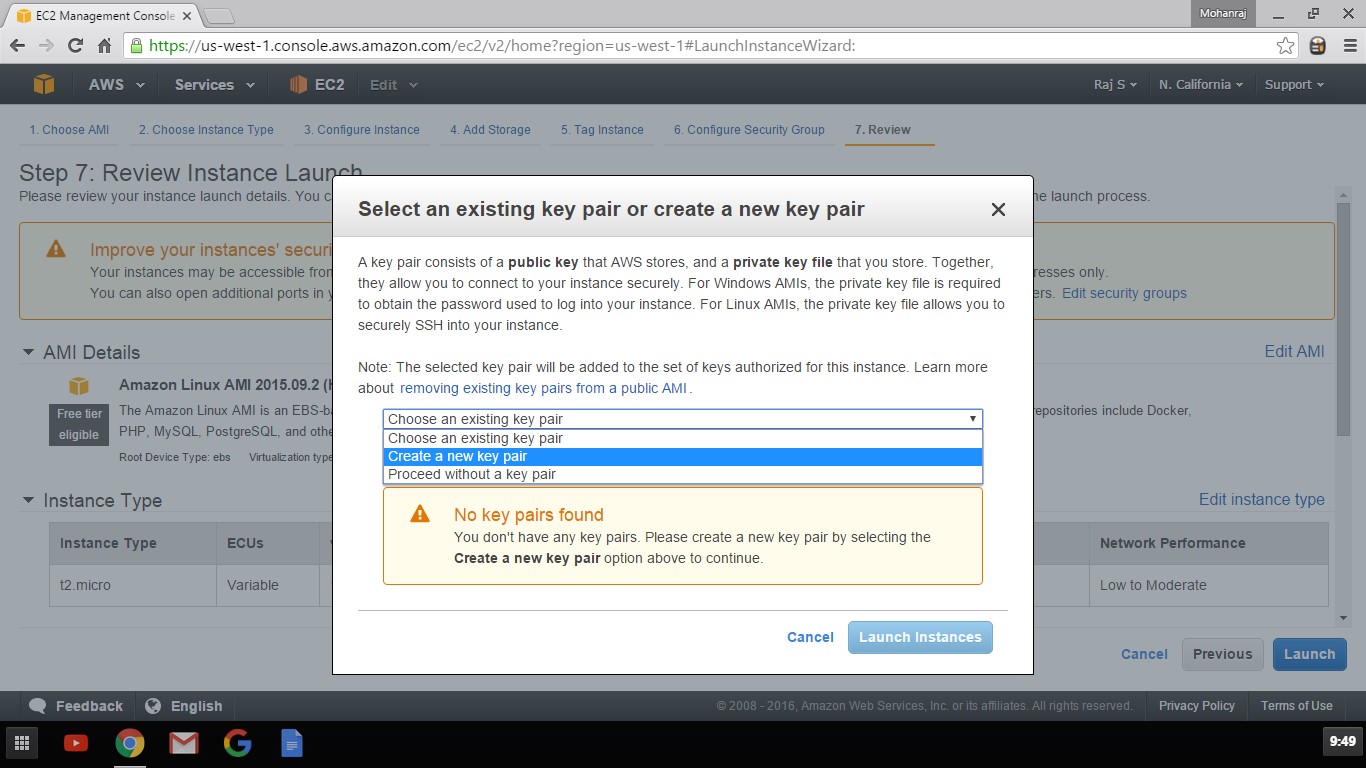
Choose T2.Micro and clock Review and Launch



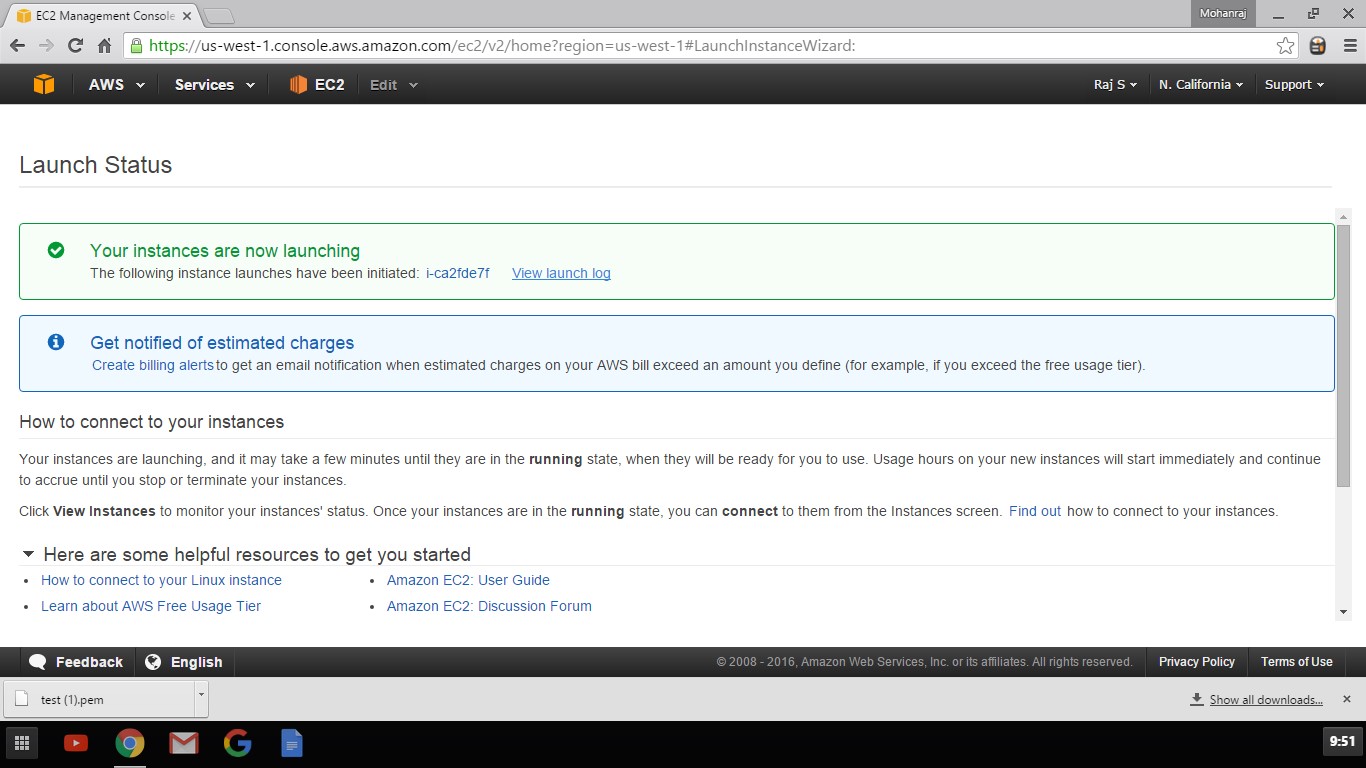
Click Launch in Review Screen



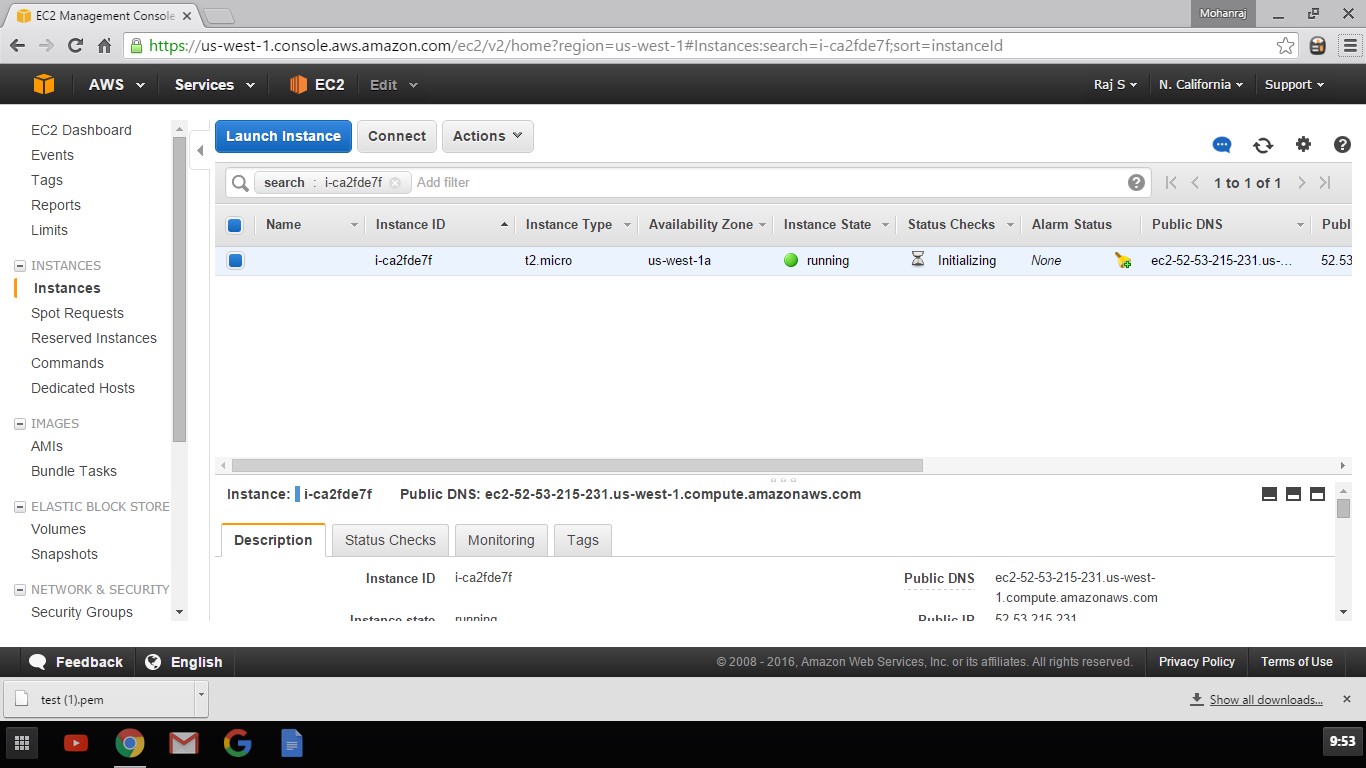
Choose Create New Key Pair and provide Name and Download Key

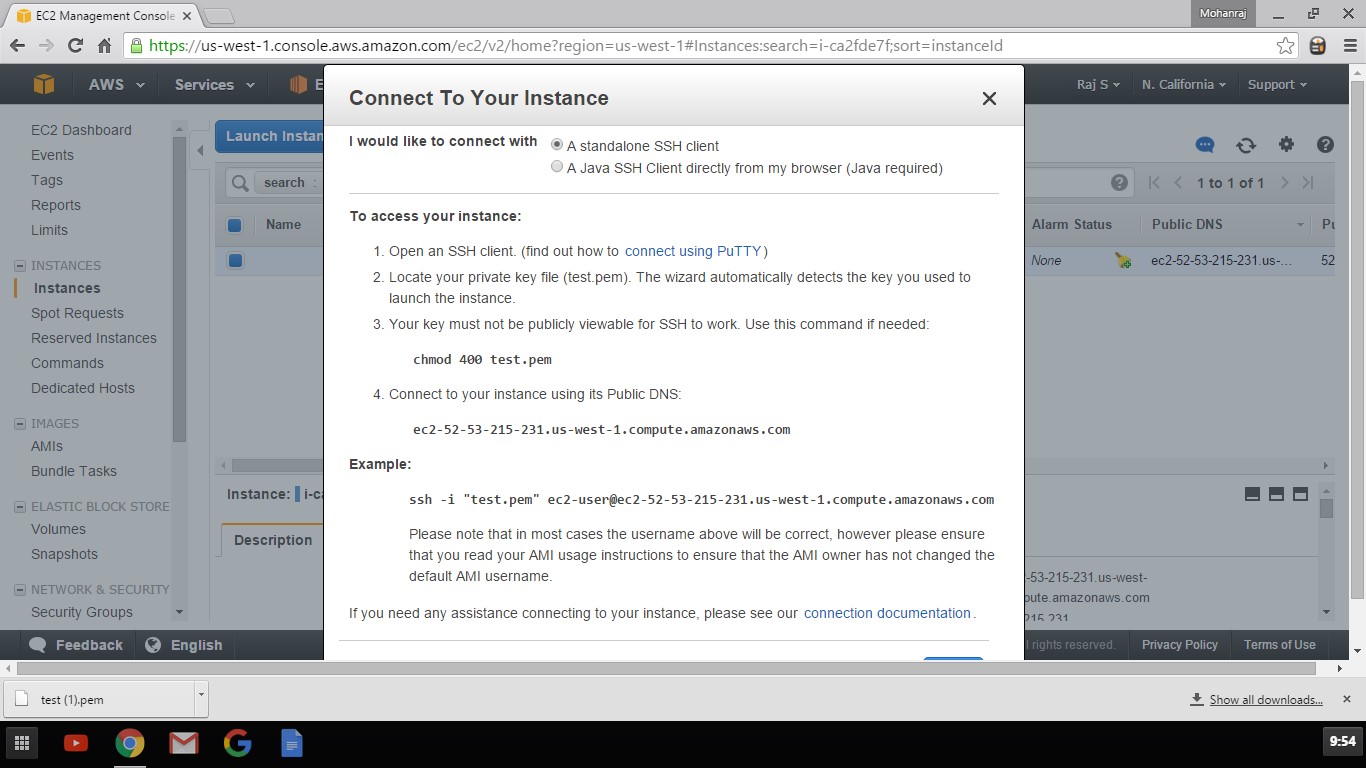


Click view instance Id i-xxxxxxx to take you EC2 Console



Click Connect tab to find the Options to connect



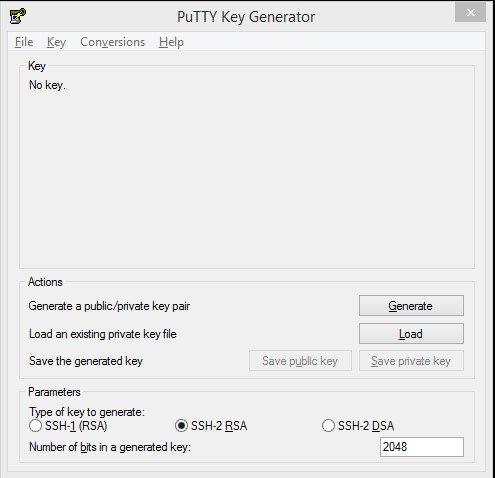


**Connecting using Putty**

Download Putty and PuttyGen from the below URL <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

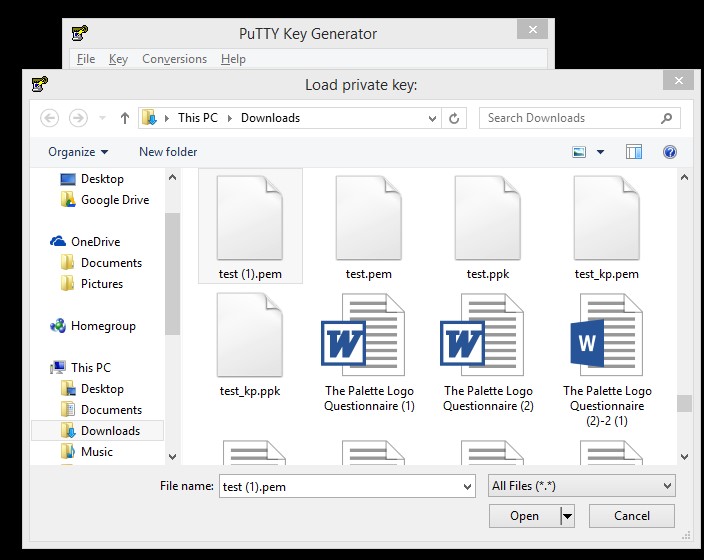
Open PuttyGen

Open Putty Key Generator

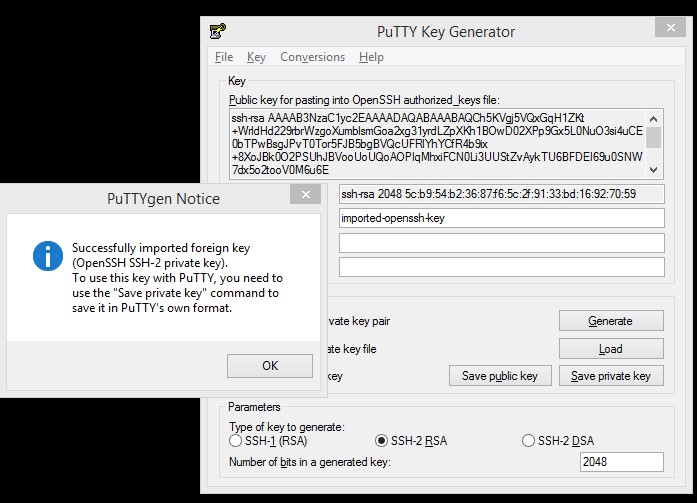


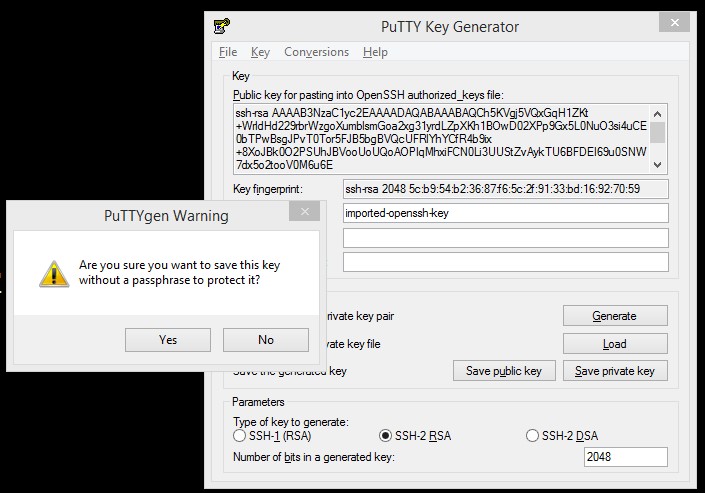
Click Load and lead the PEM File downloaded while creating the Instance

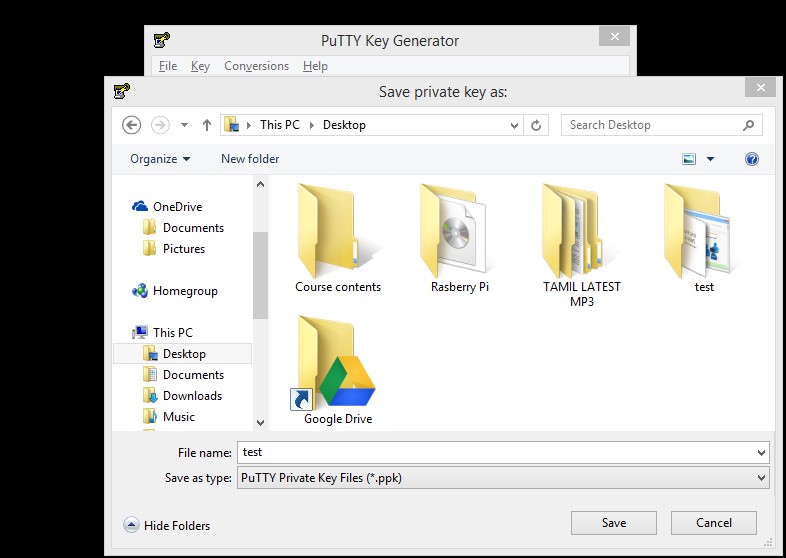
Select All Files to show the .pem files



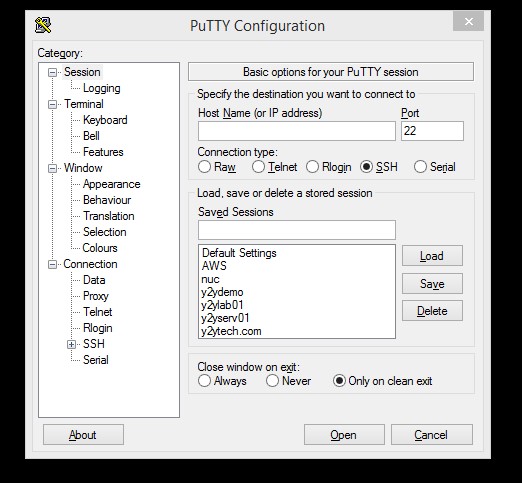
# Use Save private key to Save it in PPK Format which Putty Understands





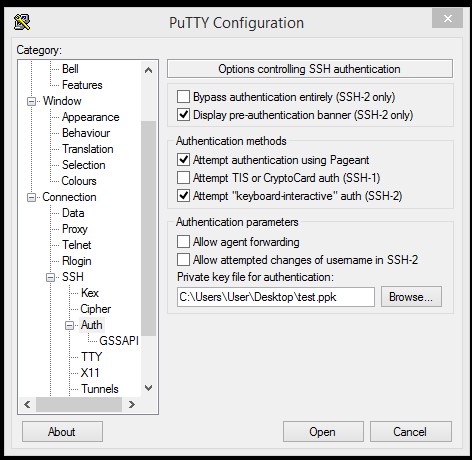


# Open Putty

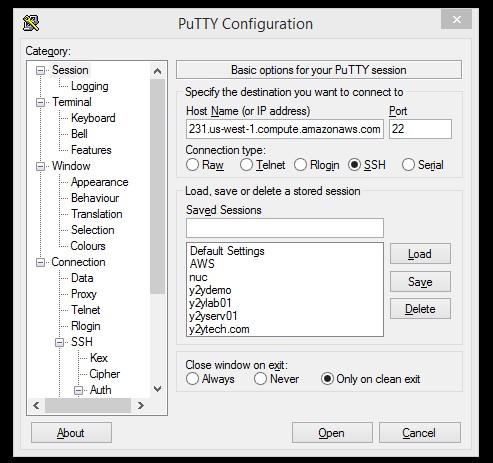


# Click SSH-> Auth -> Choose the private key saved





# Go to session and Put the DNS name show in Connect Tab of AWS console



# Enter ec2-user as User name



Step 3: Migrate your On Premise VM root disk to AWS AMI

Before Migrating the VM, We need to prepare the VM

# Prepare Your VM

Use the following guidelines to configure your VM before exporting it from the virtualization environment.

* Review the prerequisites.
* Disable any antivirus or intrusion detection, software on your VM. These services can be re-enabled after the import process is complete.
* Uninstall the VMware Tools from your VMware VM.
* Disconnect any CD-ROM drives (virtual or physical).
* Set your network to DHCP instead of a static IP address. If you want to assign a static private IP address, be sure to use a non-reserved private IP address in your VPC subnet. Amazon Virtual Private Cloud (Amazon VPC) reserves the first four private IP addresses in a VPC subnet.
* Shut down your VM before exporting it from your virtualization environment.
* Amazon EC2 automatically assigns a private DHCP IP address to your instance. The DNS name and IP address are available through the ec2-describe-instancescommand when the instance starts running.
* Your instance will have only one Ethernet network interface.
* We recommend that your Windows instances contain strong passwords for all user accounts. We recommend that your Linux instances use public keys for SSH.
* For Windows instances, we recommend that you install the latest version of the Amazon Windows EC2Config Service after you import your virtual machine into Amazon EC2.

**Windows**

* Enable Remote Desktop (RDP) for remote access.
* Make sure that your host firewall (Windows firewall or similar), if configured, allows access to RDP. Otherwise, you will not be able to access your instance after the import is complete.
* Make sure that the administrator account and all other user accounts use secure passwords. All accounts must have passwords or the importation might fail.
* Make sure that your Windows VM has .NET Framework 3.5 or later installed, as required by [Amazon Windows EC2Config Service](http://aws.amazon.com/developertools/5562082477397515).
* You can run System Preparation (Sysprep) on your Windows Server 2008 or Windows Server 2012 VM images before or after they are imported. If you run Sysprep before importing your VM, the importation process adds an answer file (unattend.xml) to the VM that automatically accepts the End User License Agreement (EULA) and sets the locale to EN-US. If you choose to run Sysprep after importation, we recommend that you use the Amazon EC2 Config service to run Sysprep.

**To include your own answer file instead of the default (unattend.xml):**

* 1. Copy the sample unattend.xml file below and set the **processorArchitecture**parameter to **x86** or **amd64**, depending on your OS architecture:

<?xml version='1.0' encoding='UTF-8'?>

<unattend xmlns:wcm='http://schemas.microsoft.com/WMIConfig/2002/State' xmlns='urn:schemas-microsoft-com:unattend'>

<settings pass='oobeSystem'>

<component versionScope='nonSxS' processorArchitecture='*x86 or amd64*' name='Microsoft-Windows-International-Core' publicKeyToken='31bf3856ad364e35' language='neutral'>

<InputLocale>en-US</InputLocale>

<SystemLocale>en-US</SystemLocale>

<UILanguage>en-US</UILanguage>

<UserLocale>en-US</UserLocale>

</component>

<component versionScope='nonSxS' processorArchitecture='*x86 or amd64*' name='Microsoft-Windows-Shell-Setup' publicKeyToken='31bf3856ad364e35' language='neutral'>

<OOBE>

<HideEULAPage>true</HideEULAPage>

<SkipMachineOOBE>true</SkipMachineOOBE>

<SkipUserOOBE>true</SkipUserOOBE>

</OOBE>

</component>

</settings>

</unattend>

* 1. Save the file in the **C:\Windows\Panther** directory with the name **unattend.xml**.
  2. Run Sysprep with the **/oobe** and **/generalize** options.

**Note**

The **/oobe** and **/generalize** options strip all unique system information from the Microsoft Windows installation and will prompt you to reset the administrator password.

* 1. Shutdown the VM and export it from your virtualization environment.
* Disable Autologon on your Windows VM.
* Open **Control Panel** > **System and Security** > **Windows Update**. In the left pane, choose **Change settings**. Choose the desired setting. Be aware that if you choose**Download updates but let me choose whether to install them** (the default value) the update check can temporarily consume between 50% and 99% of CPU resources on the instance. The check usually occurs several minutes after the instance starts. Make sure that there are no pending Microsoft updates, and that the computer is not set to install software when it reboots.
* Apply the following hotfixes:
  1. [You cannot change system time if RealTimeIsUniversal registry entry is enabled in Windows](http://support.microsoft.com/kb/2922223)
  2. [High CPU usage during DST changeover in Windows Server 2008, Windows 7, or Windows Server 2008 R2](http://support.microsoft.com/kb/2800213)
* Enable the RealTimeIsUniversal registry.

**Linux**

* Enable Secure Shell (SSH) for remote access.
* Make sure that your host firewall (such as Linux iptables) allows access to SSH. Otherwise, you will not be able to access your instance after the import is complete.
* Make sure that you have configured a non-root user to use public key-based SSH to access your instance after it is imported. The use of password-based SSH and root login over SSH are both possible, but not recommended. The use of public keys and a non-root user is recommended because it is more secure. VM Import will not configure an ec2-user account as part of the import process.
* Make sure that your Linux VM uses GRUB (GRUB legacy) or GRUB 2 as its bootloader.
* Make sure that your Linux VM uses a root filesystem is one of the following: EXT2, EXT3, EXT4, Btrfs, JFS, or XFS.

# Export Your VM from Its Virtual Environment

After you have prepared your VM for export, you can export it from your virtualization environment.

**Citrix: Export to OVF**

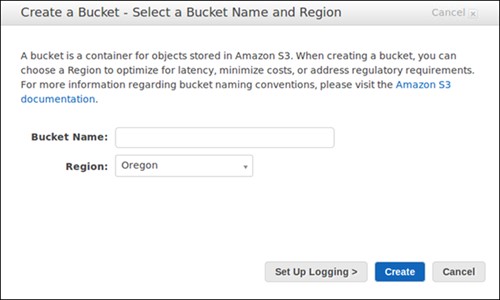
**Microsoft Hyper-V: Export to VHD**

**VMWare: Export to OVF**

**KVM: Convert to Raw Disk**

**Create a Migration bucket**

1. Sign into the AWS Management Console and open the Amazon S3 console at[https://console.aws.amazon.com/s3.](https://console.aws.amazon.com/s3)
2. Click **Create Bucket**.



1. In the **Create a Bucket** dialog box, in the **Bucket Name** box, enter a bucket name and enter Virginia as a Region
2. In the **Region** box, select a region. For this exercise, select **Oregon** from the drop-down list.
3. Click **Create**.

When Amazon S3 successfully creates your bucket, the console displays your empty bucket in the **Buckets** panel.



**Create a vmimport role**

To create the role using AWS CLI, Login in to build server

* Create a file named trust-policy.json as:

#vi trust-policy.json

{

   "Version":"2012-10-17",

   "Statement":[

      {

         "Sid":"",

         "Effect":"Allow",

         "Principal":{

            "Service":"vmie.amazonaws.com"

         },

         "Action":"sts:AssumeRole",

         "Condition":{

            "StringEquals":{

               "sts:ExternalId":"vmimport"

            }

         }

      }

   ]

}

Use AWS CLI, run the **aws iam create-role** command to create the role

aws iam create-role --role-name vmimport --assume-role-policy-document file://trust-policy.json

Create a Policy with name vmimport and attach a role called vmimport

#vi policy.json

{

"Version":"2012-10-17",

   "Statement":[

      {

         "Effect":"Allow",

         "Action":[

            "s3:ListBucket",

            "s3:GetBucketLocation"

         ],

         "Resource":[

            "arn:aws:s3:::<disk-image-file-bucket>"

         ]

      },

      {

         "Effect":"Allow",

         "Action":[

            "s3:GetObject"

         ],

         "Resource":[

            "arn:aws:s3:::<disk-image-file-bucket>/\*"

         ]

      },

      {

         "Effect":"Allow",

         "Action":[

            "ec2:ModifySnapshotAttribute",

            "ec2:CopySnapshot",

            "ec2:RegisterImage",

            "ec2:Describe\*"

         ],

         "Resource":"\*"

      }

   ]

}

Use AWS CLI, run the **aws iam put-role-policy** command to attach the policy to the role created:

aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document file://role-policy.json

**SCP the root volume and Data volume to Build server**

This is one approach I found good in slow internet, In Faster Enterprise enviroinment you can use all the commands in build server in your local enterpeise server/

Open Winscp

Machine generated alternative text:
Login-VVinSCP 
File 叼 「 0t0n0- 
Password: 
Advanced Iv 
Port number 

Get the IP of Build Server

Machine generated alternative text:
'Log •i Mig 
M Inbr Linc Dat M Inbr M Inbr p Free 
C https://console.aws.amazon.com/ec2/v2/home?region— 
—us-east-I *Insta i nstanceld 
e cre 
e Rev 
ii AW 
ii Ste, iivtvl O Mi 
Mohanraj Shanmugam 
Services v 
Launch Instance 
N. Virginia 
Support 
EC2 Dashboard 
Events 
Tags 
Reports 
Limits 
'I INSTANCES 
Instances 
Spot Requests 
Reserved Instances 
Scheduled Instances 
Commands 
Dedicated Hosts 
IS IMAGES 
AMIS 
aundle Tasks 
ELASTIC BLOCK STORE 
Volumes 
Snapshots 
'I NETWORK & SECURITY 
Security Groups 
Connect 
Actions v 
Filter by tags and attributes or search by keyword 
Name 
build server 
Instance ID 
i-316644ac 
i-dee4b859 
i-ecf53d76 
Instance Type 
t2.micro 
t2.micro 
t2.small 
Availability Zone 
us-east-Id 
us-east-I a 
us-east-lb 
Instance State 
running 
running 
stopped 
Status Checks 
2/2 checks 
212 checks 
Alarm 
Status 
Ito a ota 
Public DNS 
ec2-54-197-12-194.com_ 
ec2-54-174-207-46.com_ 
Instance: i-316644ac (build server) 
Public DNS: ec2-54-197-12-1S4.compute-1.amazonaws.com 
Description 
Status Checks 
Instance ID 
Instance state 
Instance type 
Private DNS 
Private IPS 
Monitoring 
running 
t2.micro 
Tags 
Secondary private IPS 
ip-172-31-51-174.ec2.internal 
172.31.51.174 
-1 Q7*Qk7n 
Public DNS 
Public IP 
Elastic IP 
Availability zone 
Security groups 
Scheduled events 
ec2-54-197-12-194.compute- 
1.amazonaws.com 
54.197.1219 
us-east-Id 
launch-wizard-I . view rules 
No scheduled events 
Q Feedback English 
AWS_SimpIe Icons P ...zip I • ent303-1311192106....pdf 
@2008 
- 2016, Amazon Internet Services Private Ltd or its aflliates All rights reserved 
Privacy Policy Terms of Use 
Show all downloads... X 
803 PM 
5/12/2016 

Machine generated alternative text:
Login-VVinSCP 
Host name 
File 叼 「 0t0n0- 
54-197-12 194 
Password: 
Advanced Iv 
Port number 

Machine generated alternative text:
Environment 
Directories 
Recyde bin 
scp/shell 
Connect on 
Proxy 
Key exchange 
Au then be bon 
Advanced Site Settings 
D Bypass authenbcabon enbrely (SSH-2) 
Au then bcabon options 
@Attempt authenbcabon using Pageant 
Attempt TIS or CryptoCard authenbcabon (SSH-I) 
@Attempt keyboard-interacbve' authenbcabon (SSH-2) 
@Respond with password to the first prompt 
Au then bcabon parameters 
Allon agent forwarding 
Private file: 
D : itest_key ppk 
Attempt GSSAPI authenbcabon (SSH-2) 
Allon GSSAPI credential delegation 

Machine generated alternative text:
ec2-user@54.197.12.194 
Local Mark Files Commands Session Options 
Synchronize 
ec2- user 
ec2- user 
4/19/2016 PM 
4/12/2016 PM 
4/20/2016 PM 
3/22/2016 43601 PM 
3/27/2016 i14•28PM 
4/10/2016 PM 
4/24/2016 431•14 PM 
2/27/2016 PM 
3/24/2016 7:2657 PM 
Remote Help 
Queue • Transfer Settings 
Default 
Documents 
Changed 
WinSCP 
ec2-user@i4.1g7.12.194 Newsession 
My documents 
Upload Edit X Properties 
C: \ Documents 
Name 
Custom Office Templates 
Kite matic 
My CamStudio Temp Files 
My CamStudio Videos 
My Shapes 
OneNote Notebooks 
Outlook Files 
prods oft 
Scanned Documents 
Virtual Machines 
webapp 
02016 03-22 16.00 Mohanraj s Meeting.g2m 
02016-03-27 13.32 Meet Now.g2m 
02016-04-10 10.15 Meet Now.g2m 
02016-04-24 1047 Meet Now.g2m 
02016-04-29 0825 Meet Now.g2m 
Amazon EC2 Container Service Deployment Walkthroug... 
$JAZ and Host Aggregrators.docx 
Business Insider Enables Continuous Delivery with Docke... 
CentOS 64-bit.vmdk 
na 7717 
08 of g,835 MB in Oof 66 
ec2-user 
Download 
/ ho me/ec2- use r 
Name 
container.json 
my-test.vhd 
testl .cn.'a 
testl .vmdk 
Etestl zip 
Ey2ycent.zip 
MB in Oof6 
Edit 
Size 
178 KB 
274,432 KB 
52$341 KB 
412671 KB 
14,333 KB 
16 KB 
14,653 KB 
758062... 
Type 
Parent directory 
File folder 
File folder 
File folder 
File folder 
File folder 
File folder 
File folder 
File folder 
File folder 
File folder 
File folder 
File folder 
File folder 
GoToMeeting Acti... 
GoToMeeting Actin. 
GoToMeeting Actin. 
GoToMeeting Acti... 
GoToMeeting Actin. 
VLC media file 
Microsoft Word 
VLC media file (.m... 
Security Certificate 
Virtual Machine Di... 
Rirh 
Size 
397,618 KB 
987,968 KB 
38-9,753 KB 
422117 KB 
Changed 
5/11/2016 PM 
5/12/2016 44850 AM 
5/12/201681406 AM 
5/11/201660218 PM 
5/11/2016 PM 
5/11/2016 PM 
5/12/2016 AM 
Find Files 
Rights 
rw-xr-xr-x 
rw-rw-r-- 
rw-rw-r-- 
Owner 
5/11/2016 PM 
1/19/2016 64824 PM 
12/29/2015 
3/29/2016 PM 
1/26/2016 
1/26/2016 ID3&14AM 
,'3/2016 PM 
12/21/2015 AM 
5/12/2016 PM 
2/25/2016 
5/12/2016 PM 
4/29/2016 
3/11/2016 60421 AM 
1/4/2016 PM 
3/29/2016 62026 AM 
s.1d.77 0M 
6 hidden 
ec2 
ec2 
ec2 
ec2 
-user 
-user 
-user 
-user 
7 hidden 

# **Copy the Volume from AWS Build Server to Build S3 Bucket**

Copy both Root volume and Data volume

$ aws s3 cp <filename.vhd> s3://<bucket name>/<file name.vhd>

# **Importing Your VM into Amazon EC2**

#vi containers.json

[{

"Description": "*First CLI task*",

"Format": "*ova*",

"UserBucket": {

"S3Bucket": "*my-import-bucket*",

"S3Key": "*my-windows-2008-vm.ova*"

}

}]

Import the Root disk as AMI

$ **aws ec2 import-image --description "example\_com\_web\_Server" --disk-containers file://containers.json**

Output “ Note the Import Task ID”

<ImportImageResponse xmlns="http://ec2.amazonaws.com/doc/2015-03-01/">

<progress>2</progress>

<importTaskId>import-ami-fgxn195v</importTaskId>

<status>active</status>

<description>Windows 2008 OVA</description>

<snapshotTaskDetailSet>

<item>

<diskImageSize>0.0</diskImageSize>

<userBucket>

<s3Bucket>*my-import-bucket*</s3Bucket>

<s3Key>*my-windows-2008-vm.ova*</s3Key>

</userBucket>

</item>

</snapshotTaskDetailSet>

<licenseType>AWS</licenseType>

<statusMessage>pending</statusMessage>

<requestId>1571e127-d6d8-4984-b4f1-3a21e9dbdcb5</requestId>

</ImportImageResponse>

**To check the status of your import task**

**aws ec2 describe-import-image-tasks --cli-input-json "{ \"ImportTaskIds\": [\"import-ami-fgxn195v\"], \"NextToken\": \"abc\", \"MaxResults\": 10 } "**

Output

<DescribeImportImageTasksResponse xmlns="http://ec2.amazonaws.com/doc/2015-03-01/">

<importImageTaskSet>

<item>

<platform>Windows</platform>

<importTaskId>import-ami-fgs8im0c</importTaskId>

<imageId>ami-4a6c2722</imageId>

<status>completed</status>

<description>Linux OVA</description>

<architecture>x86\_64</architecture>

<snapshotTaskDetailSet>

<item>

<diskImageSize>3.115815424E9</diskImageSize>

<deviceName>/dev/sda1</deviceName>

<description>First CLI task</description>

<format>VMDK</format>

<url>https://mys3bucket/vms/my-linux-vm.ova?AWSAccessKeyId=myAccessKeyId&Expires=expirationDate&Signature=mySignature</url>

</item>

</snapshotTaskDetailSet>

<licenseType>AWS</licenseType>

</item>

</importImageTaskSet>

<requestId>377ec1ca-6a47-42f5-8b84-aa07ff87f7b0</requestId>

</DescribeImportImageTasksResponse>

**Import Data VM disk to a Snapshot**

#vi containers2.json

{

"Description": "*First CLI task*",

"Format": "*ova*",

"UserBucket": {

"S3Bucket": "*my-import-bucket*",

"S3Key": "*my-windows-2008-vm.ova*"

}

}

Create snapshot

**aws ec2 import-snapshot --description "Data Disk" --disk-container file://containers2.json**

Output

<ImportSnapshotResponse xmlns="http://ec2.amazonaws.com/doc/2015-03-01/">

<snapshotTaskDetail>

<diskImageSize>0.0</diskImageSize>

<progress>3</progress>

<status>active</status>

<description>Windows 2008 VMDK</description>

<url>https://mys3bucket/vms/Win\_2008\_Server\_Enterprise\_R2\_64-bit.vmdk?AWSAccessKeyId=myaccesskey&Expires=expirationdate&Signature=signature\</url>

<statusMessage>pending</statusMessage>

</snapshotTaskDetail>

<importTaskId>import-snap-ffy5pvea</importTaskId>

<description>Windows 2008 VMDK</description>

<requestId>2ef5652d-6816-4c20-89b2-a4bbb0560190</requestId>

</ImportSnapshotResponse>

Verify the completion “ Replace the import Task id”

**aws ec2 describe-import-snapshot-tasks --cli-input-json "{ \"ImportTaskIds\": [\"import-snap-fgr1mmg7\"], \"NextToken\": \"abc\", \"MaxResults\": 10 } "**

**Create a auto scaling group**

**To create a launch configuration and Auto Scaling group from the launch wizard**

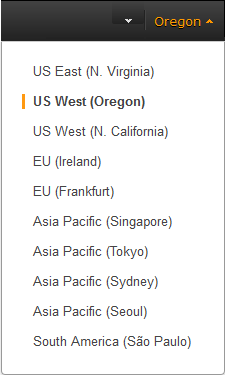
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the dashboard, choose **Launch Instance**.
3. Choose an AMI created through migration, then choose an instance type on the next page, and then choose **Next: Configure Instance Details**.
4. In **Number of instances**, enter the number of instances that you want to launch, and then choose **Launch into Auto Scaling Group**. You do not need to enter any other configuration details on the page.
5. In the confirmation dialog box, choose **Create Launch Configuration**.
6. You are switched to step 3 of the launch configuration wizard. The AMI and instance type are already selected based on the selection you made in the Amazon EC2 launch wizard. Enter a name for the launch configuration, configure any other settings as required, and then choose **Next: Add Storage**.
7. Add Data volume and create from the snapshot created from Data snapshot
8. Configure any additional volumes, and then choose **Next: Configure Security Group**.

Enable port 80 in security group

1. Create a new security group, or choose an existing group, and then choose **Review**.
2. Review the details of the launch configuration, and then choose **Create launch configuration** to choose a key pair and create the launch configuration.
3. in the **Configure Auto Scaling group details** page, the launch configuration you created is already selected for you, and the number of instances you specified in the Amazon EC2 launch wizard is populated for **Group size**. Enter a name for the group, specify a VPC and subnet (if required), and then choose **Next: Configure scaling policies**.
4. In the **Configure scaling policies** page, choose one of the following options, and then choose **Review**:
   * To automatically adjust the size of the Auto Scaling group based on criteria that you specify, choose **Use scaling policies to adjust the capacity of this group** and follow the directions
5. On the **Review** page, you can optionally add tags or notifications, and edit other configuration details. When you are done, choose **Create Auto Scaling group**.

**To access Elastic Load Balancing using the Amazon EC2 console**

1. Sign in to the AWS Management Console and open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select a region. Each load balancer is tied to the region in which you create it. You can select any region that's available to you, regardless of your location.



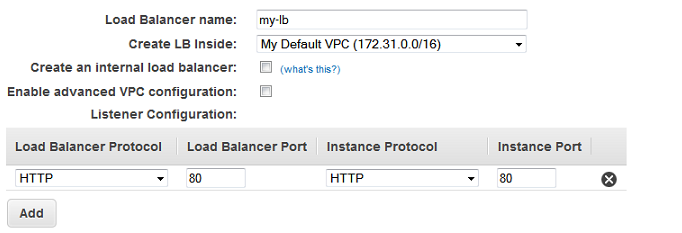
1. In the navigation pane, under **LOAD BALANCING**, click **Load Balancers**.

**To define your load balancer**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select a region for your load balancers. Be sure to select the same region that you selected for your EC2 instances.
3. In the navigation pane, under **LOAD BALANCING**, click **Load Balancers**.
4. Click **Create Load Balancer**.
5. In **Load Balancer name**, enter a name for your load balancer.

The name of your load balancer must be unique within your set of load balancers for the region, can have a maximum of 32 characters, and can contain only alphanumeric characters and hyphens.

1. From **Create LB inside**, select the same network that you selected for your instances: EC2-Classic or a specific VPC.
2. [Default VPC] If you selected a default VPC and would like to choose the subnets for your load balancer, select **Enable advanced VPC configuration**.
3. Leave the default listener configuration.

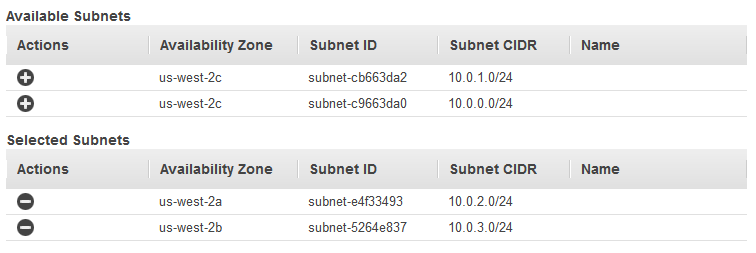


1. [EC2-VPC] Under **Select Subnets**, select at least one available public subnet. To improve the availability of your load balancer, select more than one public subnet.

**Note**

If you selected EC2-Classic as your network, or you have a default VPC but did not select **Enable advanced VPC configuration**, you do not see **Select Subnets**.

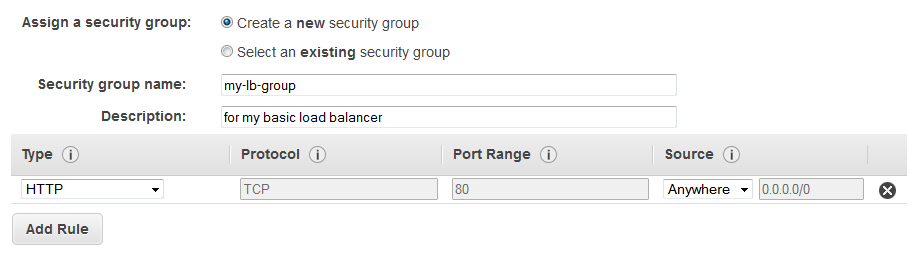
The available subnets for the VPC for your load balancer are displayed under **Available Subnets**. Select public subnets that are in the same Availability Zones as your instances. Click the icon in the **Action** column for each subnet to attach. These subnets are moved under **Selected Subnets**. You can select at most one subnet per Availability Zone. If you select a subnet from an Availability Zone where there is already a selected subnet, this subnet replaces the currently selected subnet for the Availability Zone.



1. Click **Next: Assign Security Groups**.

**To assign security group to your load balancer**

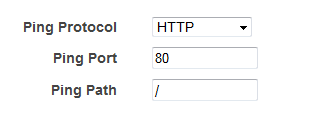
1. On the **Assign Security Groups** page, select **Create a new security group**.
2. Enter a name and description for your security group, or leave the default name and description. This new security group contains a rule that allows traffic to the port that you configured your load balancer to use.



1. Click **Next: Configure Security Settings**.

**To configure health checks for your instances**

1. On the **Configure Health Check** page, do the following:
   1. Leave **Ping Protocol** set to its default value, HTTP.
   2. Leave **Ping Port** set to its default value, 80.
   3. In the **Ping Path** field, replace the default value with a single forward slash ("/"). This tells Elastic Load Balancing to send health check queries to the default home page for your web server, such as index.html or default.html.



* 1. Leave the other fields set to their default values.

## Add a Load Balancer to auto scaling group Using the Console

Use the following procedure to attach a load balancer to your Auto Scaling group.

**To attach a load balancer to a group**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under Auto Scaling, click Auto Scaling Groups.
3. Select your group.
4. In the bottom pane, on the Details tab, click Edit.
5. In Load Balancers, select the load balancer.
6. Click Save.

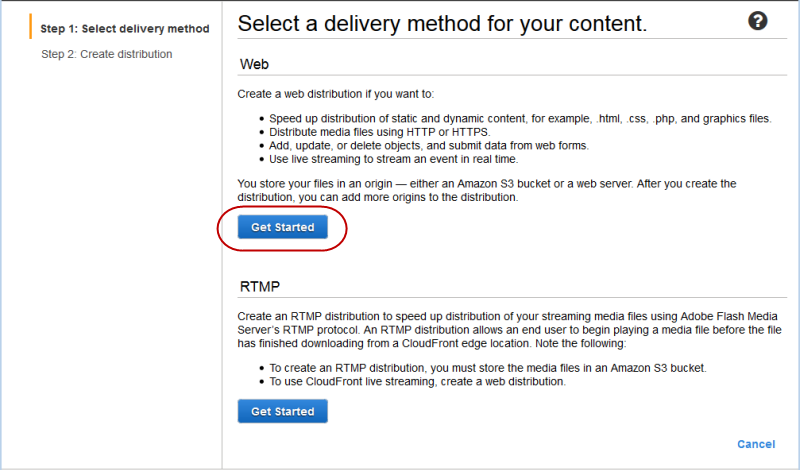
**Copy website images to Image Bucket and edit the code**

1. Click Create Bucket.
2. In the Create a Bucket dialog box, in the Bucket Name box, enter a bucket name. The bucket name you choose must be unique across all existing bucket names in Amazon S3. ...
3. In the Region box, select a region. For this exercise, select Virginia from the drop-down list. ...
4. Click Create.
5. Copy website images to the bucket
6. Do the code changes to the website code to point to S3 URLs for the images

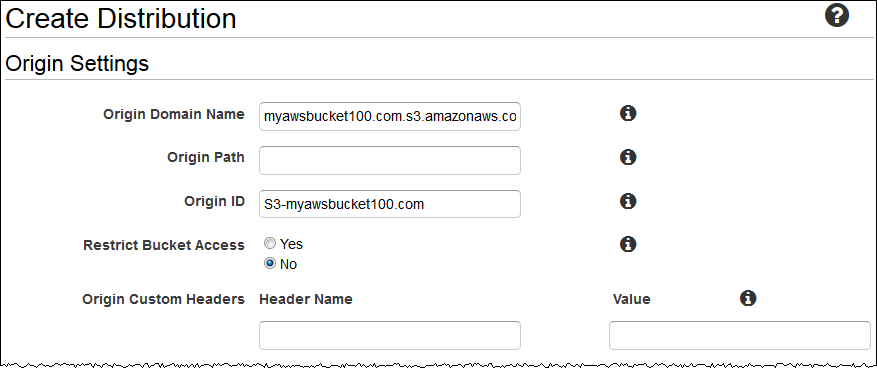
**Create the Cloud Front distribution**

**To create a CloudFront web distribution**

1. Open the CloudFront console at <https://console.aws.amazon.com/cloudfront/>.
2. Choose **Create Distribution**.
3. On the **Select a delivery method for your content** page, in the **Web** section, choose **Get Started**.

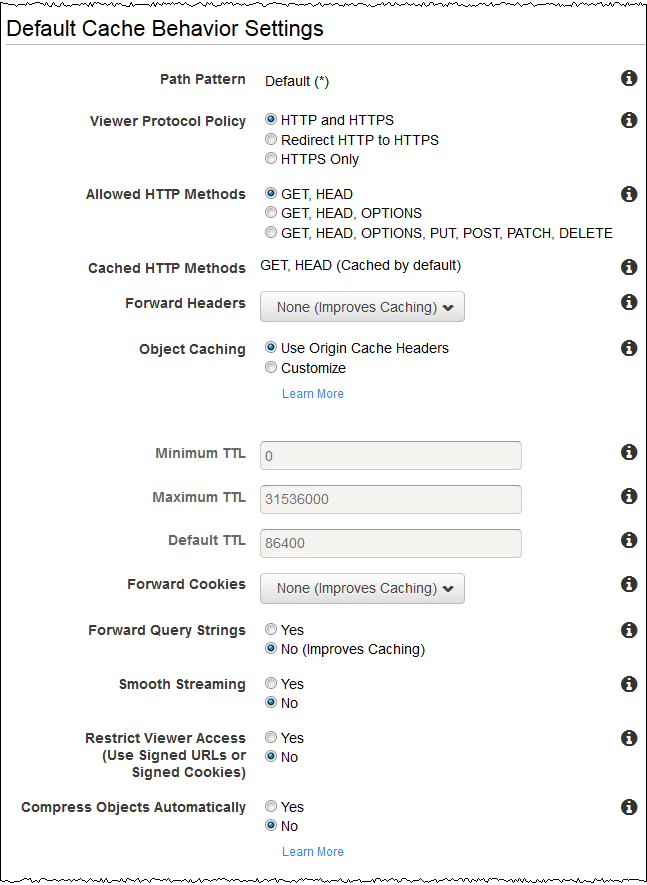


1. On the **Create Distribution** page, under **Origin Settings**, choose the Amazon S3 bucket that you created earlier. For **Origin ID**, **Origin Path**, **Restrict Bucket Access**, and **Origin Custom Headers**, accept the default values.



1. Under **Default Cache Behavior Settings**, accept the default values, and CloudFront will:
   * Forward all requests that use the CloudFront URL for your distribution (for example, http://d111111abcdef8.cloudfront.net/image.jpg) to the Amazon S3 bucket that you specified in Step 4.
   * Allow end users to use either HTTP or HTTPS to access your objects.
   * Respond to requests for your objects.
   * Cache your objects at CloudFront edge locations for 24 hours.
   * Forward only the default request headers to your origin and not cache your objects based on the values in the headers.
   * Exclude cookies and query string parameters, if any, when forwarding requests for objects to your origin. (Amazon S3 doesn't process cookies and processes only a limited set of query string parameters.)
   * Not be configured to distribute media files in the Microsoft Smooth Streaming format.
   * Allow everyone to view your content.
   * Not automatically compress your content.

For more information about cache behavior options, see [Cache Behavior Settings](http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/distribution-web-values-specify.html#DownloadDistValuesCacheBehavior).



1. Under **Distribution Settings**, enter the applicable values:

**Price Class**

Select the price class that corresponds with the maximum price that you want to pay for CloudFront service. By default, CloudFront serves your objects from edge locations in all CloudFront regions.

For more information about price classes and about how your choice of price class affects CloudFront performance for your distribution, go to [Choosing the Price Class for a CloudFront Distribution](http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/PriceClass.html). For information about CloudFront pricing, including how price classes map to CloudFront regions, go to [Amazon CloudFront Pricing](http://aws.amazon.com/cloudfront/pricing/).

**AWS WAF Web ACL**

If you want to use AWS WAF to allow or block HTTP and HTTPS requests based on criteria that you specify, choose the web ACL to associate with this distribution. For more information about AWS WAF, see the [AWS WAF Developer Guide](http://docs.aws.amazon.com/waf/latest/developerguide/).

**Alternate Domain Names (CNAMEs) (Optional)**

Specify one or more domain names that you want to use for URLs for your objects instead of the domain name that CloudFront assigns when you create your distribution. For example, if you want the URL for the object:

/images/image.jpg

to look like this:

http://www.example.com/images/image.jpg

instead of like this:

http://d111111abcdef8.cloudfront.net/images/image.jpg

you would create a CNAME for www.example.com.

**Important**

If you add a CNAME for www.example.com to your distribution, you also need to create (or update) a CNAME record with your DNS service to route queries for www.example.com to d111111abcdef8.cloudfront.net. You must have permission to create a CNAME record with the DNS service provider for the domain. Typically, this means that you own the domain, but you may also be developing an application for the domain owner. For more information about CNAMEs, see [Using Alternate Domain Names (CNAMEs)](http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/CNAMEs.html).

For the current limit on the number of alternate domain names that you can add to a distribution, see [Amazon CloudFront Limits](http://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html#limits_cloudfront) in the *Amazon Web Services General Reference*. To request a higher limit, go to <https://console.aws.amazon.com/support/home#/case/create?issueType=service-limit-increase&limitType=service-code-cloudfront-distributions>.

**SSL Certificate**

Accept the default value, **Default CloudFront Certificate**.

**Default Root Object (Optional)**

The object that you want CloudFront to request from your origin (for example, index.html) when a viewer requests the root URL of your distribution (http://www.example.com/) instead of an object in your distribution (http://www.example.com/product-description.html). Specifying a default root object avoids exposing the contents of your distribution.

**Logging (Optional)**

If you want CloudFront to log information about each request for an object and store the log files in an Amazon S3 bucket, select **On**, and specify the bucket and an optional prefix for the names of the log files. There is no extra charge to enable logging, but you accrue the usual Amazon S3 charges for storing and accessing the files. CloudFront doesn't delete the logs automatically, but you can delete them at any time.

**Cookie Logging**

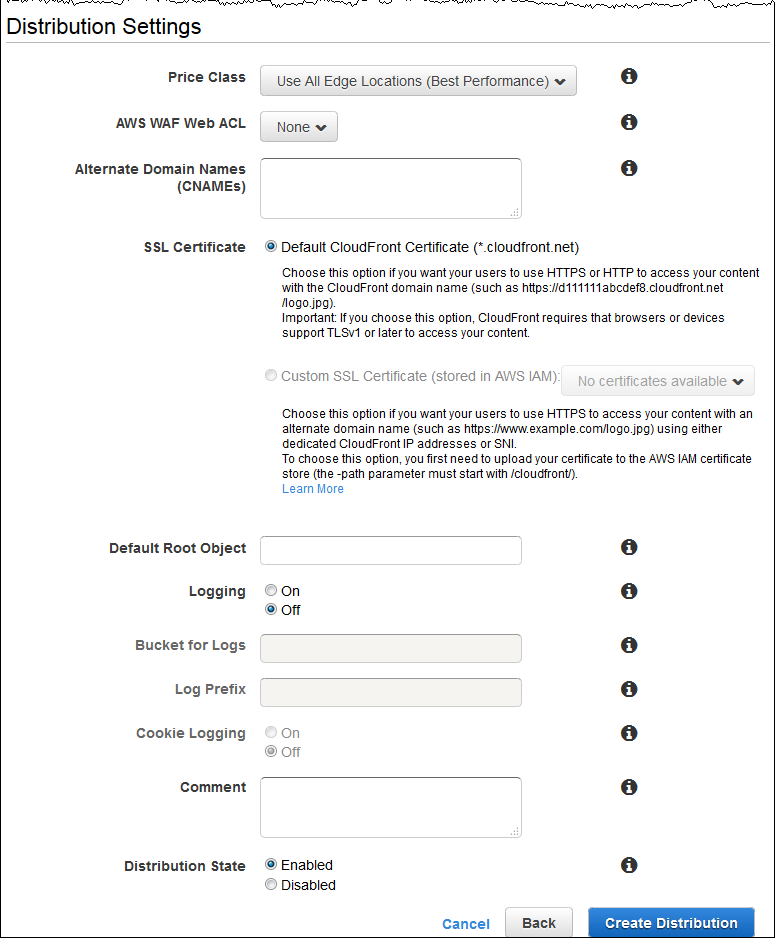
In this example, we're using Amazon S3 as the origin for your objects, and Amazon S3 doesn't process cookies, so we recommend that you select **Off** for the value of **Cookie Logging**.

**Comment (Optional)**

Enter any comments that you want to save with the distribution.

**Distribution State**

Select **Enabled** if you want CloudFront to begin processing requests as soon as the distribution is created, or select **Disabled** if you do not want CloudFront to begin processing requests after the distribution is created.



1. Choose **Create Distribution**.
2. After CloudFront has created your distribution, the value of the **Status** column for your distribution will change from **InProgress** to **Deployed**. If you chose to enable the distribution, it will then be ready to process requests. This should take less than 15 minutes.

The domain name that CloudFront assigns to your distribution appears in the list of distributions. (It also appears on the **General** tab for a selected distribution.)

**If you wanted to move your domain or subdomain to Route53, Create a Route 53 Publicly hosted zone (This is done only in active domain in amazon or for a subdomain)**

**To create a hosted zone using the Amazon Route 53 console**

1. Sign in to the AWS Management Console and open the Amazon Route 53 console at <https://console.aws.amazon.com/route53/>.
2. If you already have a hosted zone for your domain, skip to step 4. If you don't, perform the following steps:
   1. Click **Create Hosted Zone**.
   2. For **Domain Name**, enter the name of your domain.
   3. *Optional:* For **Comment**, enter a comment about the hosted zone.
   4. Click **Create**.
3. On the **Hosted Zones** page, choose the name of the hosted zone in which you want to create resource record sets.
4. Click **Create Record Set**.