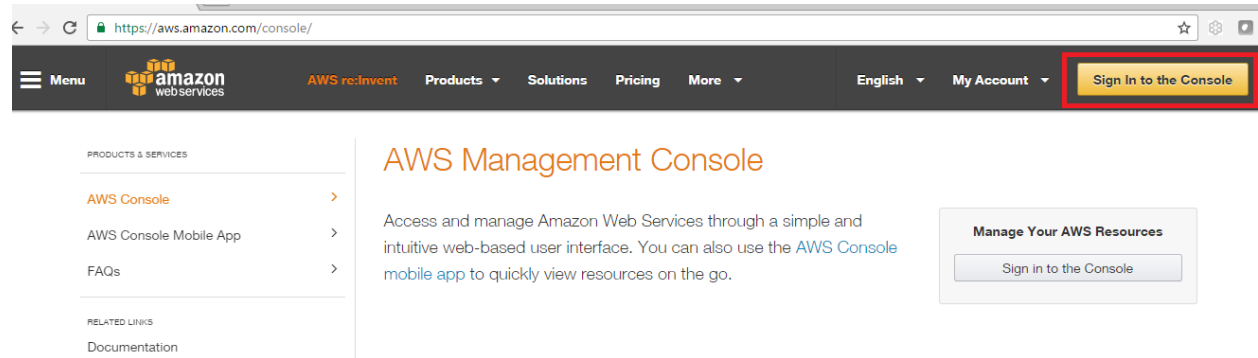


Got to

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

Click on **“Sign In to the Console”**.

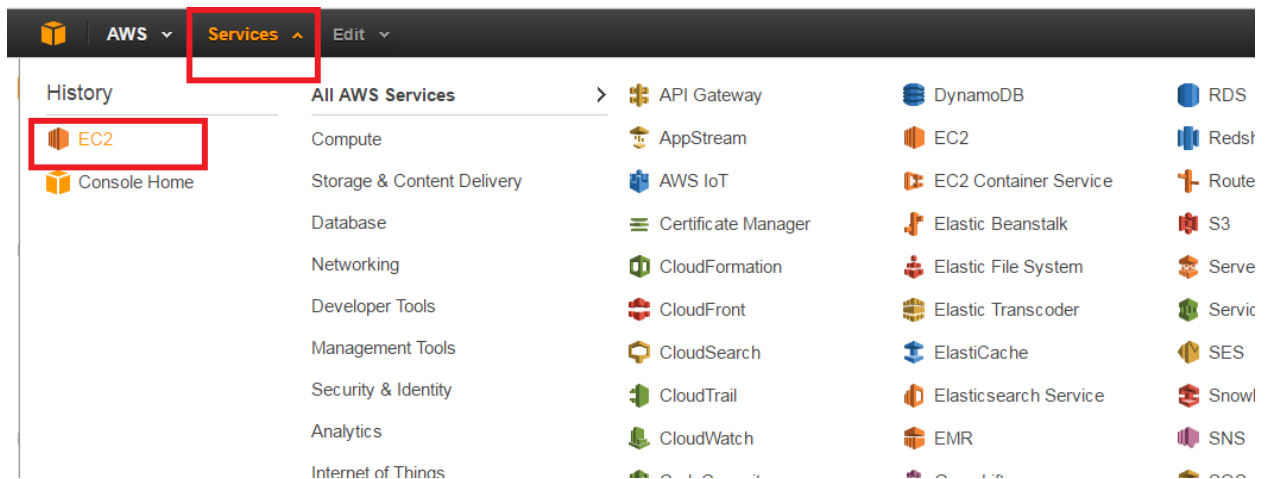


Enter your email and password to login.

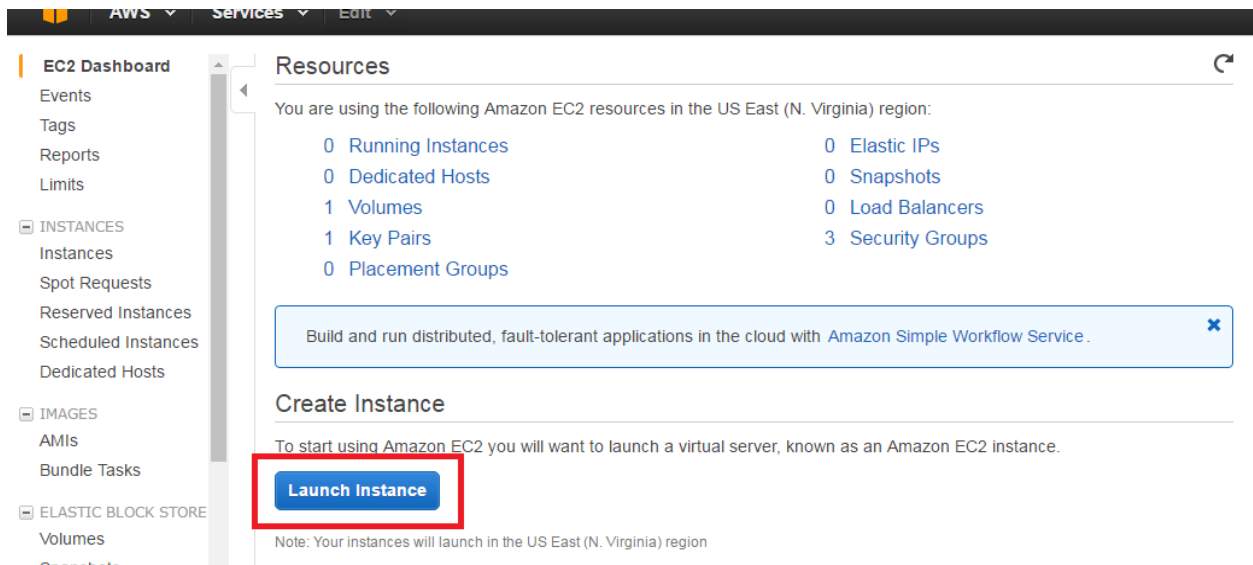
Click on **“I am a new user”** if you don’t have account and then sign up by entering all details.



Click on “Services” and then “EC2”



Click on “Launch Instance”.



Select Ubuntu from different types of instances

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

My AMIs

AWS Marketplace

Community AMIs


☐ Free tier only ⓘ

**Amazon Linux AMI 2016.09.0 (HVM), SSD Volume Type** - ami-b73b63a0 Select

Amazon Linux
Free tier eligible

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.


Root device type: ebs Virtualization type: hvm

**Red Hat Enterprise Linux 7.2 (HVM), SSD Volume Type** - ami-2051294a Select

Red Hat
Free tier eligible

Red Hat Enterprise Linux version 7.2 (HVM), EBS General Purpose (SSD) Volume Type


Root device type: ebs Virtualization type: hvm

**SUSE Linux Enterprise Server 12 SP1 (HVM), SSD Volume Type** - ami-1eeab909 Select

SUSE Linux
Free tier eligible

SUSE Linux Enterprise Server 12 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.

Root device type: ebs Virtualization type: hvm

**Ubuntu Server 16.04 LTS (HVM), SSD Volume Type** - ami-40d28157 Select

Ubuntu
Free tier eligible

Ubuntu Server 16.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/instances>)

64-bit

Select t2.micro instance (which is free) and click on “Review and Launch “

Filter by: All instance types Current generation Show/Hide Columns

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

	Family	Type	vCPUs ⓘ	Memory (GiB)	Instance Storage (GB) ⓘ	EBS-Optimized Available ⓘ	Network Performance ⓘ
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
<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
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	m4.large	2	8	EBS only	Yes	Moderate

Cancel Previous Review and Launch Next: Configure Instance Details

Click on **“Edit Security group”** for different incoming and outgoing requests.(Ex:Http,ssh ,rdp,tcp etc)

AMI Details

Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-40d28157

Free tier eligible

Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
Root Device Type: ebs Virtualization type: hvm

Edit AMI

Instance Type

Edit instance type

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

Security Groups

Edit security groups

Security group name launch-wizard-3

Click on **“Add Rules”**

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:

Create a new security group

Select an existing security group

Security group name:

launch-wizard-3

Description:

launch-wizard-3 created 2016-11-01T12:53:20.790+05:30

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere 0.0.0.0/0

Add Rule

I have added **“All Traffic”** rule, which says machine can accepts all kind of protocols and it accessed from anywhere.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:

Create a new security group

Select an existing security group

Security group name:

launch-wizard-3

Description:

launch-wizard-3 created 2016-11-01T12:53:20.790+05:30

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere 0.0.0.0/0
All traffic	All	0 - 65535	Anywhere 0.0.0.0/0

Add Rule

Click on “Review and Launch”

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	
SSH ▾	TCP	22	Anywhere ▾ 0.0.0.0/0	✕
All traffic ▾	All	0 - 65535	Anywhere ▾ 0.0.0.0/0	✕

Add Rule

Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous **Review and Launch**

Click on “Launch”

▼ Instance Type [Edit instance type](#)

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

▼ Security Groups [Edit security groups](#)

Security group name	launch-wizard-3
Description	launch-wizard-3 created 2016-11-01T12:53:20.790+05:30

Cancel Previous **Launch**

Select “**Create a new key pair**” option from drop down, give name to Key-Pair and Download Key Pair

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name

aws_ubuntu

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue.
Store it in a secure and accessible location. You will not be able to download the file again after it's created.

Click on **“Launch Instances”** which will start Ubuntu instance.


A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name
aws_ubuntu

Download Key Pair



You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel

Launch Instances

To access **Ubuntu** instance from **Windows** we need to convert Key pair **.pem** file to **.ppk** file.

(No need to convert .pem to .ppk if you are accessing from Ubuntu/mac/any linux).

Download PuTTYgen(to convert **.Pem** file to **.ppk**) and PuTTY (accessing Ubuntu instance using ssh)from below ULR

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

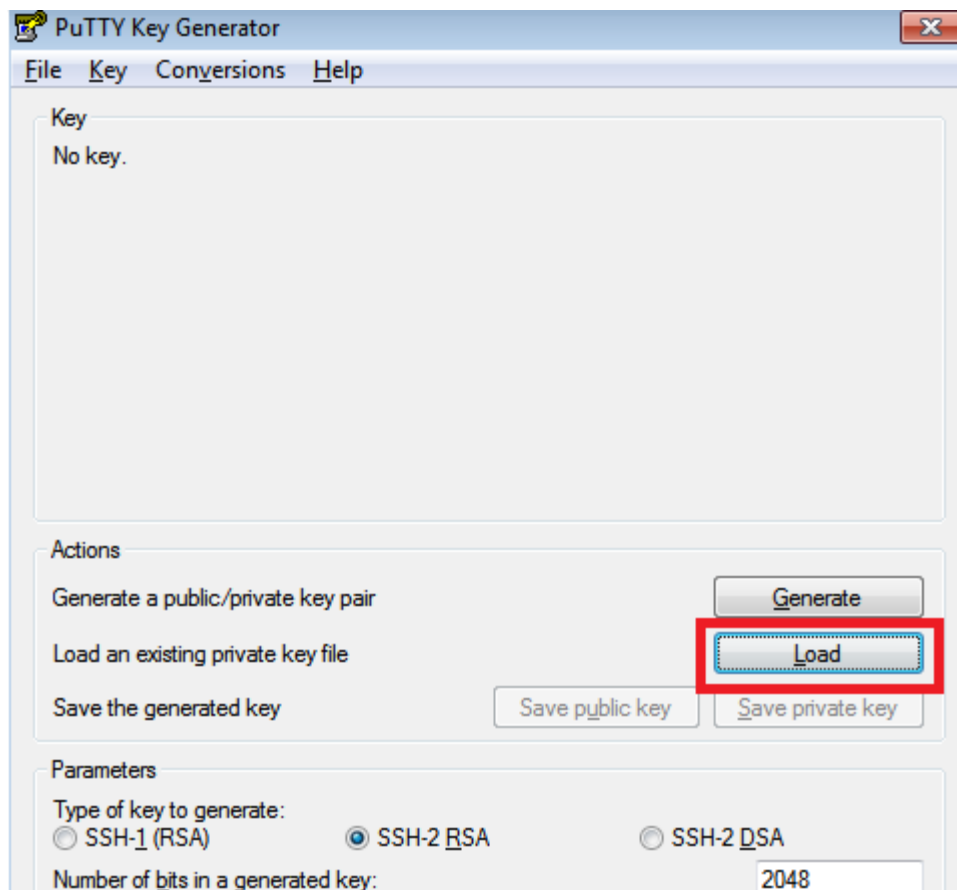
The latest release version (beta 0.67)

This will generally be a version we think is reasonably likely to work well. If you have a problem already fixed the bug, before reporting it.

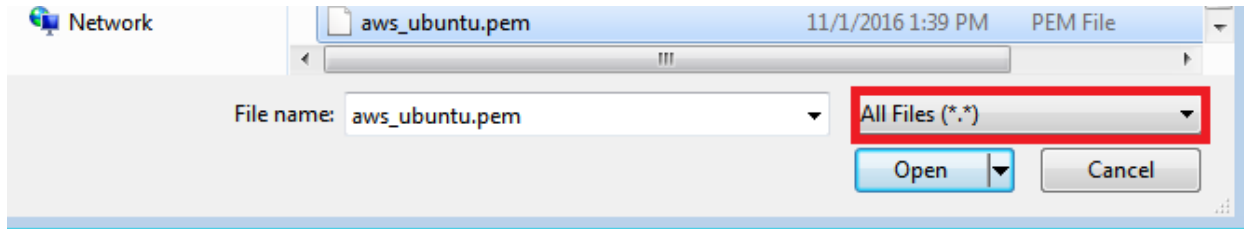
For Windows on Intel x86

PuTTY:	putty.exe	(or by FTP)	(signature)
PuTTYtel:	puttytel.exe	(or by FTP)	(signature)
PSCP:	pscp.exe	(or by FTP)	(signature)
PSFTP:	psftp.exe	(or by FTP)	(signature)
Plink:	plink.exe	(or by FTP)	(signature)
Pageant:	pageant.exe	(or by FTP)	(signature)
PuTTYgen:	puttygen.exe	(or by FTP)	(signature)

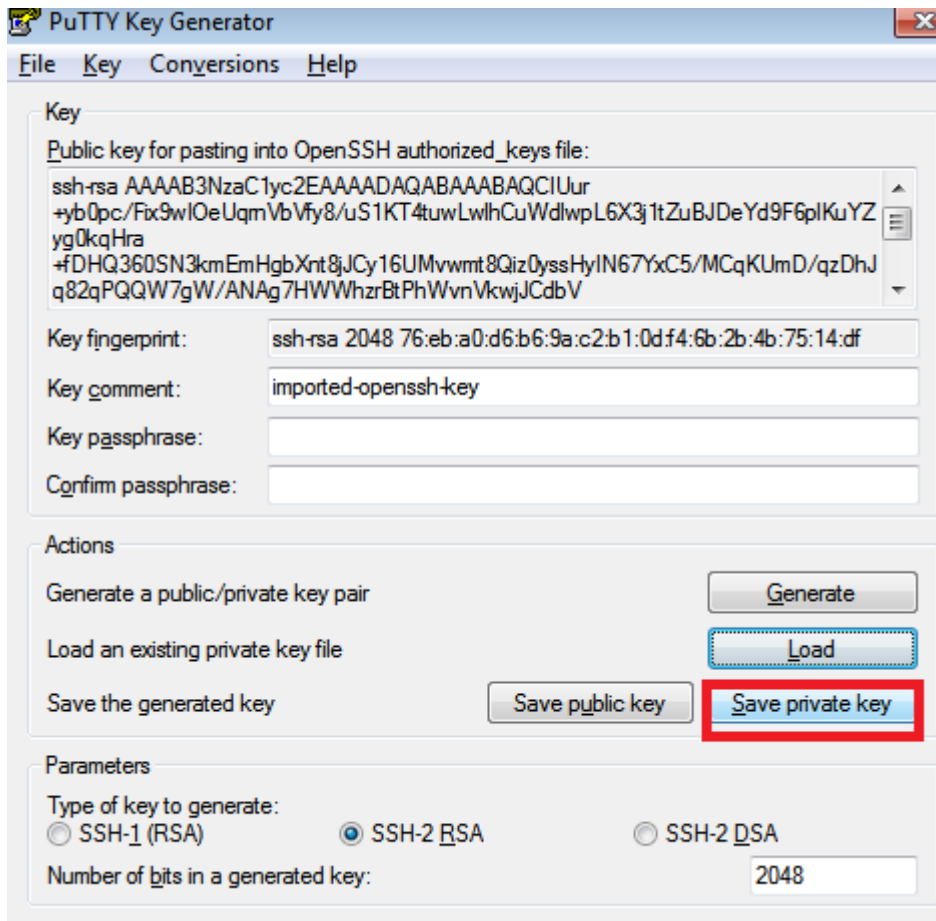
Open PuTTYgen and Click on "Load" to load .pem file.



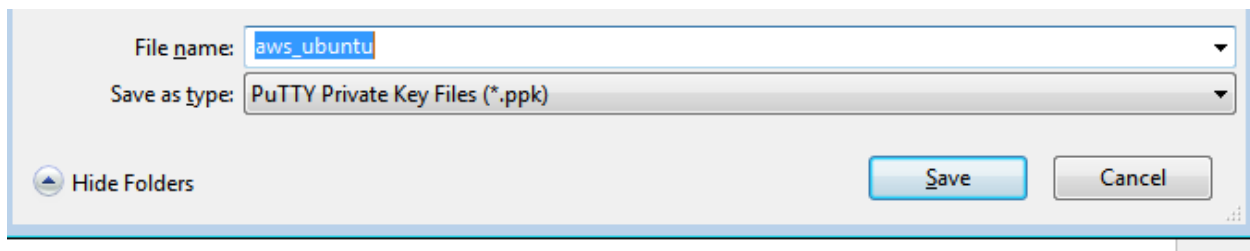
Navigate to **.pem** file location and select **All Files** from option and click on **“Open”**



Click on **“Save Private Key”** and click on Yes.



Enter name and click on “Save”



File name:

Save as type:

Go to AWS console click on “View Instances”.

Launch Status

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. [Find out](#) how to connect to your instances.

▼ Here are some helpful resources to get you started

- [How to connect to your Linux instance](#)
- [Amazon EC2: User Guide](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: Discussion Forum](#)

While your instances are launching you can also

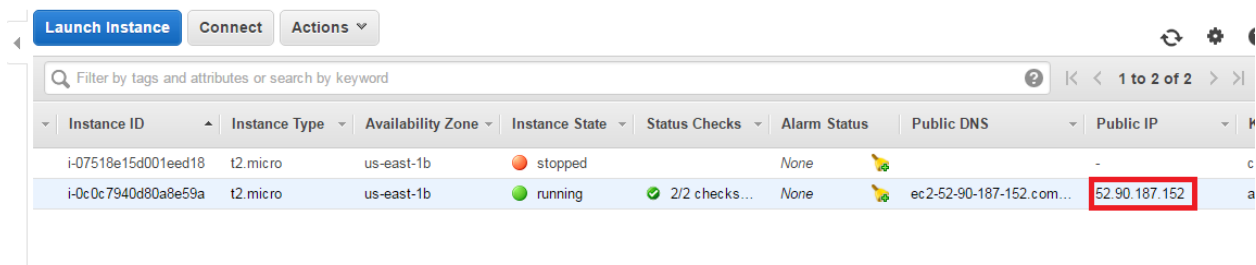
[Create status check alarms](#) to be notified when these instances fail status checks. (Additional charges may apply)

[Create and attach additional EBS volumes](#) (Additional charges may apply)

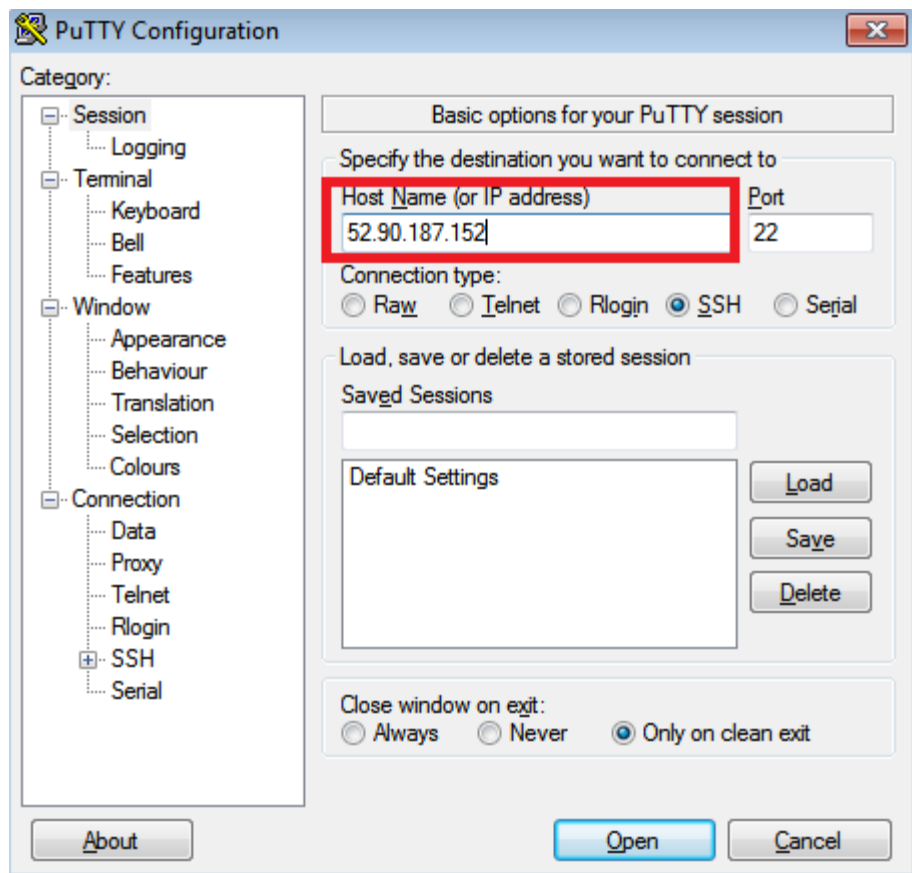
[Manage security groups](#)

[View Instances](#)

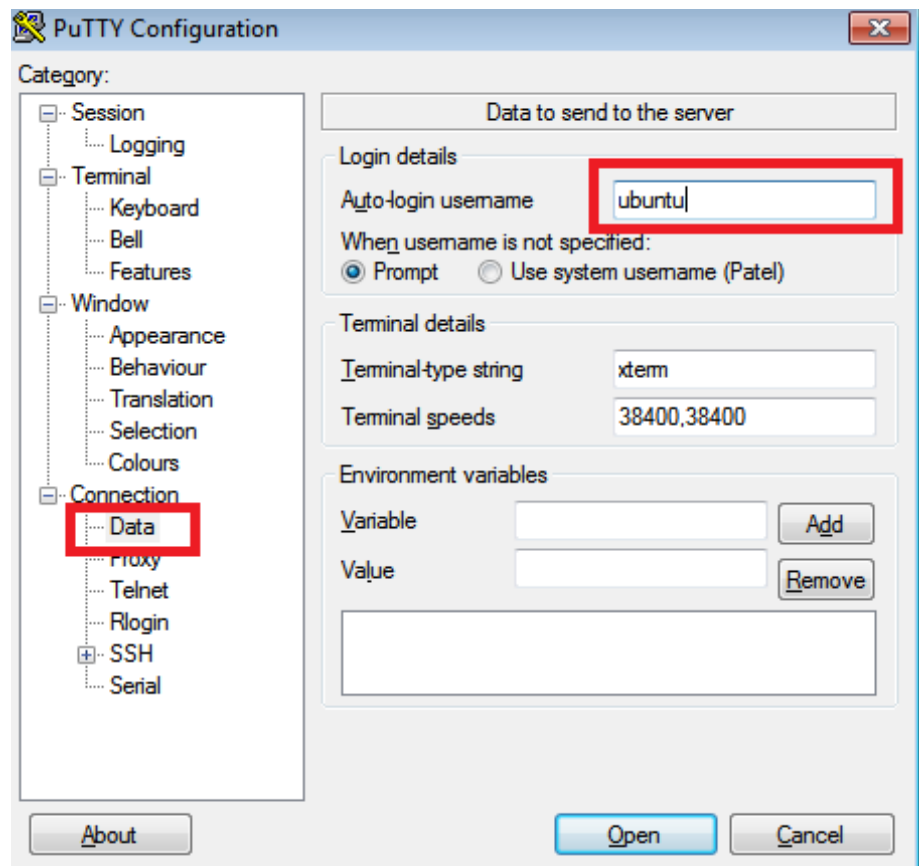
Copy **Public IP** of instance, open **putty** and enter the same.



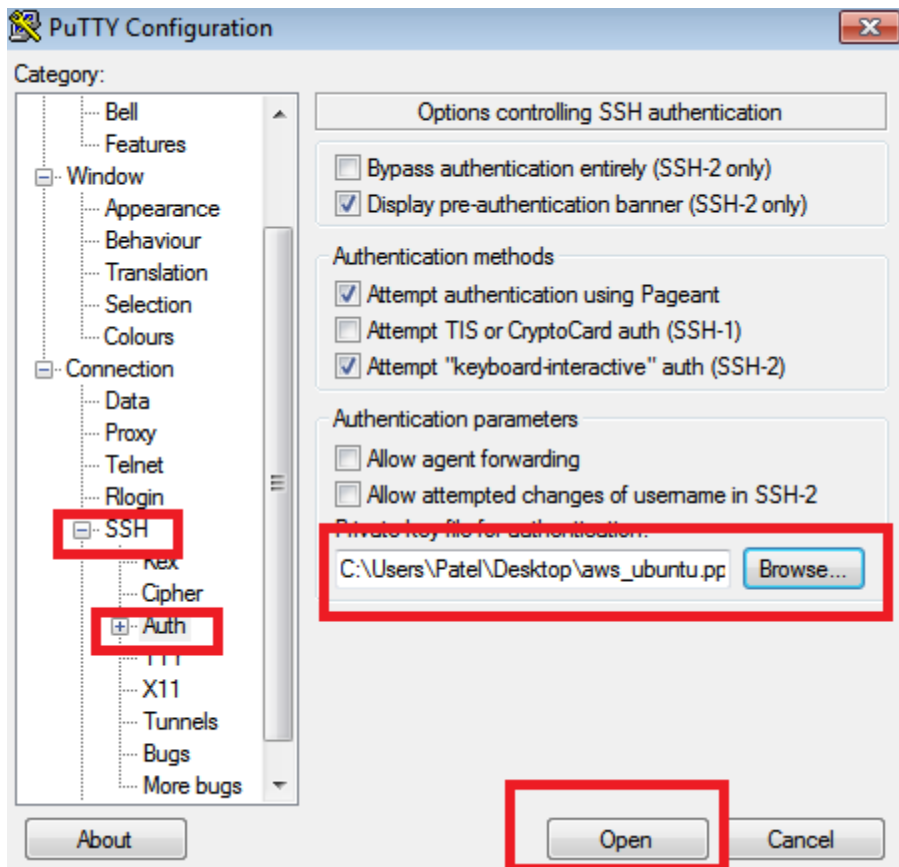
Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
i-07518e15d001eed18	t2.micro	us-east-1b	stopped		None		
i-0c0c7940d80a8e59a	t2.micro	us-east-1b	running	2/2 checks...	None	ec2-52-90-187-152.com...	52.90.187.152



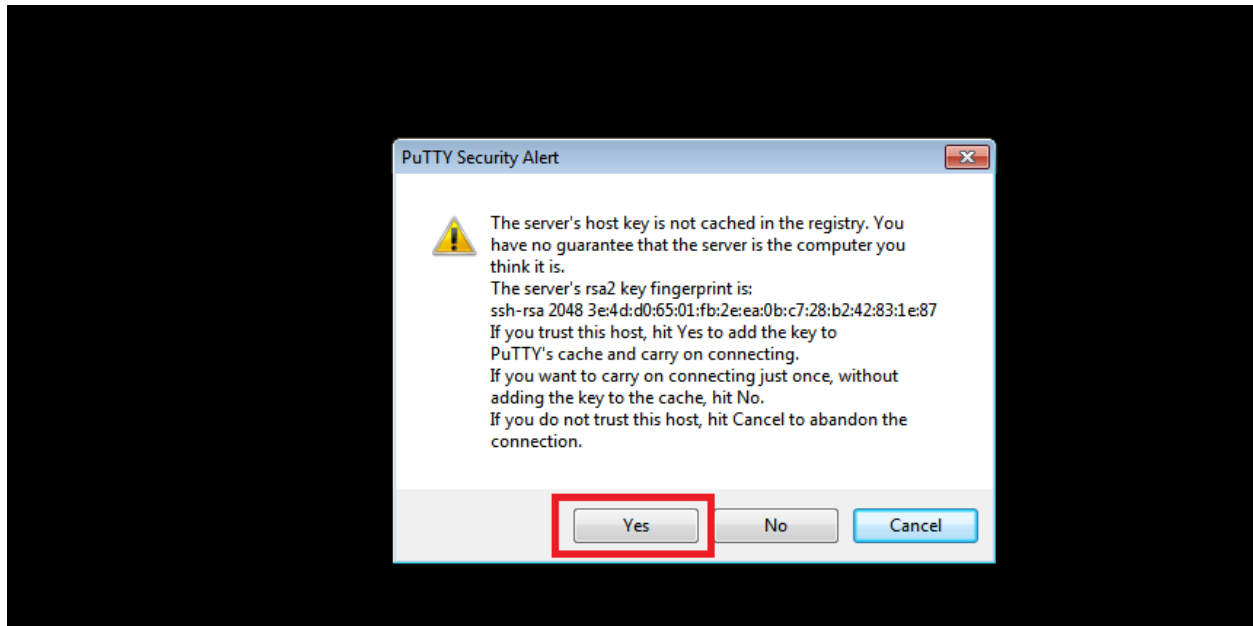
Click on **"Data"** and enter Auto-login username as **"ubuntu"**



Click on **SSH ->Auth** and browse **.ppk** and finally click on **open**.



Click on Yes.



You are ready to use aws Ubuntu free machine .