

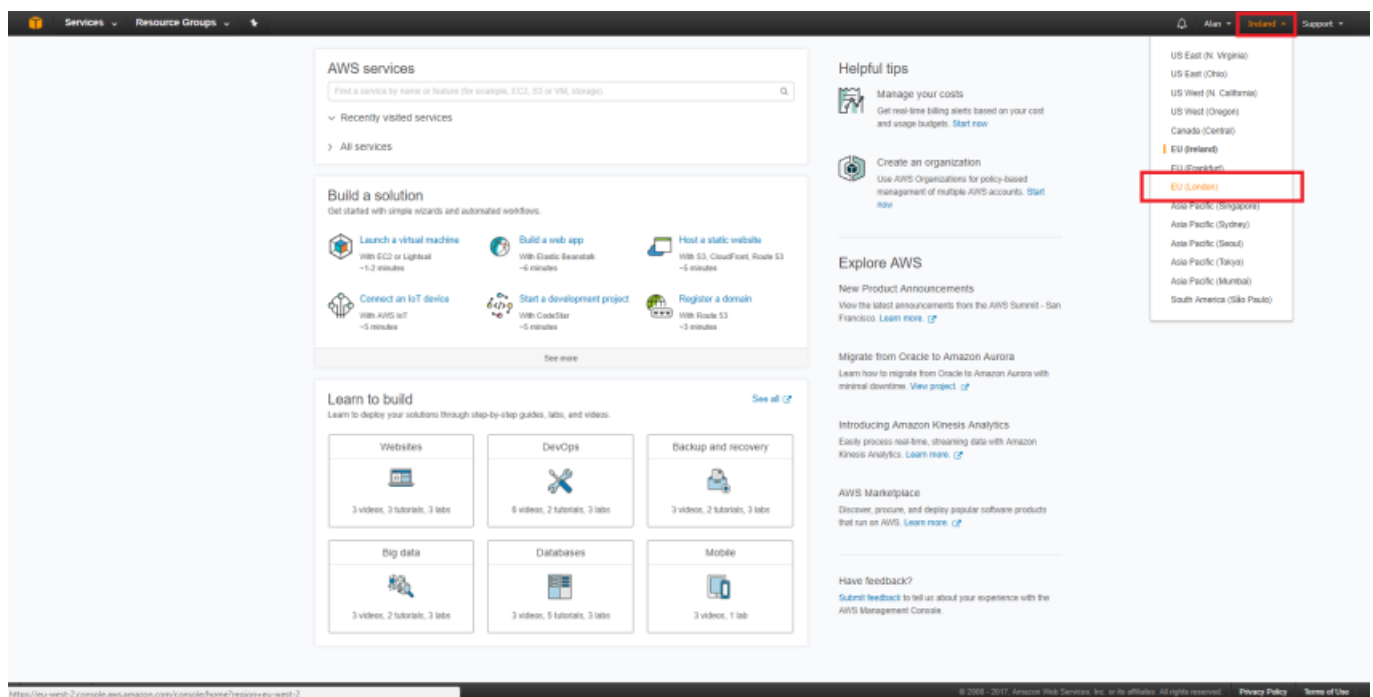
Creating an Amazon AWS EC2 Instance

May 2017

Introduction

This guide assumes that you have created an Amazon account to be able to access the Amazon Web Services (AWS) pages, and that you have successfully logged in.

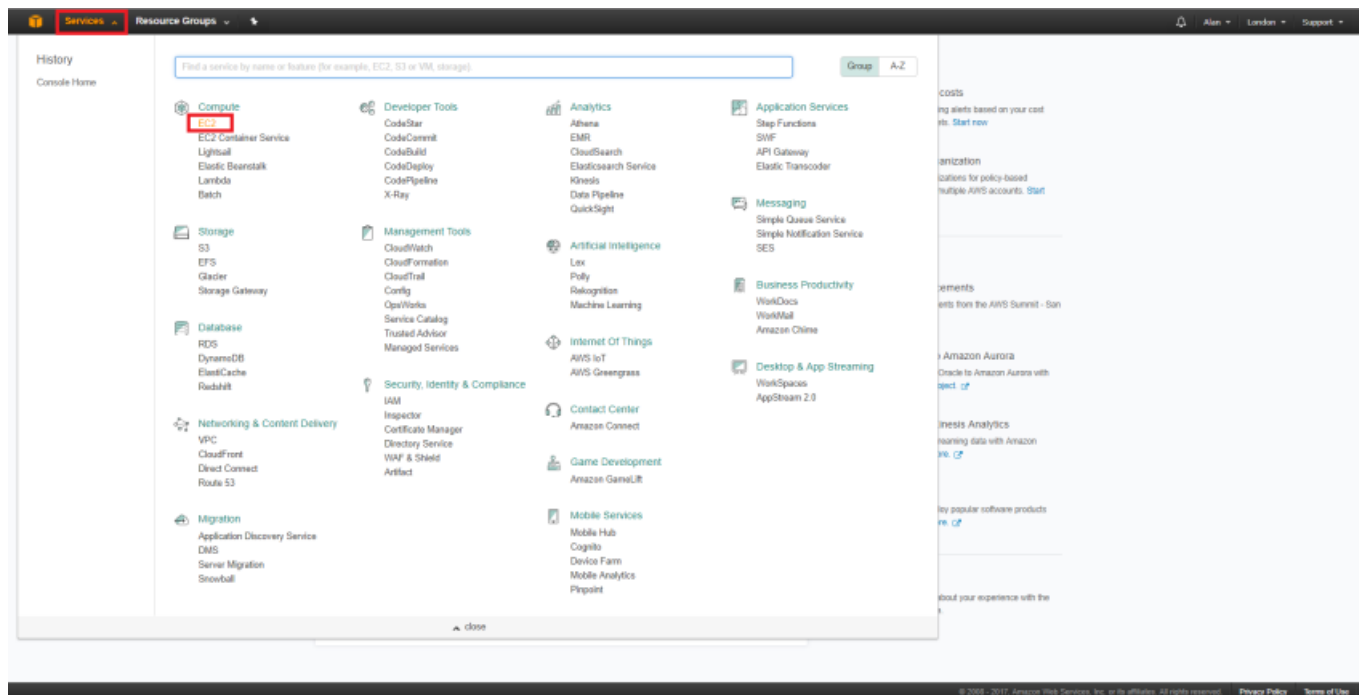
Once logged in you will see the following page (be warned, the site layout changes quite frequently, so you might have to do some digging).



The first mistake I made was not checking the region in which I was creating my AWS instance. So I created mine in the USA. Not that this really matters for serving a few webpages, but if speed is important, ensure you select a region closest to you.

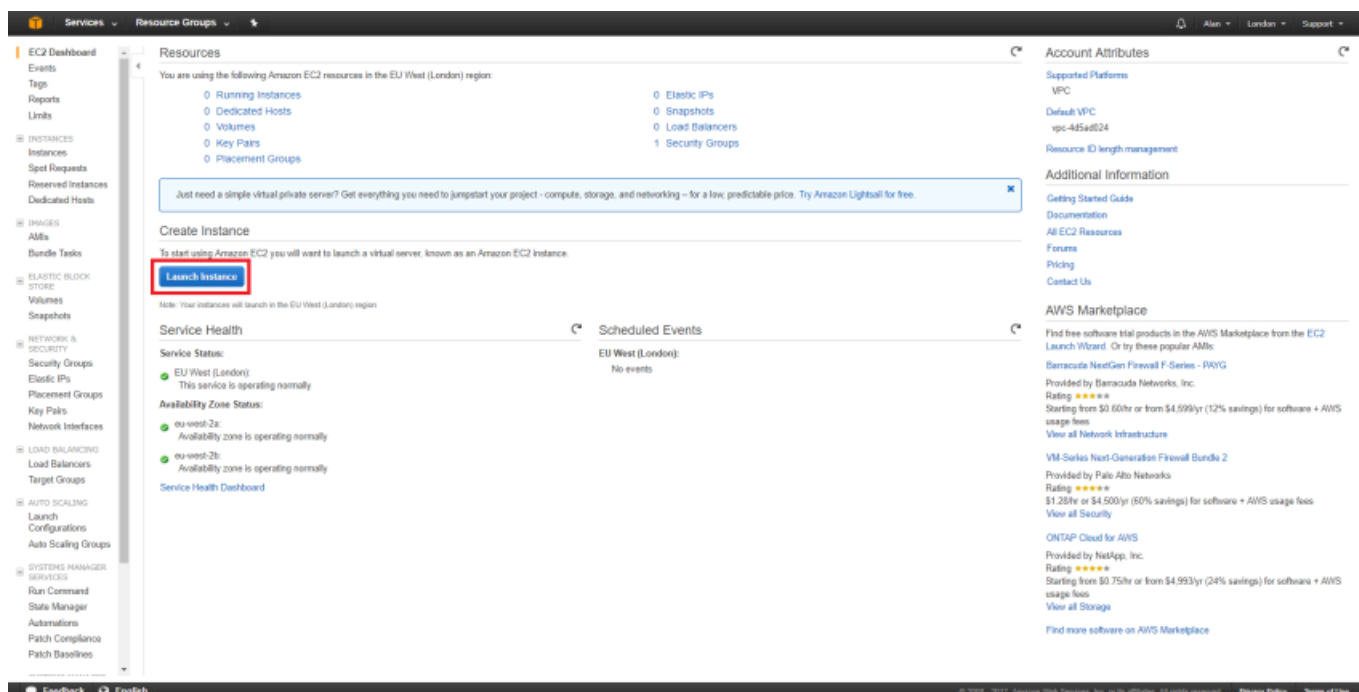
You can do this by clicking the current region in the top right hand corner (highlighted in Orange) and select a suitable local region.

We have to chose what we are going to create.



From the top left of the page, select Services, then select **EC2** from the Compute section.

Now we can select some options and launch our virtual server.



Click on the **Launch Instance** button.

A page with different Operating Systems will open.

Step 1: Choose an Amazon Machine Image (AMI)
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace, or you can select one of your own AMIs.

Quick Start

My AMIs

AWS Marketplace

Community AMIs

☐ Free tier only (1)

Amazon Linux AMI 2017.03.1 (HVM), SSD Volume Type - ami-e0100699
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 **Select** 64-bit

SUSE Linux Enterprise Server 12 SP2 (HVM), SSD Volume Type - ami-a3ca0c0d
SUSE Linux Enterprise Server 12 Service Pack 2 (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 **Select** 64-bit

Red Hat Enterprise Linux 7.3 (HVM), SSD Volume Type - ami-4c0a0f24
Red Hat Enterprise Linux version 7.3 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs Virtualization type: hvm **Select** 64-bit

Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-cc7066a8
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 **Select** 64-bit

Microsoft Windows Server 2016 Base - ami-5a0963ce
Microsoft Windows 2016 Datacenter edition. [English]
Root device type: ebs Virtualization type: hvm **Select** 64-bit

Are you launching a database instance? Try Amazon RDS.
Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale your database on AWS by automating time-consuming database management tasks. With RDS, you can easily deploy Amazon Aurora, MariaDB, MySQL, Oracle, PostgreSQL, and SQL Server databases on AWS. Aurora is a MySQL-compatible, enterprise-class database at 1/10th the cost of commercial databases. [Learn more about RDS](#)
Launch a database using RDS

Microsoft Windows Server 2016 Base with Containers - ami-5c0e9b38
Microsoft Windows 2016 Datacenter edition with Containers. [English]
Root device type: ebs Virtualization type: hvm **Select** 64-bit

Microsoft Windows Server 2016 Base Nano - ami-5a51409e
Microsoft Windows 2016 Datacenter Edition Nano. [English]
Root device type: ebs Virtualization type: hvm **Select** 64-bit

I am going to use the top option of **Amazon Linux AMI**. It's a Linux system similar to CentOS. There are many choices of OS here, and I have not tried many of them. My intention is to use this server for a bit of testing, and I don't want it to cost me anything, so I am unsure if picking say Windows will incur any costs (I would guess it does). As long as it has a 'Free tier Available' label you should be good to go.

Select **Amazon Linux AMI 64bit** (Blue button on the right hand side of the page.)

Next we have to choose our instance type. This is setting up how many processors (vCPUs) memory, hard disk space etc that we want.

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 (1)	Memory (GiB)	Instance Storage (GB) (1)	EBS-Optimized Available (1)	Network Performance (1)	IPv6 Support (1)
☐	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
☒	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
☐	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
☐	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
☐	General purpose	m4.large	2	8	EBS only	Yes	Moderate	Yes
☐	General purpose	m4.xlarge	4	16	EBS only	Yes	High	Yes
☐	General purpose	m4.2xlarge	8	32	EBS only	Yes	High	Yes
☐	General purpose	m4.4xlarge	16	64	EBS only	Yes	High	Yes
☐	General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gbps	Yes
☐	General purpose	m4.16xlarge	64	256	EBS only	Yes	20 Gbps	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

*** DO NOT CLICK Configure and Launch *** There are a lot of options here, but for those of us wanting a free server, there is only one, the **t2.micro (Free tier eligible)** so that makes our decision process pretty simple.

Select **Configure Instance Details** from the bottom right of the page.

Before we launch our instance, we can make some changes. Click the **Add Storage** tab.

The screenshot shows the 'Step 4: Add Storage' configuration page in the AWS Management Console. The navigation bar at the top includes 'Services', 'Resource Groups', and a progress indicator with steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage (highlighted with a red box), 5. Add Tags, 6. Configure Security Group, and 7. Review. The main heading is 'Step 4: Add Storage'. Below it, a text block explains that the instance will be launched with specific storage settings and that additional EBS volumes can be attached after launch. A table lists the storage configuration for the 'Root' volume:

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-0190f070962758e1	8	General Purpose SSD (GP2)	100 / 3000	N/A	☒	Not Encrypted

Below the table is an 'Add New Volume' button. A blue information box states: 'Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.' At the bottom right, there are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Add Tags'.

By default you get a storage area of 8GB, if this is not enough then you can increase it. There is an information box that states (at the time of writing) you can have up to 30GB free. Remember, if you plan to have more than one EC2, you have to split that across them.

I left my configuration at 8GB.

Click the **Add Tags** tab.

The screenshot shows the 'Step 5: Add Tags' configuration page in the AWS Management Console. The navigation bar at the top includes 'Services', 'Resource Groups', and a progress indicator with steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags (highlighted with a red box), 6. Configure Security Group, and 7. Review. The main heading is 'Step 5: Add Tags'. Below it, a text block explains that a tag consists of a case-sensitive key-value pair and that tags will be applied to all instances and volumes. A table lists the tags:

Key (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes
Test Webserver		☒	☒

Below the table is an 'Add another tag' button with the text '(Up to 50 tags maximum)'. At the bottom right, there are buttons for 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure Security Group'.

You don't have to do this, but if you are going to have a lot of instances, it does become necessary. Just add a name for this instance (I used Test Webserver.)

Click the **Configure Security Group** tab.

Services Resource Groups

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

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:

Description:

Type	Protocol	Port Range	Source
SSH	TCP	22	Custom 0.0.0.0/0
HTTP	TCP	80	Custom 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](#)

Feedback English

© 2009 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

You can look at security groups as a firewall. All ports are closed unless you open them. By default port 22 (SSH) is open, or you would have no access to your OS. As this is going to be a web server, I will also open Port 80.

Click **Add Rule** and select **HTTP** as the type. **0.0.0.0/0, ::0** means accept traffic from any IP Address. For the SSH I would limit that to your own IP (say your work or home IP).

Now you can select **Review and Launch**.

You will be shown a review screen of your selected settings.

Services Resource Groups

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

Step 7: Review Instance Launch

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

Improve your instances' security. Your security group, **launch-wizard-1**, is open to the world. Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

Amazon Linux AMI 2017.03.1 (HVM), SSD Volume Type - ami-ed100689

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.

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GiB)	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-1
Description: launch-wizard-1 created 2017-07-23T15:53:28.995+01:00

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0/0
HTTP	TCP	80	0.0.0.0/0
HTTP	TCP	80	::0

Instance Details [Edit instance details](#)

Storage [Edit storage](#)

[Cancel](#) [Previous](#) [Launch](#)

Feedback English

© 2009 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

If you are happy, select **Launch** in the bottom right hand corner of the page.

You will be presented with a pop up that is asking about **Key Pairs**. When using SSH, **key pairs** are used rather than a simple username and password as **key pairs** are much more secure.

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](#).

Choose an existing key pair

Select a key pair

No key pairs found



No key pairs found

You don't have any key pairs. Please create a new key pair by selecting the **Create a new key pair** option above to continue.

Cancel

Launch Instances

Where you see [Choose an existing key pair](#), you need to change this to [Create a new 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](#).

Choose an existing key pair

Choose an existing key pair

Create a new key pair

Proceed without a key pair



No key pairs found

You don't have any key pairs. Please create a new key pair by selecting the **Create a new key pair** option above to continue.

Cancel

Launch Instances

Select a new for the new key pair, and click the [Download Key Pair](#) button.

You will see the [.pem](#) file download (this is browser specific).

Click the [Launch Instances](#) button.

Launch Status

✓ **Your instances are now launching**
The following instance launches have been initiated: `i-0228b1b647ec82a93` [View launch log](#)

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

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](#)

The next page confirms that your instance is launching.

Click the **View Instances** button.

Instances

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Launch Time
	i-0228b1b647ec82a93	t2.micro	eu-west-2a	running	Initializing	None	ec2-35-176-205-194.eu...	35-176-205-194	-	TestWebSvr	disabled	July 23, 20...

Instance: **i-0945a01641bcta807** Public DNS: -

Description | Status Checks | Monitoring | Tags

Instance ID: i-0945a01641bcta807
Instance state: terminated
Instance type: t2.micro
Elastic IPs: -
Availability zone: eu-west-2a
Security groups: -
Scheduled events: -

Public DNS (IPv4): -
IPv4 Public IP: -
IPv6 IPs: -
Private DNS: -
Private IPs: -
Secondary private IPs: -
VPC: IT: -

On this page we can see the information about our instance. The part we will need initially is the IP Address.

Connecting to the EC2 Instance

To connect to the EC2 instance you will need your **IP Address** (obtained from the EC2 Status Page) and you will need the **.pem key pair** file you downloaded.

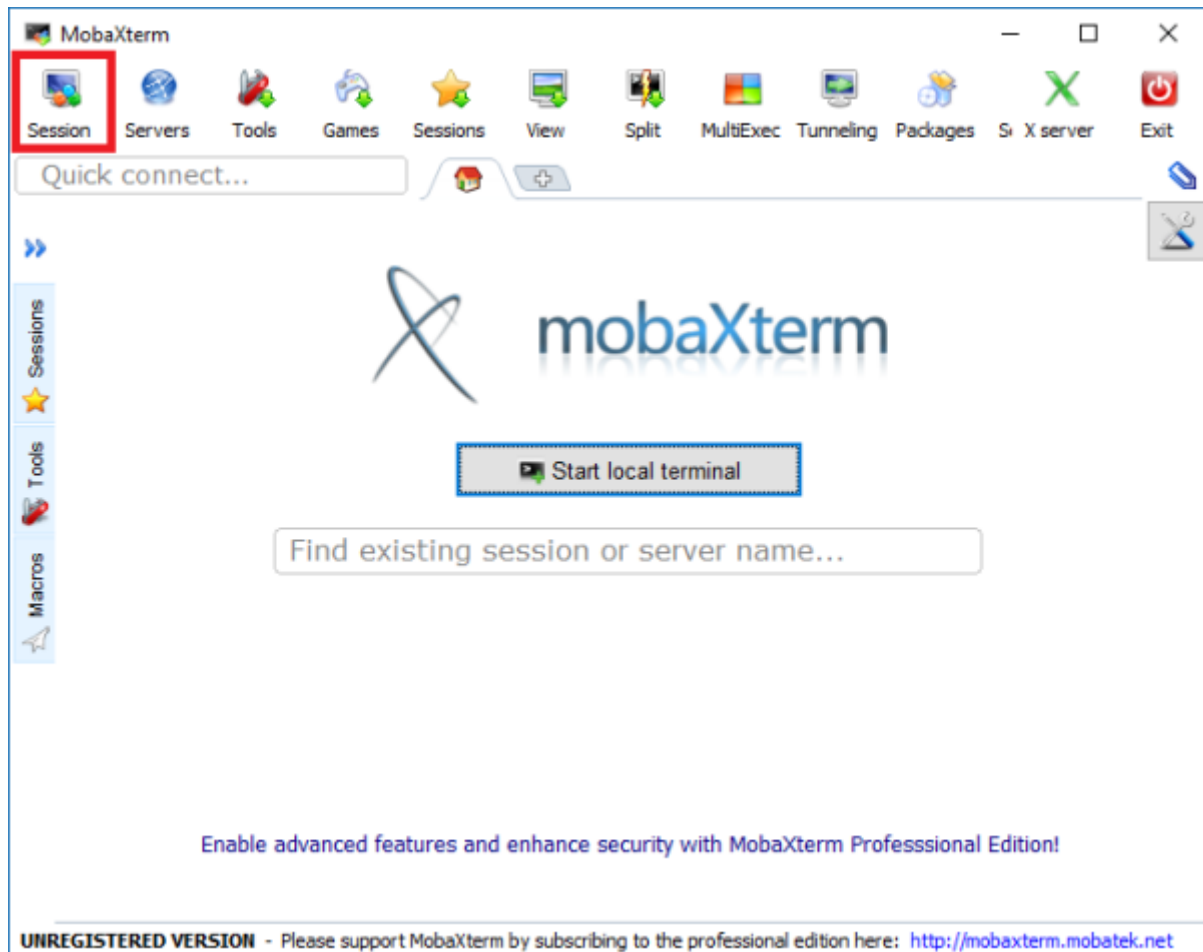
Instances

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring	Launch Time
	i-0228b1b647ec82a93	t2.micro	eu-west-2a	running	Initializing	None	ec2-35-176-205-194.eu...	35-176-205-194	-	TestWebSvr	disabled	July 23, 20...

If you have neither of these, please read the entire guide.

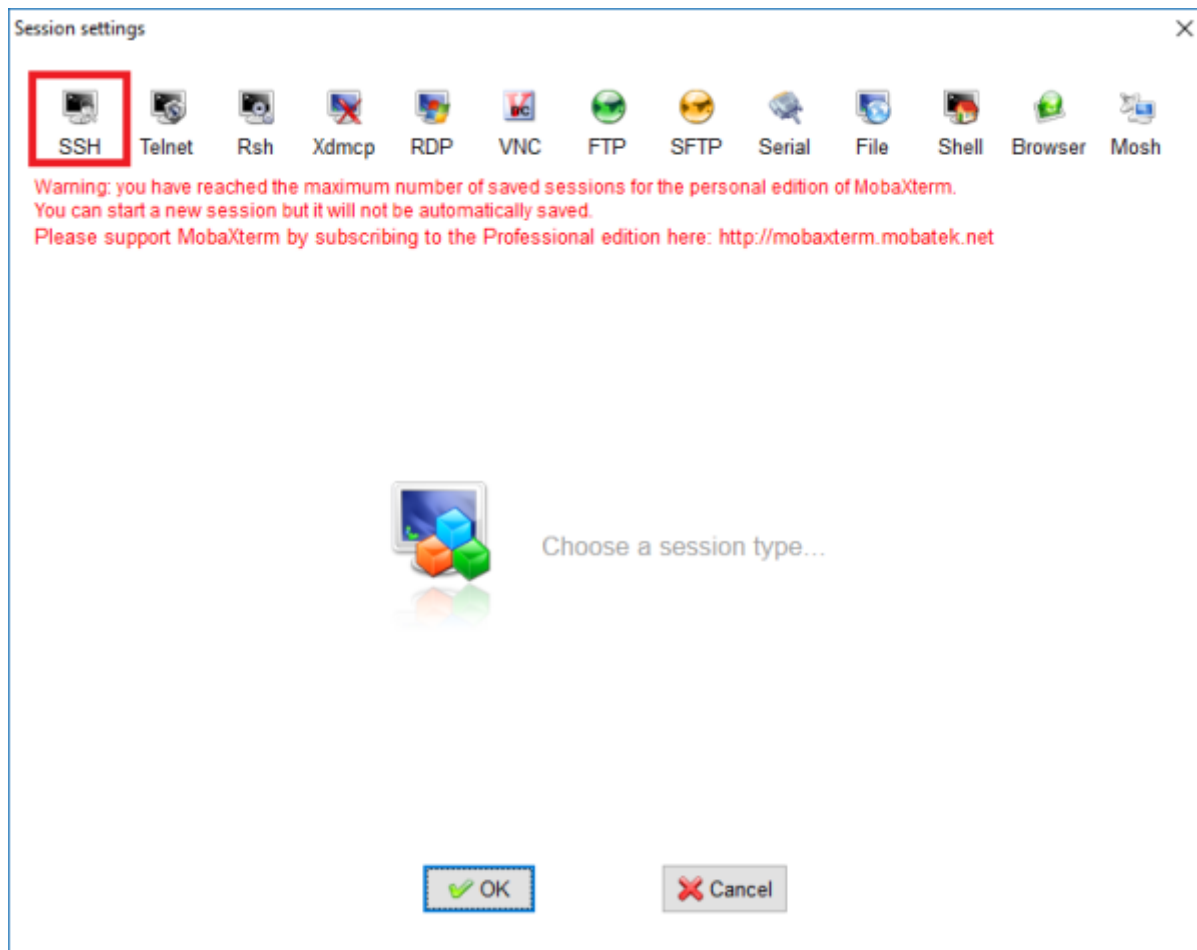
To be able to connect you will need a suitable SSH program, of which there are many. The most popular is probably Putty, however to use putty you need to use Puttygen to convert the .pem key first. So I am using another free application called **MobaXterm** which you can download from [HERE](#):

Once installed, run MobaXterm.

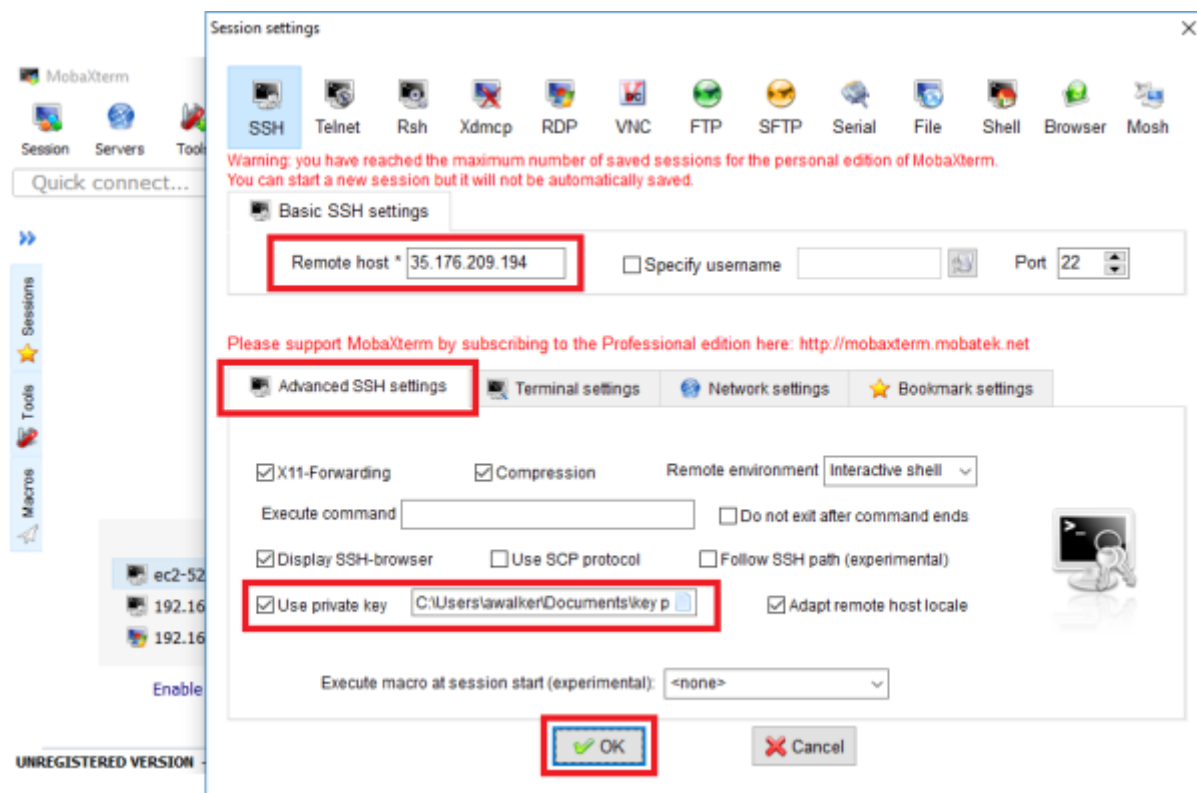


At the top of the page, select **Session**.

Then select **SSH**.



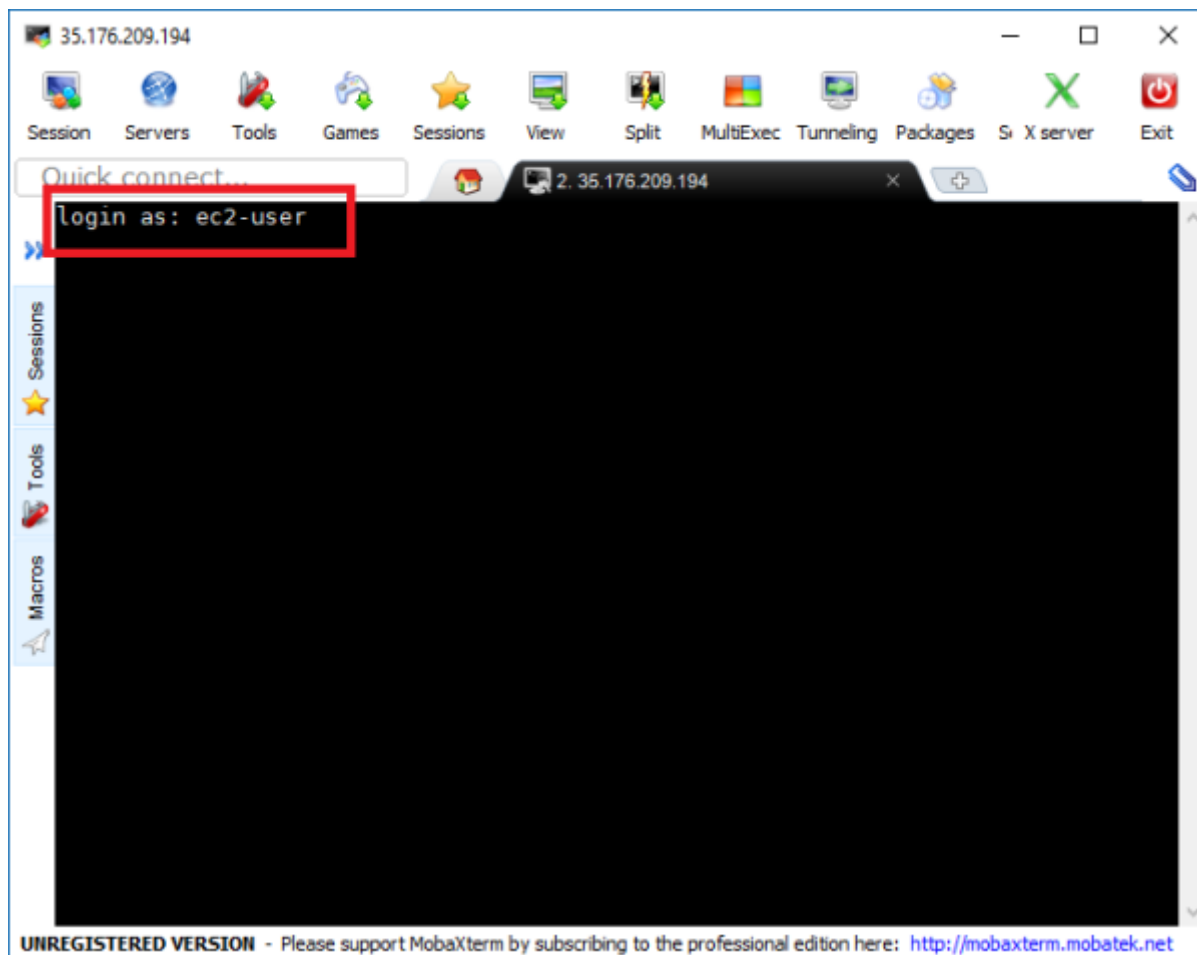
The SSH Page will open and we have to enter the following information.



Remote Host: This is the IP Address of your EC2 Instance. **Advanced Settings:** Click this tab, select **Use Private Key** and navigate to where you saved your **key pair**.

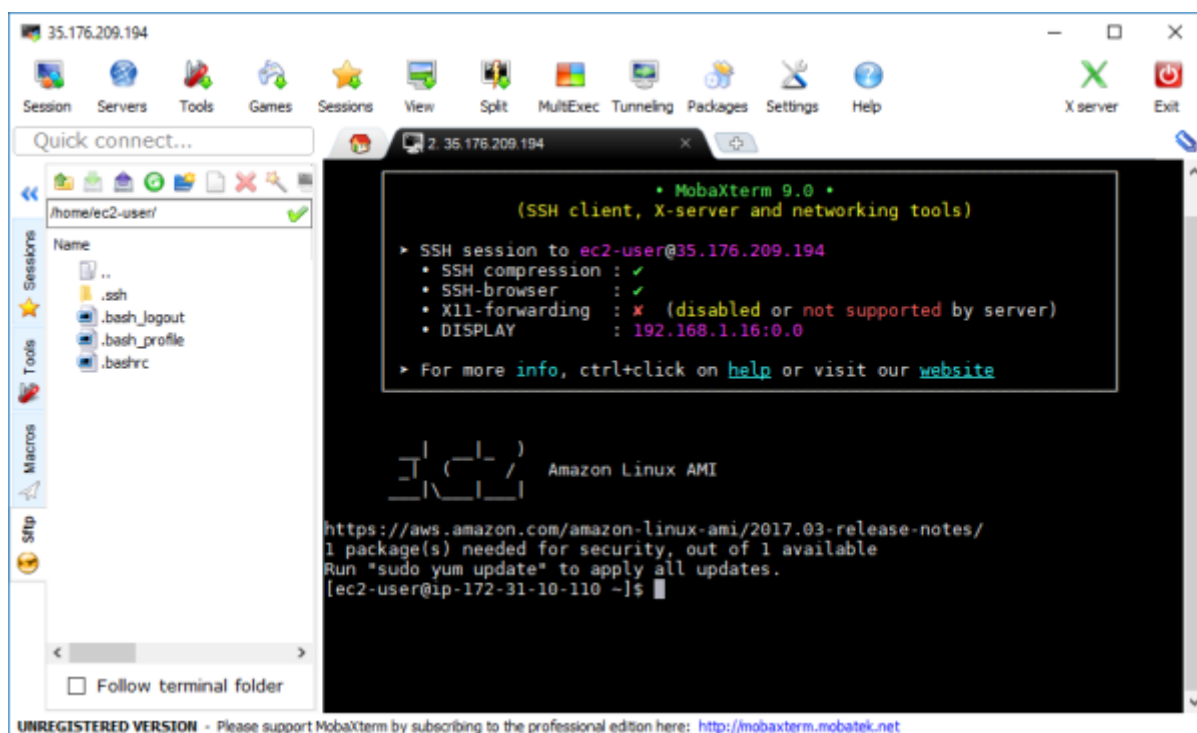
Click **OK**.

MobaXterm will now connect to your EC2 instance, using your key pair to authenticate.



You will have to enter a password. The default password is `ec2-user` (case sensitive)

Once you enter the password you should be at the console.

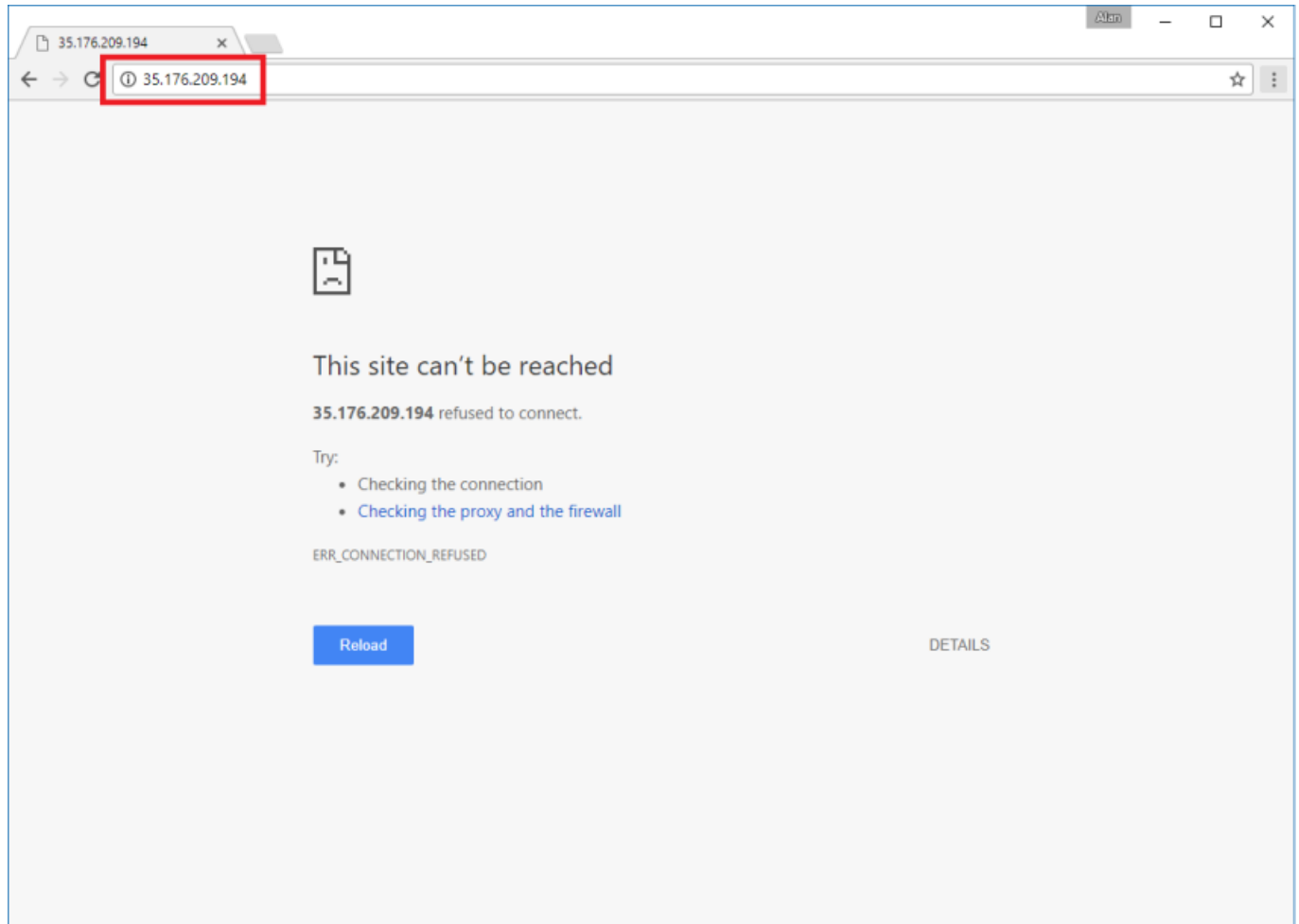


You can now enter standard Linux commands, that's it, you are now connected to your EC2 instance.

Install the Web Server

As this example is a simple web server, lets install a web server and give it a little test.

First thing to do is to navigate to the IP Address of you EC2 instance, to ensure there is nothing already running.



You should see that there is nothing on this IP Address.

So we can install a web server. Open MobaXterm and from the command line enter:

```
sudo yum install httpd -y
```

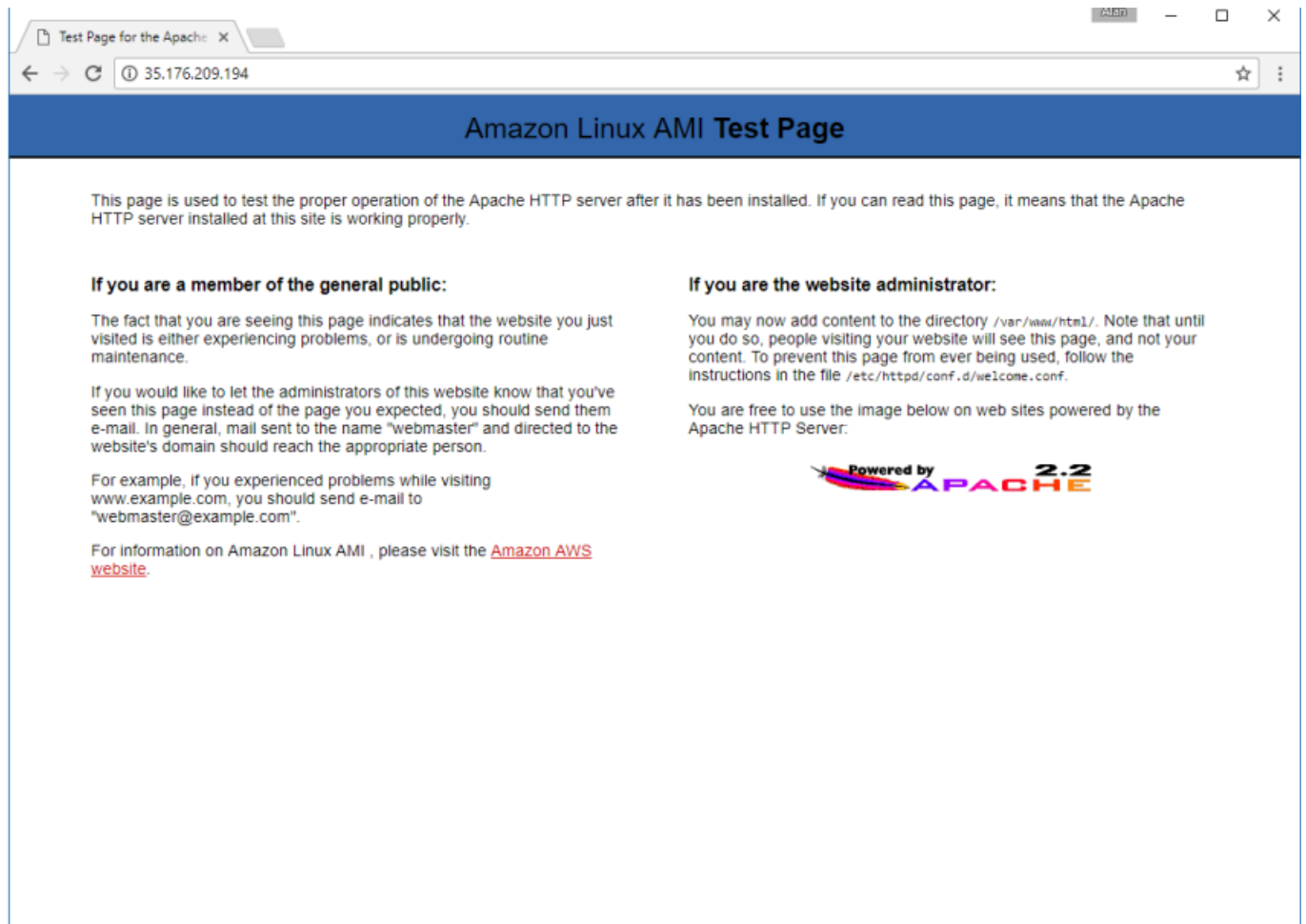
This will install Apache web server, and configure the basic system files required for it to run.

Once installed, you will have to start the Apache service. You can do this with:

```
sudo service httpd start
```

You should see a confirmation that looks like: Starting httpd: [OK]

Now you can retest your connection from a web browser.



You should see an Apache test web page, this means your server is running and you can start to build and upload your test website.

From:
<https://wiki.alanwalker.uk/> - WalkerWiki - wiki.alanwalker.uk

Permanent link:
<https://wiki.alanwalker.uk/doku.php?id=test&rev=1678401305>

Last update: 2023/03/09 22:35

