

abYsis

Using on AWS

Table of Contents

Table of Contents	2
Introduction	3
Installing a RHEL/Rocky 9 EC2 instance	3
Launching an Instance	3
Logging in to EC2 Machine	6
Generating separate Key Pair for an external user	7
On a local Mac or Linux machine:	7
On a Windows machine:	8
Security, Domain name, HTTPS	9
Username and Password Protection	9
Persistent IP address assignment	10
Domain name assignment	10
Control Access through Firewall	11
Encrypt data using HTTPS	13
Snapshot for AWS	14
Create your instance	14
Noting details and detaching the volume	14
Turn Snapshot into a volume	15
Attach abYsis volume to your instance	16
Start new abYsis instance	17

Introduction

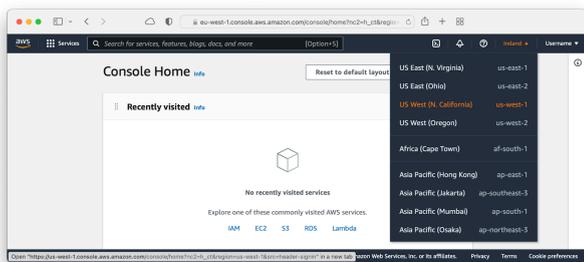
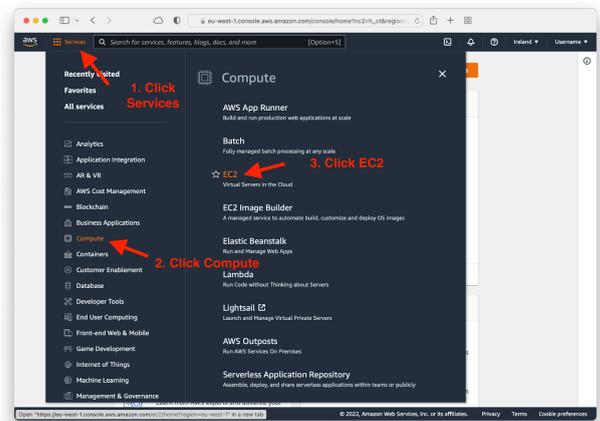
This manual is intended to give people an introduction to how abYsis might be used on a hosted service. We use AWS as an exemplar based but the information given here should not be considered comprehensive. Readers should access the AWS manuals and information. As always your own IT professionals should independently determine whether this is the appropriate option for your company and what steps are required to secure the system to your own requirement.

Installing a AWS Linux EC2 instance

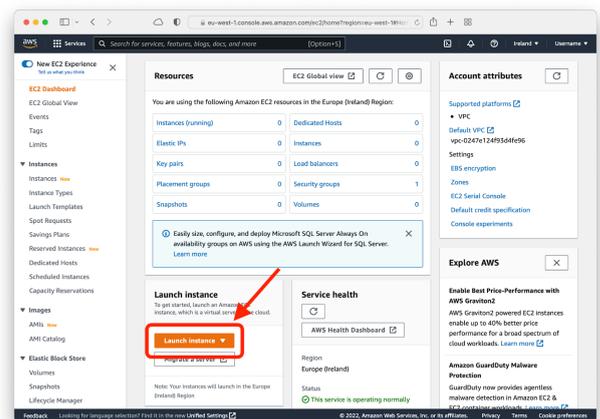
Launching an Instance

Before any work can begin you will need to create an Amazon AWS account and login to the Management Console.

Make sure that you select the appropriate cluster location - typically geographically close to your own location.

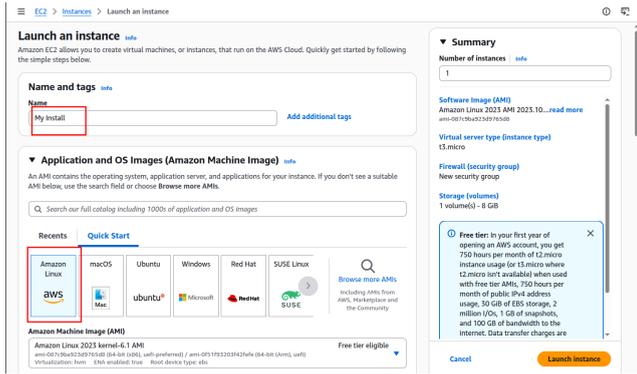


Now select the 'EC2' link under Services/Compute and click the 'Launch Instance' button.



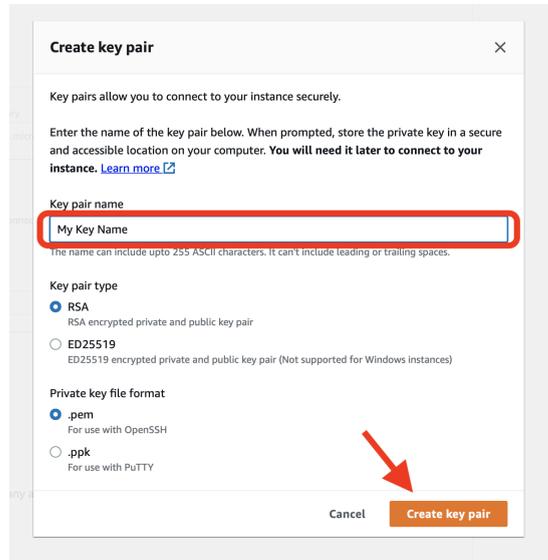
Now go through the following steps:

- a) Name your instance so you can easily recognise it later and then search for your preferred OS.

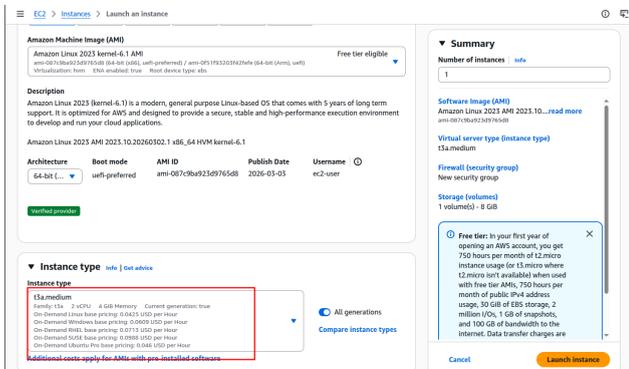


- b) The most suitable AMI is likely to be found in the AMI Marketplace which may indicate that a cost is involved. For simplicity we have chosen Amazon Linux
- c) If necessary, click 'Select' for the OS release you wish to use. You may be asked to confirm the cost of the subscription for using the OS.
- d) In the "Instance Type" panel, select your instance type. Below we have selected 'm4.large'.

- Enter and name for the key.
- Click "Create key pair."

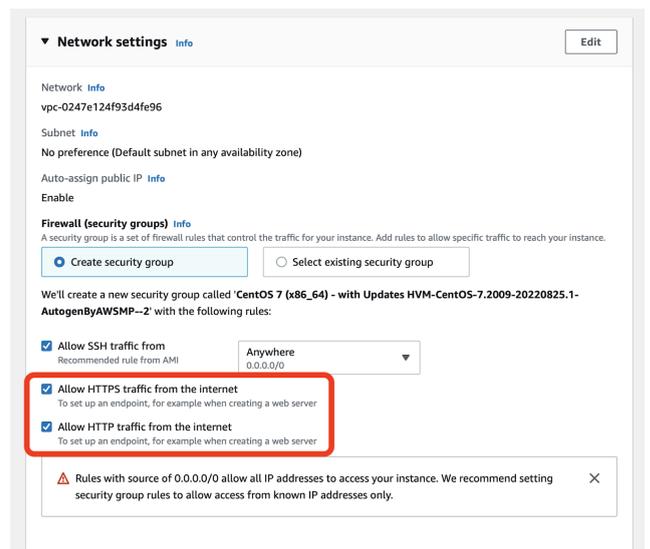


- The '.pem' key file will be downloaded. This file is important as it is required to connect to the server.

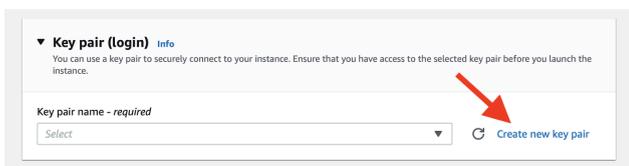


- e) In the "Key pair (login)" panel create an SSH key.
 - Select 'Create a new key pair'

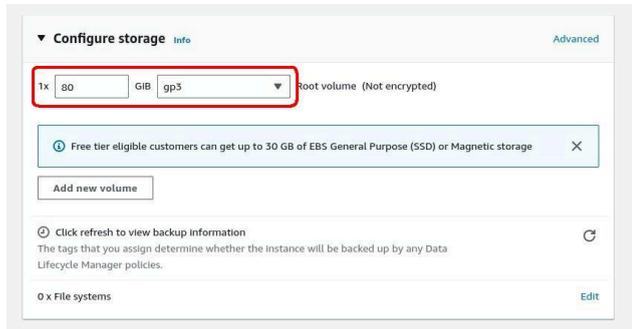
- f) In the "Network Settings" panel select "Allow SSH", "Allow HTTPS" and "Allow HTTP" traffic from the internet.



- g) In the "Configure storage" panel, edit the size as required. 60Gb is the minimum currently

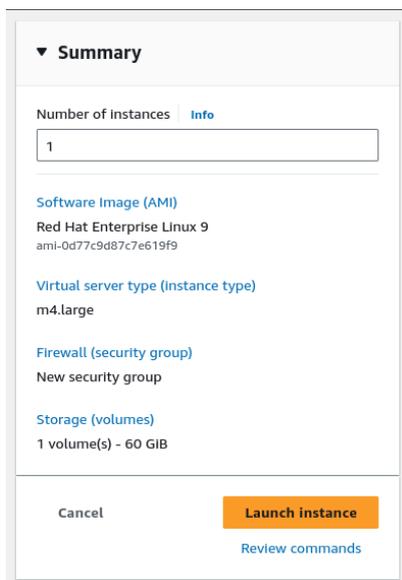


recommended. Typical suitable storage media is either gp2 or gp3. It is best to review as their usage will involve cost.

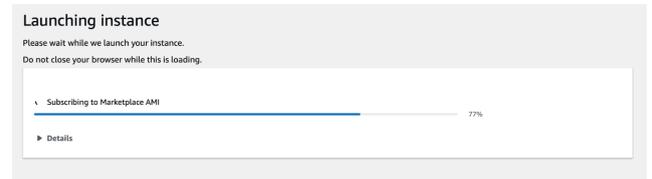


h) When your instance is **stopped**, the storage volume (disk) is retained (at a small cost) so that the instance can be started again. When your instance is **terminated** this volume is automatically deleted. Under “Advanced” there is an option for “Termination protection”. When it is **enabled** your instance cannot be terminated and your installation disk will not be deleted without first manually changing the termination protection to **disabled**.

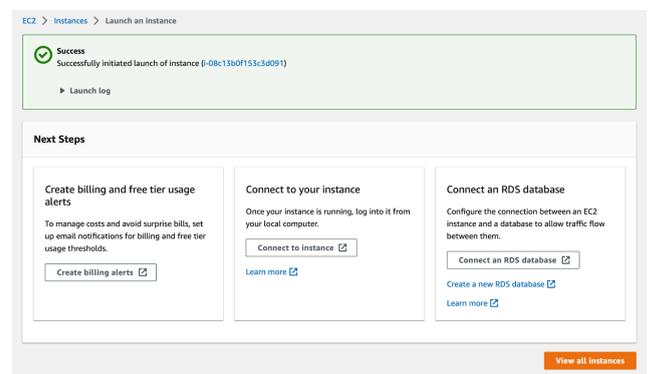
i) The “Summary” panel to the right shows the options you have configured. Check they are correct and click “Launch instance”



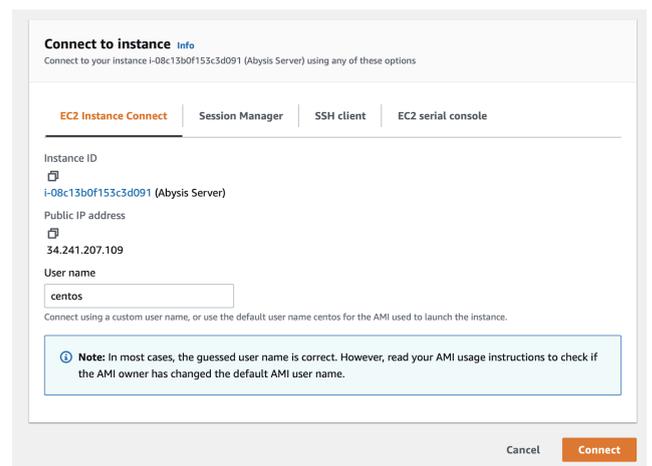
j) A screen will be shown with the progress of the instance setup, this can take a few minutes.



k) You should be shown a confirmation screen once the instance is ready

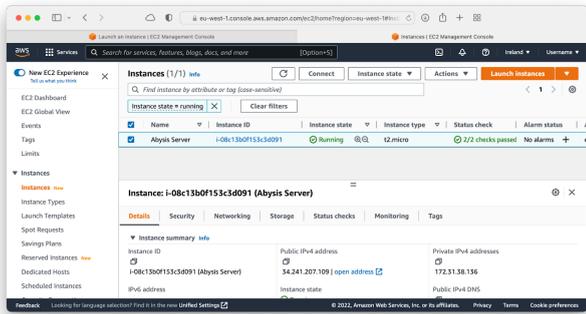


l) Clicking “Connect to instance” will take you to a screen with details on how to connect to your instance, including its public IP address.



Please note that the login account for a RHEL machine is **ec2-user** and for a Rocky server it is **rocky**.

Alternatively going to your instance list should show your running instance:

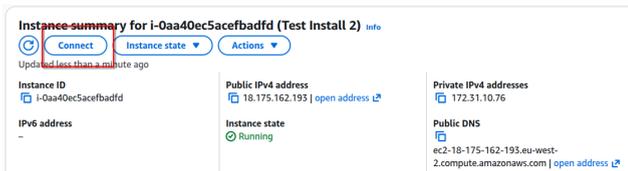


You are now able to connect to the server without being logged in to the AWS platform.

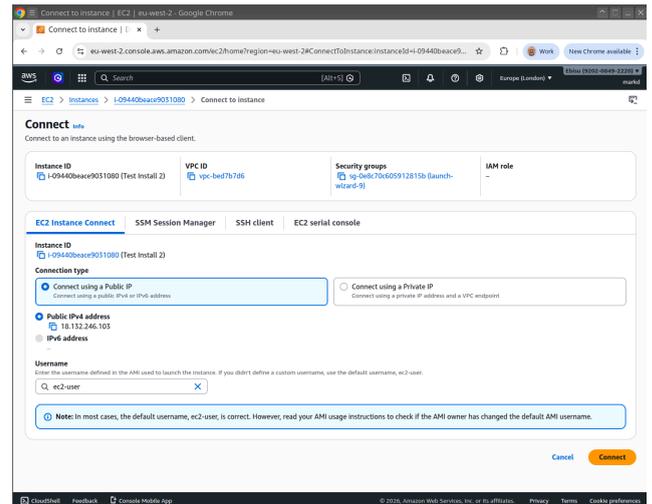
In the following example, our private key file is called abysis.pem and our Public DNS is ec2-123-456-789-101.eu-west-2.compute.amazonaws.com:

Logging in to EC2 Machine

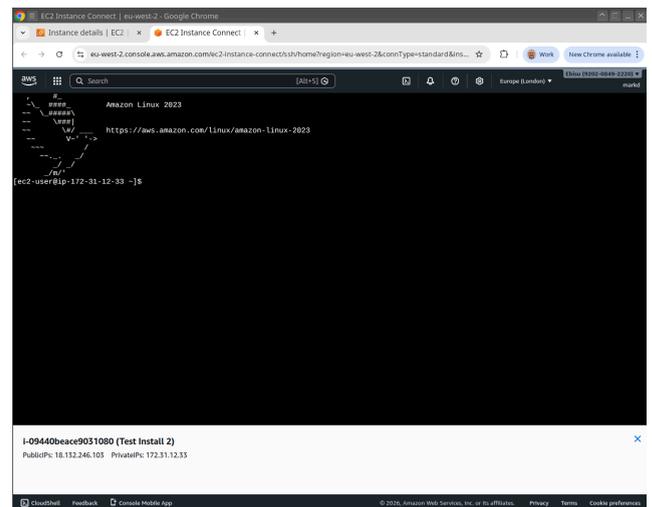
Connect from AWS



- You can connect to your server through the AWS console. Select your instance and then click the Connect button



- You will have several connection options. You can use Connect Using a Public IP and click the Connect button.



- A new browser tab will open with a bash terminal for you to use for the installation.

Connect from a local Windows, Mac or Linux:

- Open a terminal or PowerShell window.
- Copy the *.pem file to a local directory and change the access rights as follows:
- `chmod 400 abysis.pem`
- Connect to Amazon Linux as follows:
`ssh -i abysis.pem ec2-user@ec2-123-456-789-101.eu-west-2.c`

ompute.amazonaws.com

This user will have password-free sudo permissions.

- You should now be connected.
- If you are installing abYsis yourself you can start from here.

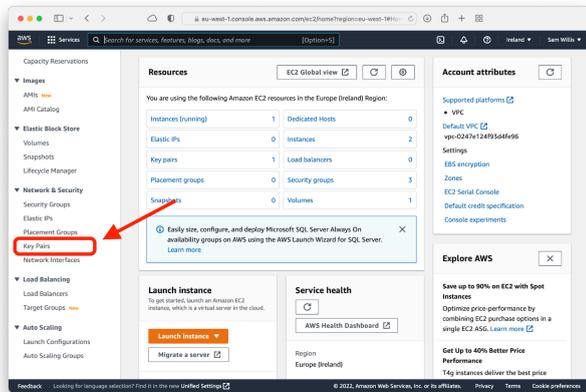
Generating separate Key Pair for an external user

If you are not installing abYsis yourself and are asking an external provider to perform the installation, you may want create an additional Key Pair which can be deleted at a later date to keep access under your control.

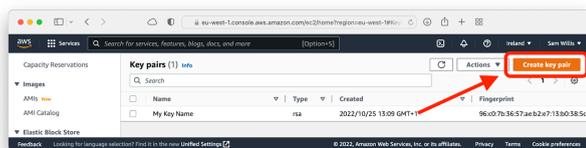
This can be achieved in the following manner.

On the AWS management console

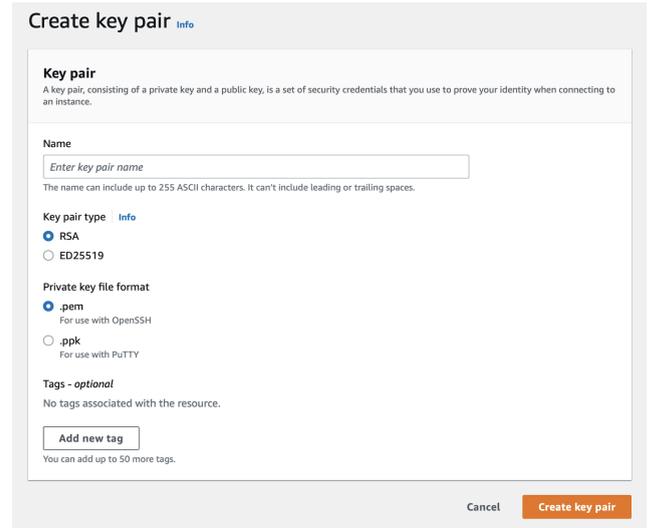
Go to 'Key Pairs' in the Network & Security section.



Click 'Create Key Pair'.



Give the pair a name and click 'Create key pair'.



- The new Key Pair for this additional user we have called abysis_user. You can call the file whatever name you prefer.
- You will again be prompted to save the private key file (abysis_user.pem).
- You need to keep a local copy of this file and also send to the person who may require access.
- Now, you will need to add the associated Public Key to your user's authorization file.

On a local Windows, Mac or Linux machine

In the local directory on your local machine, type

- `ssh-keygen -y`

You will be prompted for the file name, so type the file, e.g. abysis_user.pem

This will reveal the Public Key.

In a separate terminal window on your machine, connect to the **EC2 server** using the primary Private Key (i.e. the key generated in item (d) above that we called abysis).

- Change directory to `/home/centos/.ssh`

- Edit the file called 'authorized_keys' within the .ssh directory. (You will need to use the standard vi editor and switch to insert mode).
- You should see the Public Key for your current connection in the authorized_keys file.
- In the local window, highlight the text of new Public Key.
- Add the new Public key to this file.
- Save the file.

You have now added the extra Public Key to allow access using the new Key Pair.

In the future, if you wish to revoke access by abysis_user, you can remove this line from the authorization file.

What to send to the person installing abYsis

You need to send the appropriate private key to the person installing abYsis.

In our example this is abysis_user.pem

Remember to keep a local copy of this file.

Security, Domain name, HTTPS

When installing abYsis it is important to consider how users will access the system and how to ensure it is not accessed by non-authorized persons. Below are suggestions on how you might secure access to abYsis. However, you should use your own IT facilities and personnel to ensure that securing abYsis access meets your own criteria.

For the purposes of these instructions it is assumed that you have already installed abYsis on an AWS server and it is running. It will be using HTTP by default which is unencrypted and should be secured to ensure that others cannot access.

Username and Password Protection

It is possible to configure your abYsis apache server to require users to provide a username and password before using it. Setting this up would be part of your abYsis installation process and can be found in the **Securing abYsis** document.

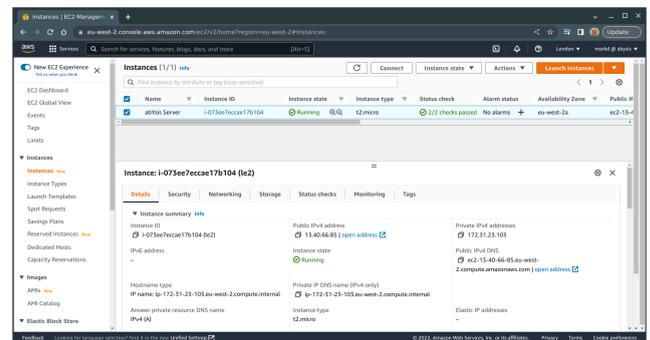
For this level of security you would not be required to change any AWS configurations.

Persistent IP address assignment

On an AWS server it is very likely that when the server reboots, a different IP address will be

assigned unless action is taken to assign the server a static IP address to your instance.

Go to your Instance page on AWS. Take a note of the Instance ID, Private IP address and Availability Zone (under the Network tab)

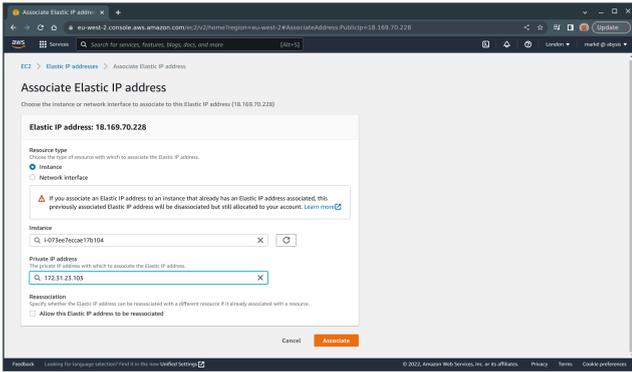


Scroll down the left hand menu and select Elastic IPs and then click the **Allocate Elastic IP address** button.

Make sure the Network Border Group is similar to your Availability Zone (you took note of above) and click the **Allocate** button.

A new IP will now be selected for you. To attach this to your instance click **Associate Elastic IP address** in the **Action** menu.

Select the Instance ID and the Private IP address (you took note of above) that this public IP address will point to and click **Associate**.



<http://81.123.45.45/abysis/index.cgi>

To edit this file use a text editor such as nano

`sudo nano /etc/httpd/conf.d/abysis.conf`

Your instance now has a new public IP address that you can use to access your server.

- This will be important if you wish to provide an encrypted HTTPS connection

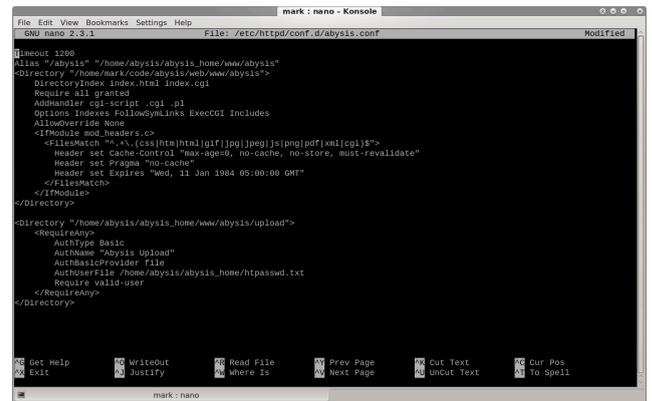
Domain name assignment

You will want to point your **Domain Name** to your static IP address. This will make it easier for users to access and is essentially necessary if you are planning to use HTTPS. Domain name assignment is implemented using the DNS Records editor supplied by the organisation that you bought the domain name from.

Once added to your DNS Records, you will also need to change your apache conf file on your abYsis server to include this domain name.

To do this, Log on to your abYsis server as a user with sudo permissions

If you used the default settings the name of your apache configuration file will be abysis.conf. The abysis part of the filename is the same as the abysis part of the URL you use to access the server.



You need to add the following lines to the top of the file, using your registered domain name

<VirtualHost *:80>

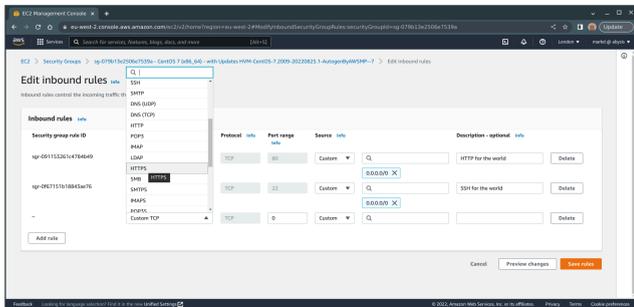
ServerName www.yourdomain.com

And add the following line to the very end of the file

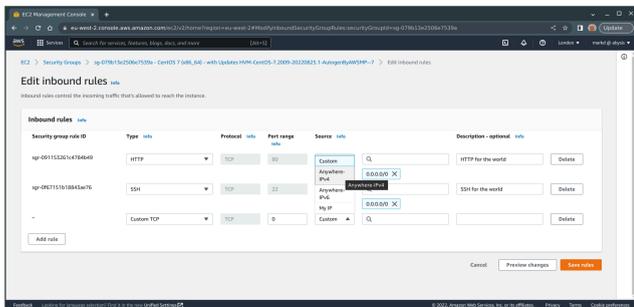
</VirtualHost>

Your file should now look like the next image. Use Control-X and answer **Y**es to save the file.

When you click the **Add rule** button a new line will appear.



The first drop down menu will allow you to choose from lots of preset configurations for different types of internet traffic. Choose HTTPS to allow HTTPS access to your server. Then you need to choose who can access the server using HTTPS.



The main options are:-

- Anywhere IPv4 - anyone on the internet
- My IP - only the computer you are currently using
- Custom - you can put a specific IP address (e.g. a colleagues) in the next box

You can add as many rules as you like. For example, if you want access to be limited to just 3 IP addresses then you will need 1 rule for each IP address and each access type.

At the end click **Save rules**. Those rules will now be enforced on your server.

Note

- You should always have a rule for ssh access. If you limit it to a specific IP and you can no longer use that IP address then you will lose access to your server. Therefore, it is best to leave ssh access open to everyone (it is password protected).
- If you have an SSL certificate installed on your server you need HTTPS rules. If not then it is not necessary to set them.
- If you use Let's Encrypt to install certificates on your server, always leave HTTP open to everyone.

Encrypt data using HTTPS

In order to provide HTTPS access to an abYsis server you need an SSL certificate from a Trusted Certificate Authority and configure apache on your abYsis server.

The instructions for configuring the server can be found in the **abYsis Installation and Data Loading** document. You will need to configure the following on your AWS server to be able to set up an SSL certificate.

- You would need a persistent IP address. Instructions for this can be found above.
- You would need a domain name attached to the IP address above.
- You would need to make sure your HTTPS port 443 is open on your abYsis server for people to access your encrypted pages. This would be set during section (e) of the **Launching an Instance** section above.
- An SSL certificate provider such as Let's Encrypt generates a certificate that usually lasts for 90 days. It uses a certbot program to automatically renew a

At some point after an initial installation you might decide to bring back an abYsis installation via an AWS snapshot.

Whilst abYsis is not supplied as a Snapshot, this documentation is written to help people who might wish to generate their own and deploy at a later date.

A snapshot is of a fully configured OS (e.g. RHEL or Rocky 9) instance with an abYsis installation. By following the steps below you can turn a snapshot into an instance that is ready to go.

Create your instance

The first step is to create a machine to run abYsis on. This is the exact same process as outlined in the **Launching an Instance** process at the beginning of this document.

- You will then go to a page confirming that your instance has been launched.

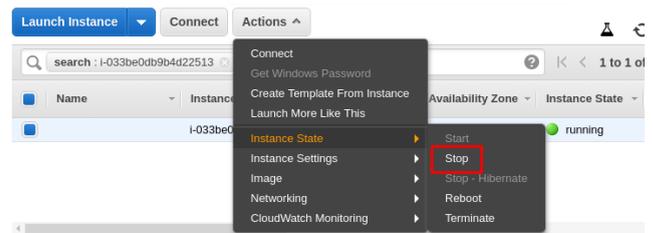
certificate. To enable this it is advisable to ensure that HTTP port 80 is accessible by everyone. Your abYsis site should be secure because of the redirect to HTTPS, but should be checked.

- certbot ensures that normal HTTP traffic is automatically redirected to HTTPS.
- It may take a couple of hours for your certificate to be recognised and accepted by browsers

Snapshot for AWS

Scroll to the bottom of the page and click the **View Instance** button on the right.

- This will take you to the **instance** page which lists all instances you have. Select the instance you just created (it may be pending or initializing). Using the **Action** button hover over **Instance State** and click on **Stop**.



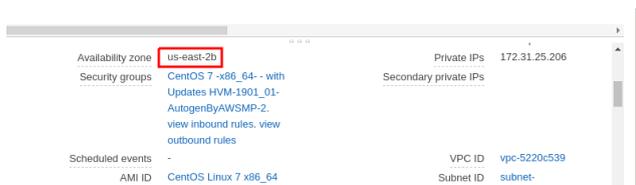
- j) You will be asked to confirm stopping the instance.

Noting details and detaching the volume

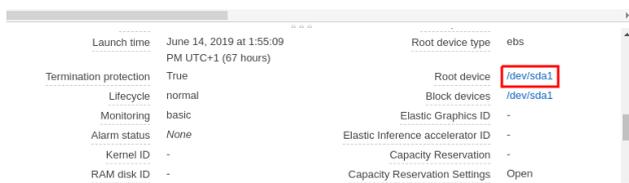
Attached to your new server (instance) is an 80GB volume, the server's harddrive. This is going to be replaced with the abYsis volume. While you do this you will need to take note of 2

things:- the **availability zone** and the **device name**.

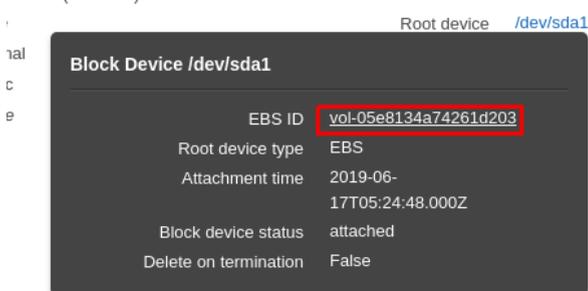
On the instance page make sure your instance (and no other if there are more than one) is selected. Then, in the bottom panel scroll down until you find **Availability zone**. Take a note of this as you will need this later.



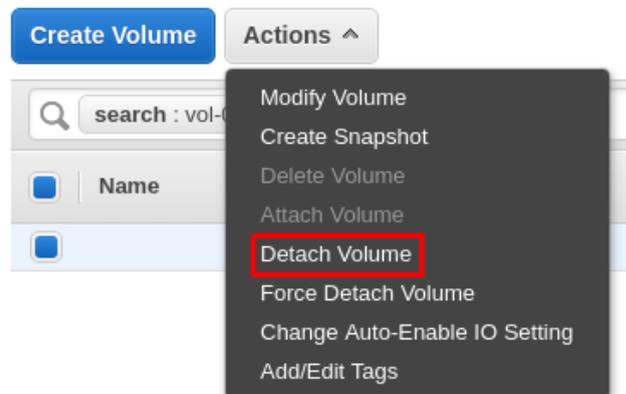
Continue to scroll down until you find **Root device**. Again, take a note of this. It will probably be **/dev/sda1**.



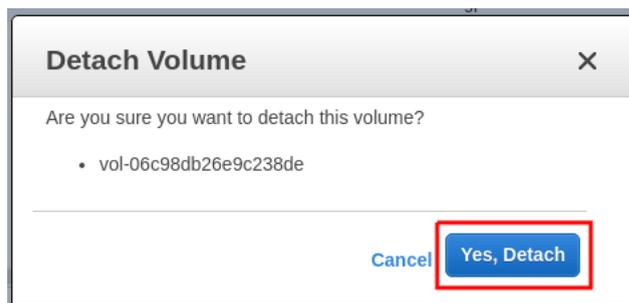
The device name is also a link. Click the link (such as [/dev/sda1](#)) and a **Block Device** pop-up will appear. Click the underlined EBS ID link and you will be taken to the **Volume** page.



On the **Volume** page your 80GB volume should be selected. Using the **Action** button next to the **Create Volume** button, select **Detach Volume**.



You will be asked to confirm that you wish to detach the volume. Click **Yes, detach**. The instance is now detached and ready for you abYsis volume to be attached.

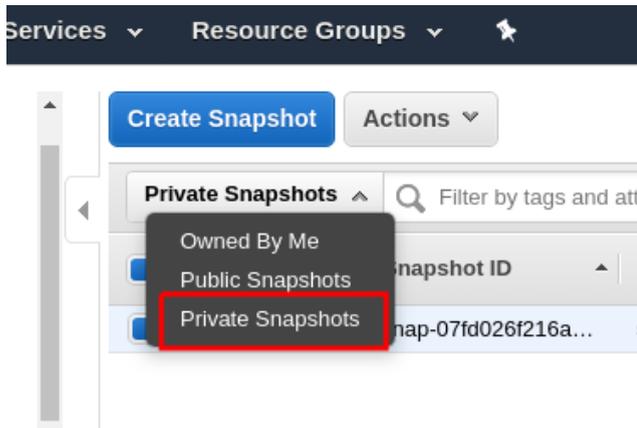


Turn Snapshot into a volume

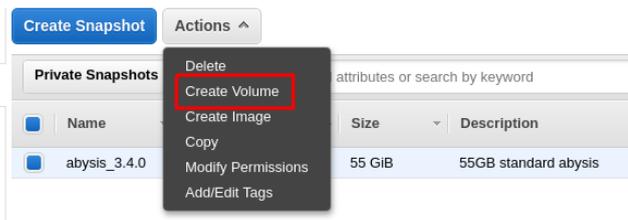
When a snapshot is available to you it will be possible to create a new harddrive (volume) from it.

In the left hand side panel, scroll down and click Snapshot. In the main panel, just under the **Create Snapshot** button click the dropdown menu (which probably says **Owned By Me**).

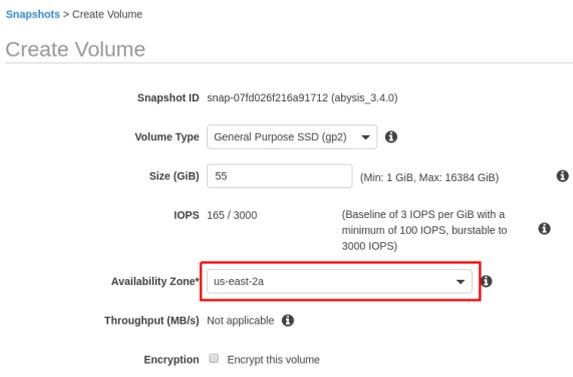
Change this to **Private Snapshots** and the relevant abYsis snapshot should appear.



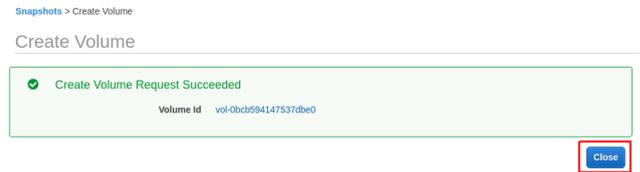
Click to select the abYsis snapshot and then using the **Actions** button above select **Create Volume**.



This will take you to a new **Create Volume** page. Take a note of the Half way down the page is the option **Availability zone**. Change that to the one that matches the one you noted earlier for your new instance. Then, scroll to the bottom and click the **Create Volume** button.



You will be told that the volume was created successfully. Click the **Close** button and you will be returned to the snapshot page.



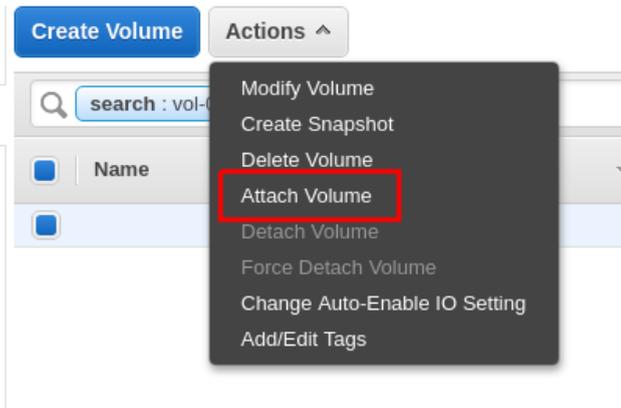
Attach abYsis volume to your instance

Your abYsis snapshot is now being built into a volume. You have essentially taken a copy of an installed disk and put it on a new disk. Now you have to attach it to your instance.

Use the left-hand panel to go to the **Volume** page. Find the volume you have created (you can use the **Created** date column for this) and select that volume.



Use the **Actions** menu and select **Attach Volume**

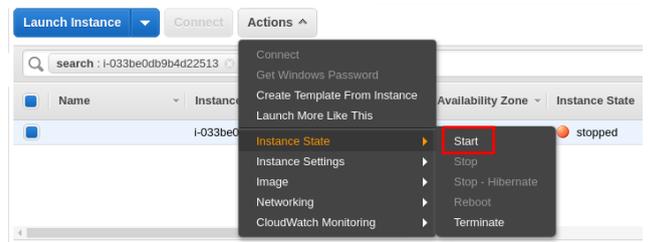


Start new abYsis instance

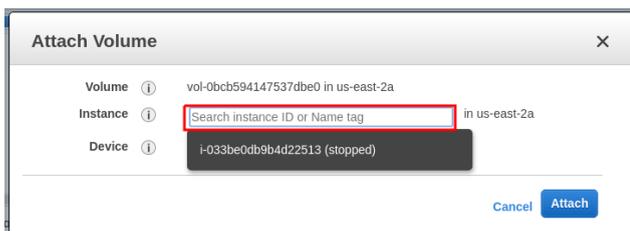
The abYsis instance has now been built. Now it needs to be started.

Use the left-hand panel to go to the **Instance** page.

Select the instance you created earlier and, using the **Actions** button, hover over **Instance State** and click **Start**. When asked to confirm click **Yes, Start**.



This will take you to a pop-up window, **Attach Volume**. The first thing to do is to click in the **Instance** box. It will display the instance that are in the same availability zone as the volume. Select the one you have just created.



Your instance will first go into a **pending** state and then **running** (use the  icon to refresh the page).

Then edit the **Device** from **/dev/sdf** to the device name you nated earlier, probably **/dev/sda1**. Then click the **Attach** button in the bottom right of the page to finish. You will then be returned to the **Volume** page.

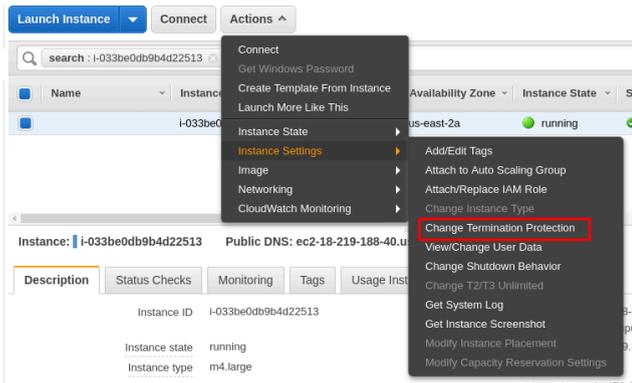


When your instance is running, in the bottom panel you will see the **IPv4 Public IP**. Copy this address, paste it into your browser and add **/abysis** to the end of the URL. This will take you to your new abYsis server.



There is an option when controlling instances to **Terminate** the instance. In AWS terms this means stopping the instance and then deleting it

and the EBS volume storage. To protect against that click the **Actions** button, hover over **Instance Settings** and click **Change Termination Protection**.



If the current setting is **disabled** then click the **Yes, Enable** button. Now the instance and volume cannot be terminated unless this setting is manually altered.

Should you need to access the instance's command line using ssh, click **Connect** and follow the instructions on the pop-up page. Please note, for a **RHEL** server the sudo username is **ec2-user** and for a **Rocky** server the username is **rocky**.

