lab

lab title

Introduction to AWS
V2.07

Course title

BackSpace Academy
AWS Certified Cloud Practitioner

Table of Contents

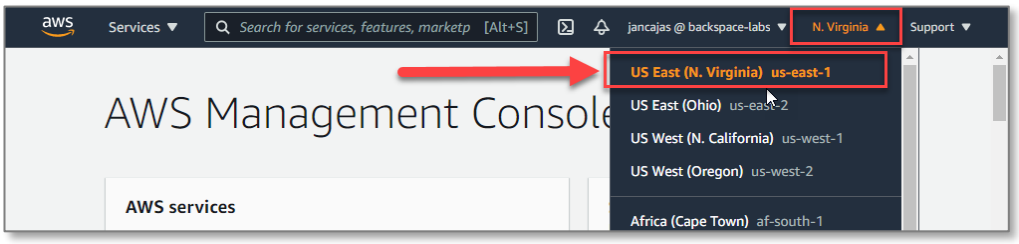
Contents

Table of Contents 1
About the Lab 3
Controlling Costs with AWS 4
Checking your AWS Usage and Monthly Bill 4
Using AWS Cost Explorer 5
View a Costs Report 6
Create an AWS Budget 7
Troubleshooting 13
Clean Up 14
Creating an S3 Bucket and Uploading Files 15
Create a Bucket 15
Uploading Files to your Bucket 16
Downloading files from your bucket 18
Troubleshooting 19
Clean Up 20
Creating a SQL Database with RDS 23
Creating a Security Group 23
Creating an RDS Database 26
Connecting to your RDS Instance 31
Troubleshooting Connection Issues 36
Clean Up 39
Creating a Web Server with EC2 41
Launch an EC2 Instance 41
Viewing your web server 47
Troubleshooting viewing your WordPress application 48
Finding the Username and Password for your WordPress application 51
Troubleshooting logging in to the WordPress application 53
Clean up 54
Monitoring User Activity with AWS CloudTrail 55
Clean Up 60

Sending Emails with Amazon SES 62
Requesting full access to SES 67
Creating an IAM User 68
Clean Up 70
Creating a Highly Available Architecture with Elastic Beanstalk 71
Clean Up 74

About the Lab

Please note that not all AWS services are supported in all regions. Please use the US-East-1 (North Virginia) region for this lab.



These lab notes are to support the hands-on instructional videos of the Introduction to AWS section of the BackSpace AWS Cloud Practitioner Course.

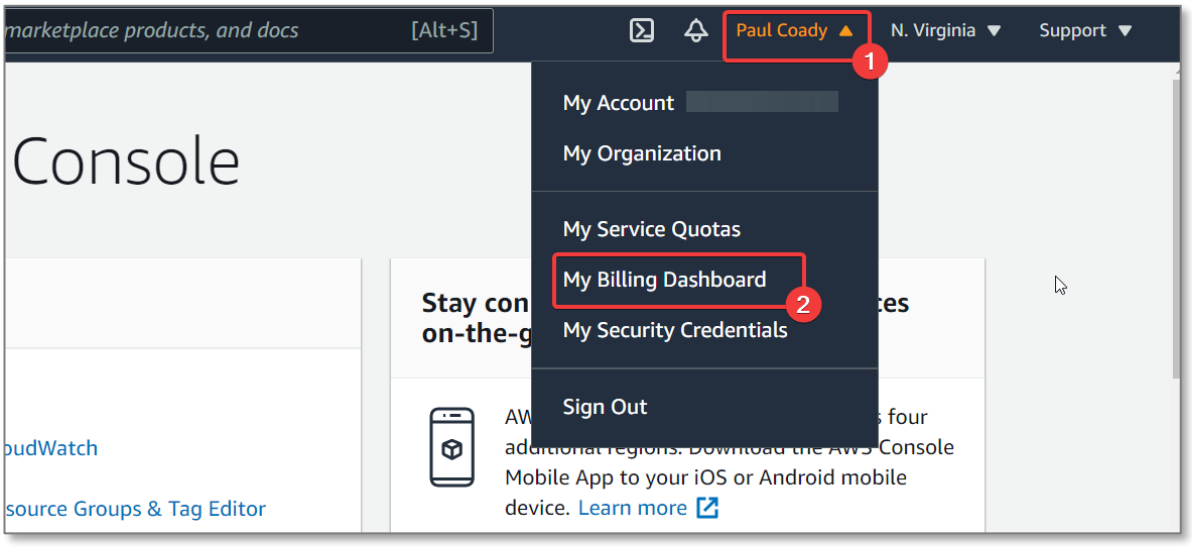
Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the latest version with any updates or corrections.

Controlling Costs with AWS

In this section we will learn how to use the AWS Billing & Cost Management Dashboard to keep track of costs. We will then create a Budget using AWS Budgets that will alert us when costs are exceeded.

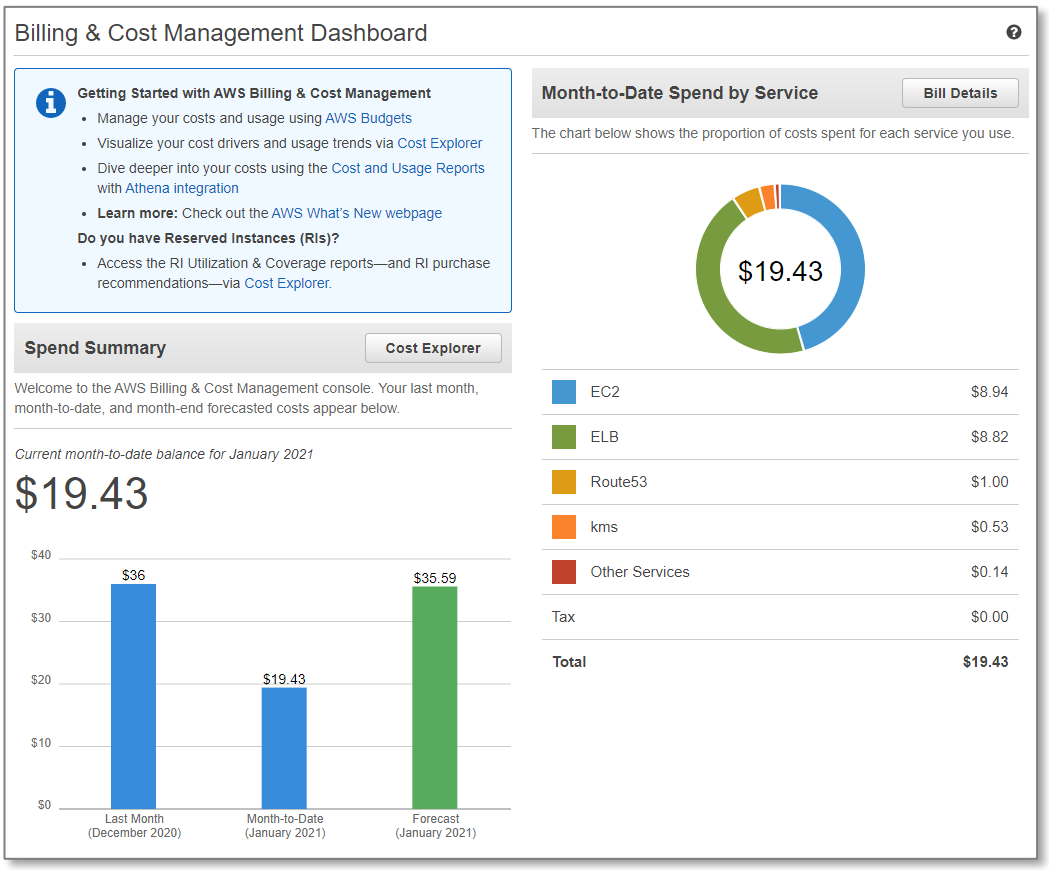
Checking your AWS Usage and Monthly Bill

From the AWS management console select *My Billing Dashboard* from the account drop down menu.



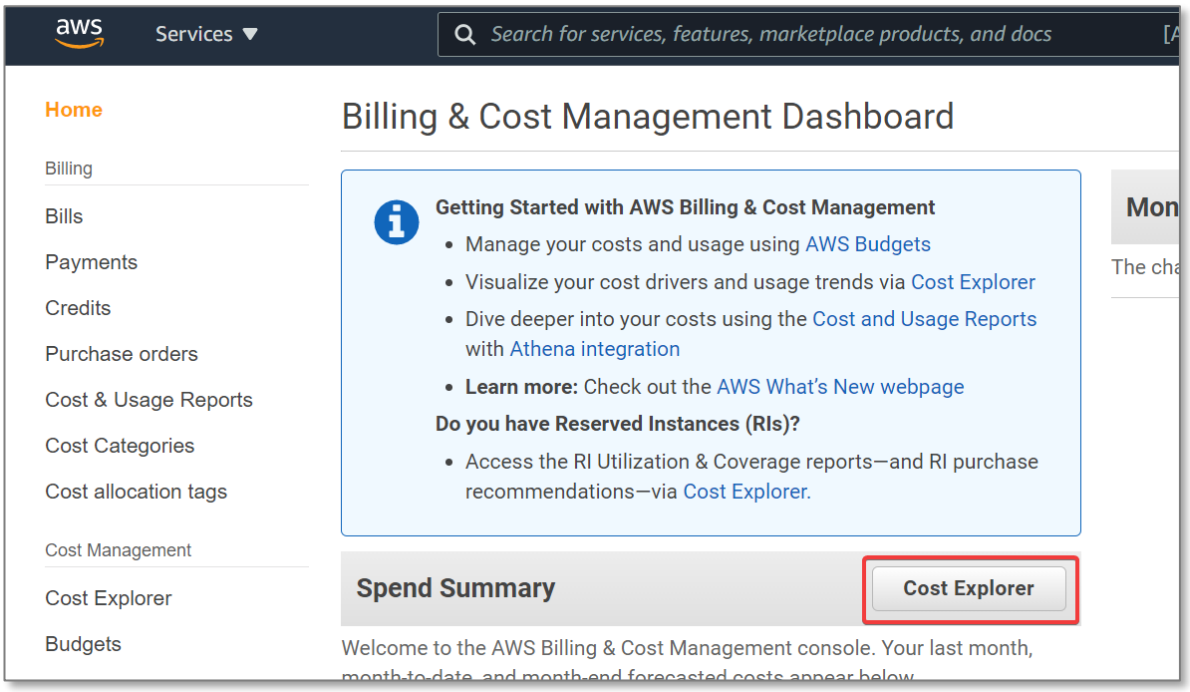
You will now see your total spend summary, spend by service and forecast spend.

Yours will of course differ from the example below.



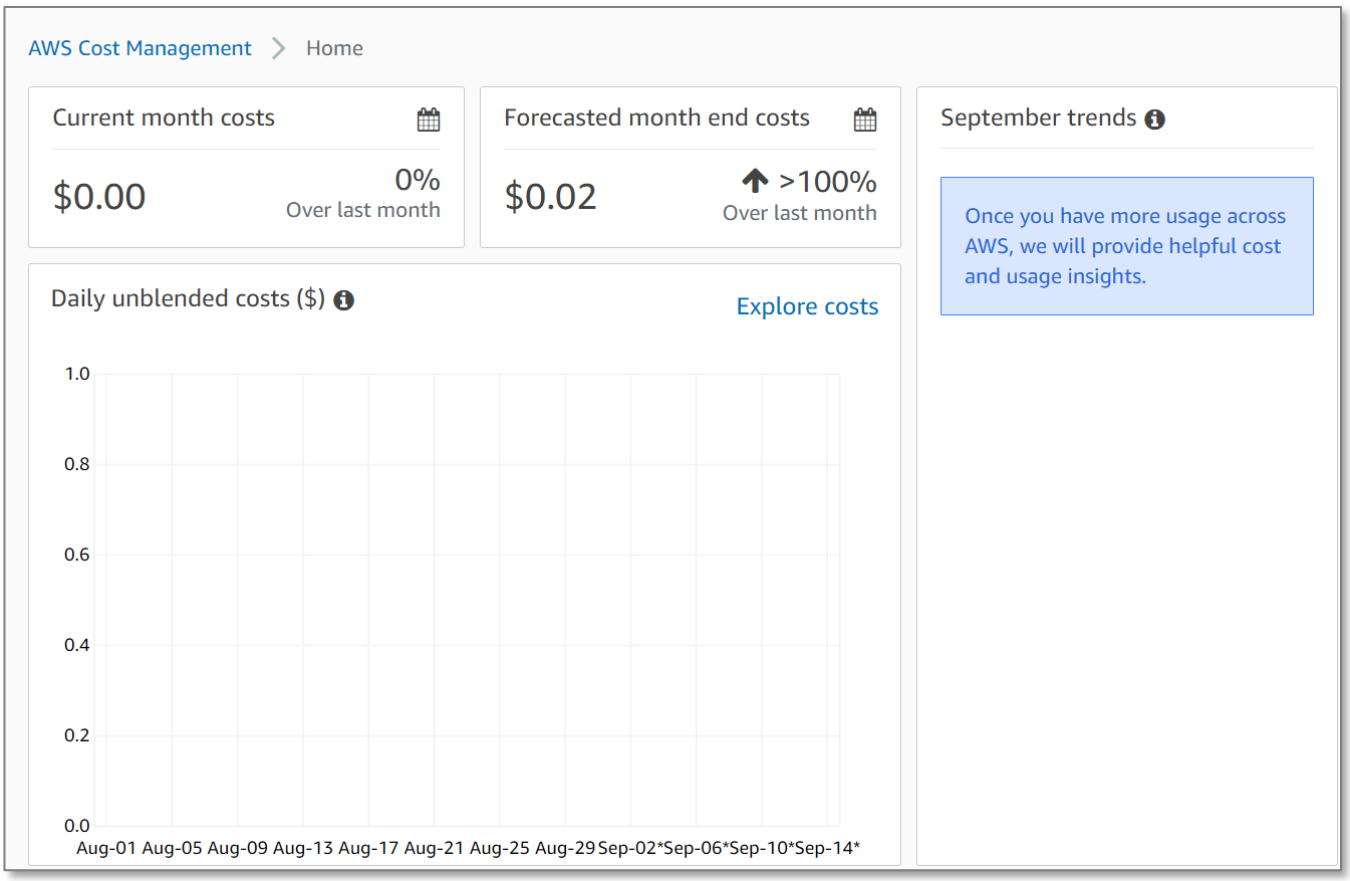
Using AWS Cost Explorer

Click *Cost Explorer*



You will now see a breakdown of costs and forecast costs.

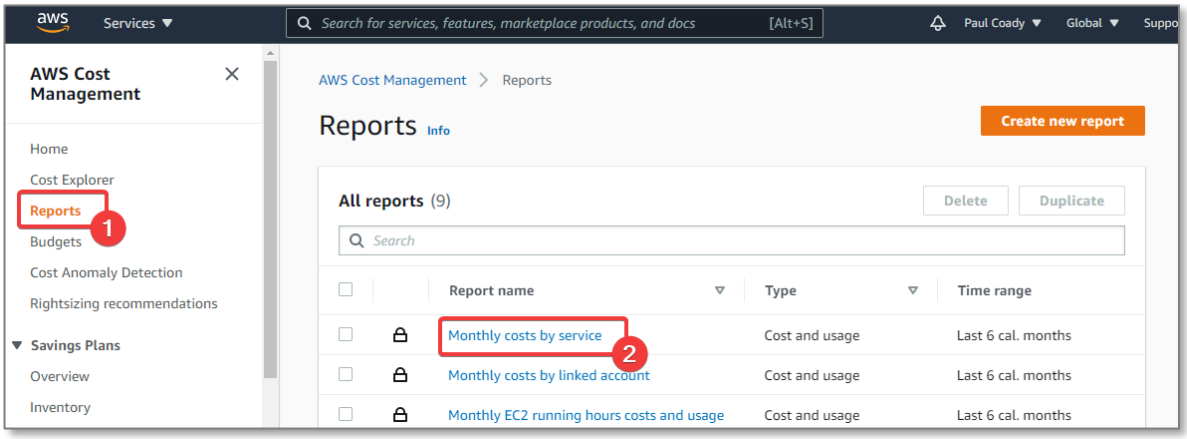
Yours will of course differ from the example below.



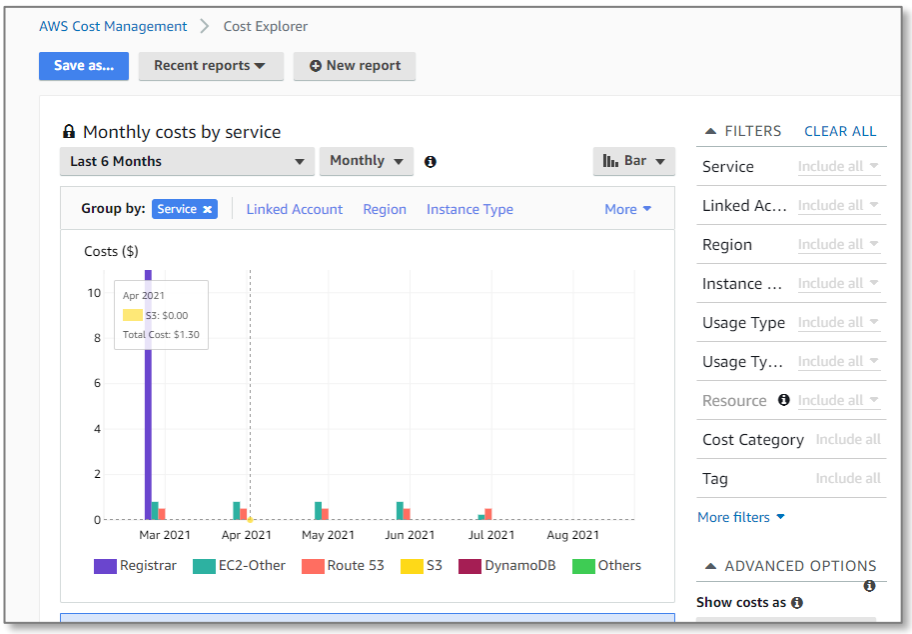
View a Costs Report

Select *Reports*

Select *Monthly costs by service*



Yours will of course differ from the example below.

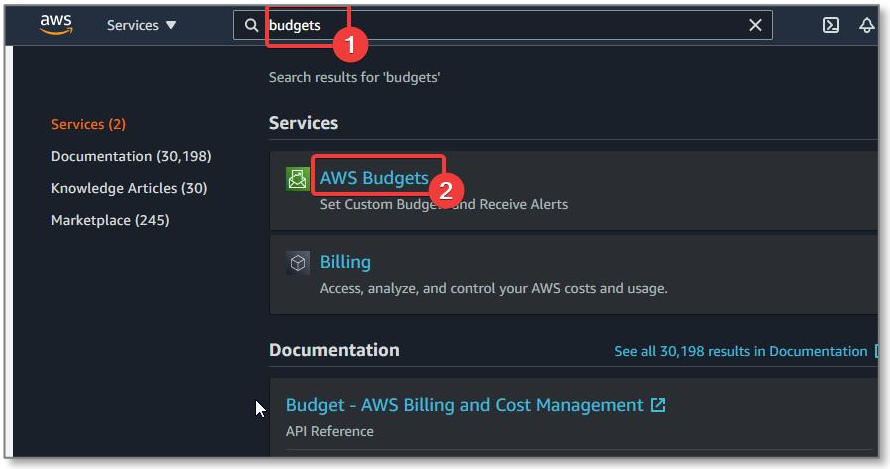


Create an AWS Budget

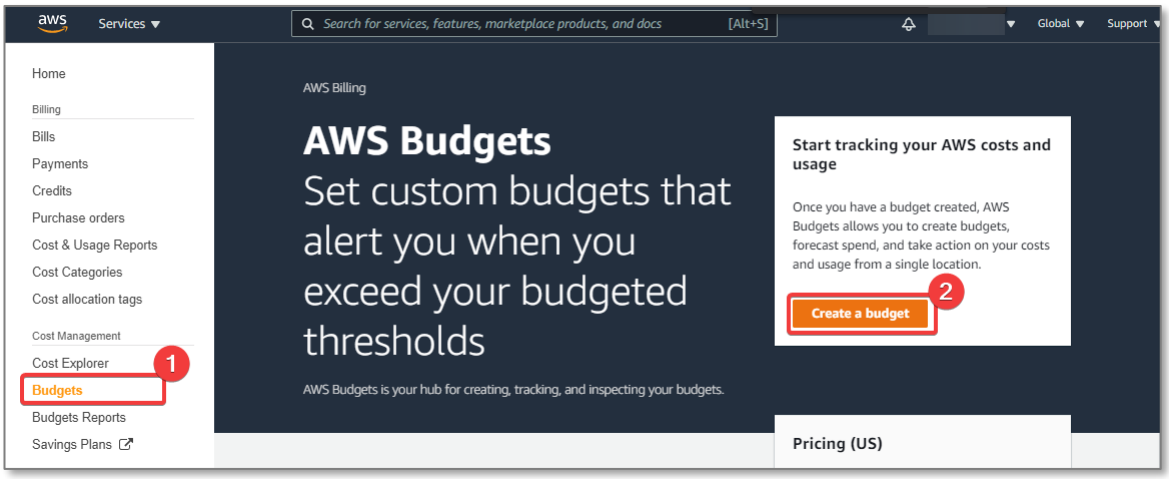
Make sure you are in US-East (N. Virginia) region.

From the AWS console search *Budgets*

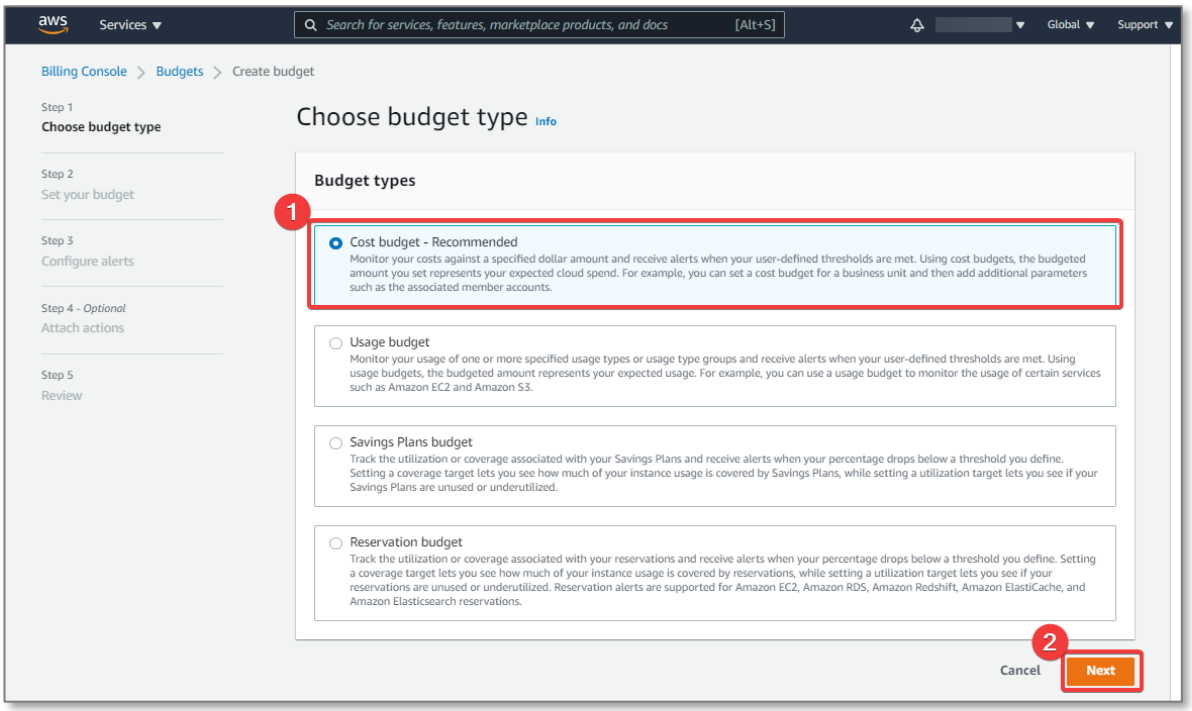
Select *AWS Budgets*



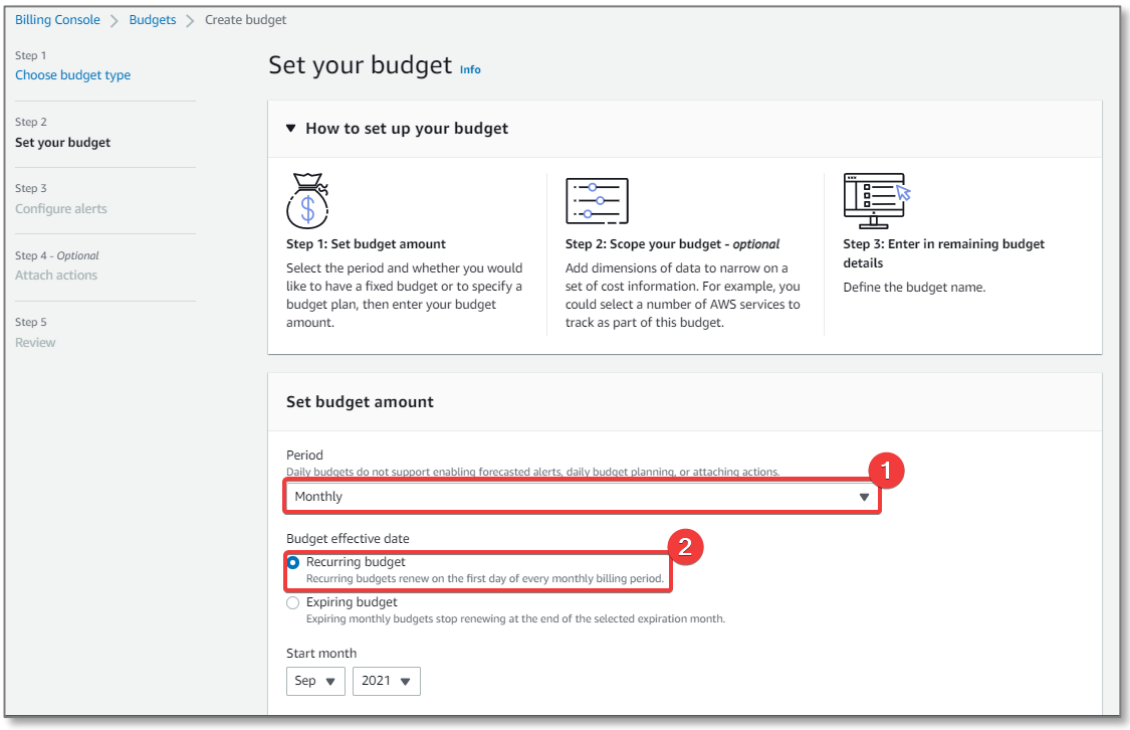
Click on *Create a Budget*



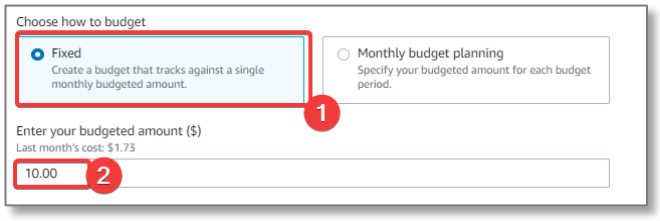
Select *Cost budget* then click *Next*



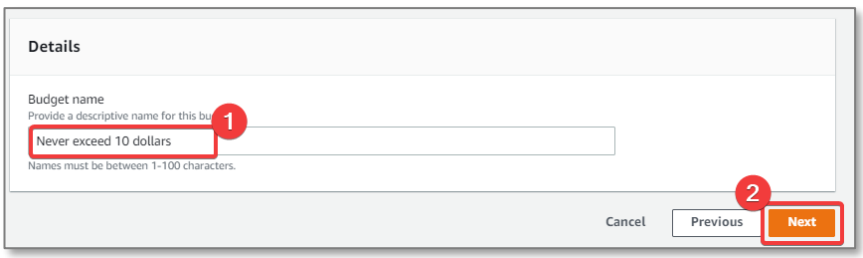
For Period select *Monthly* then *Recurring budget* for the Budget effective date



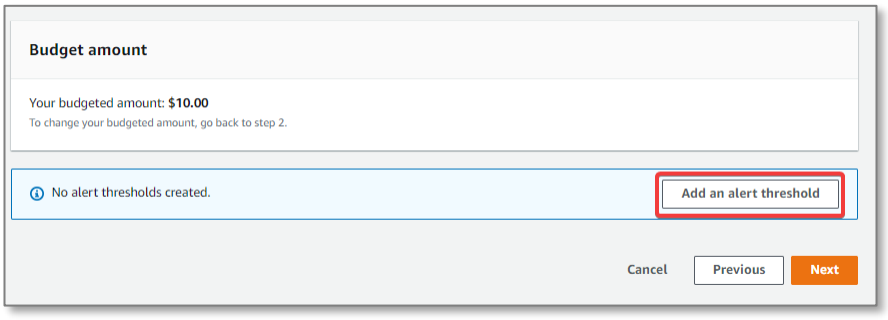
Select *Fixed* and enter your desired budget *$10.00*



Name the budget then click *Next*

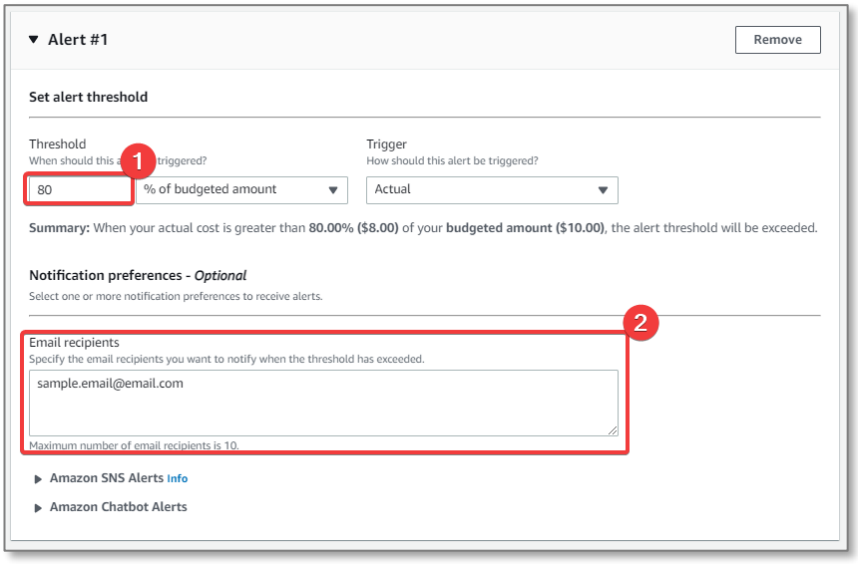


Cick *Add an alert threshold*

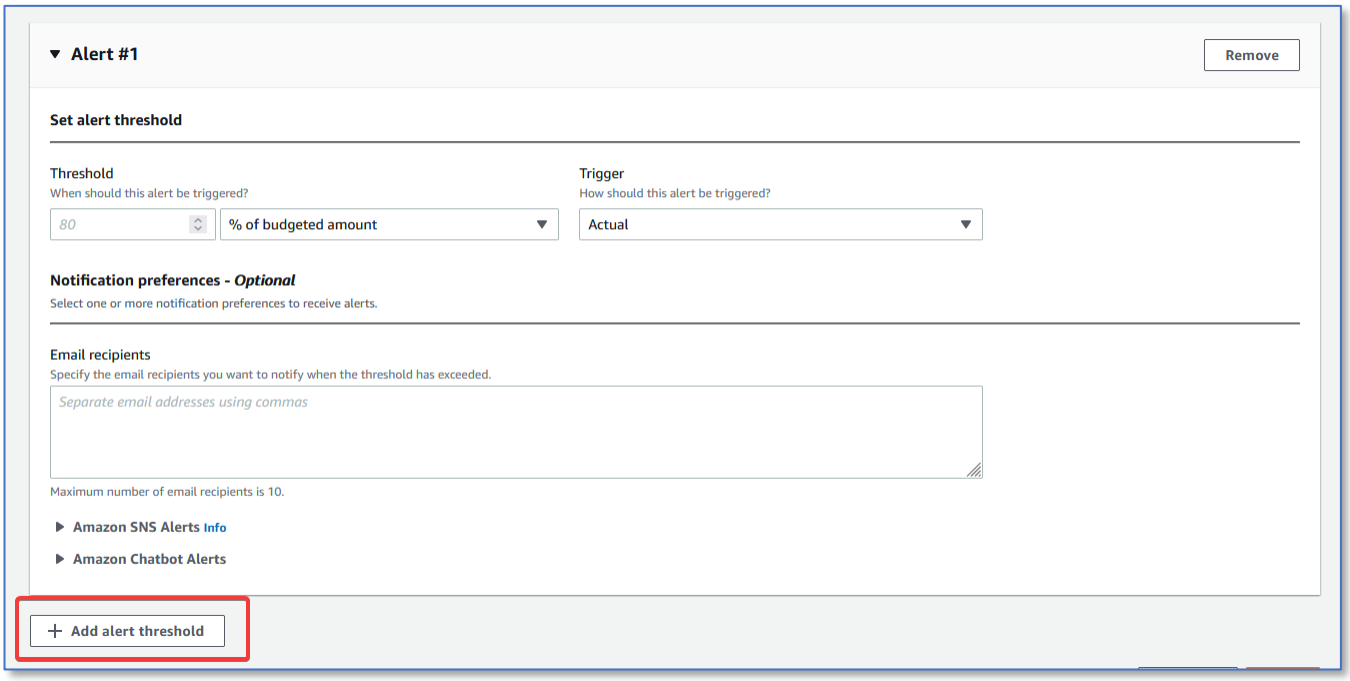


Set a number for the treshold (80) then enter email information.

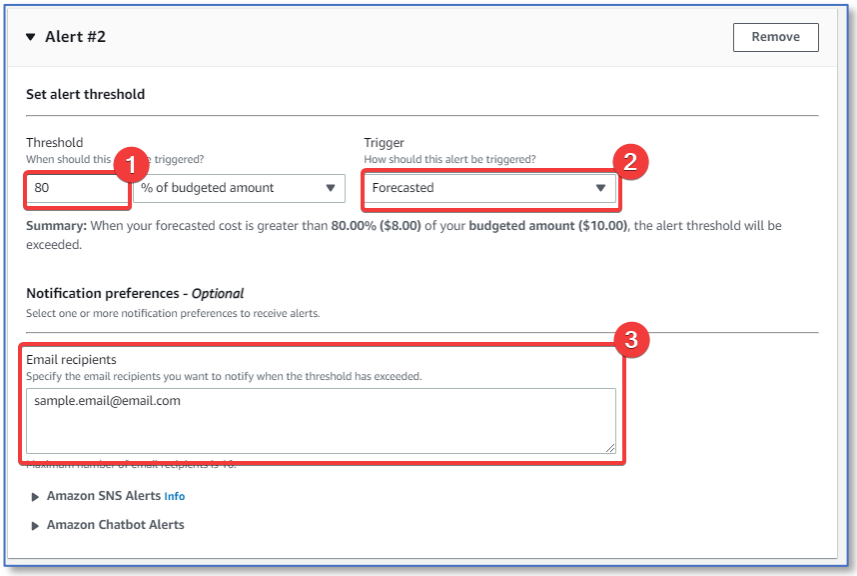
Leave *Amazon SNS Alerts* and *Amazon Chatbot Alerts* we are not using them.



Click *Add alert threshold* to create another alert

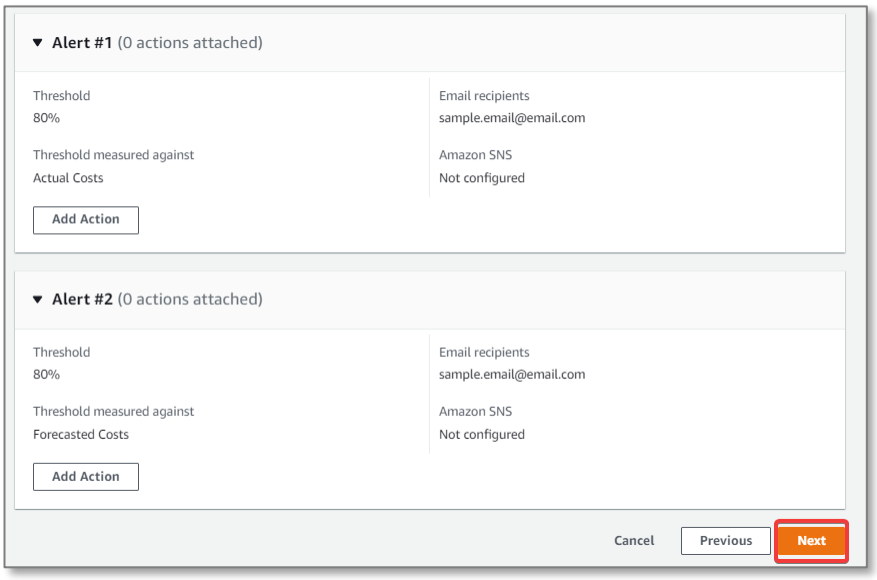


Create another alert threshold for Forcasted Trigger.
Set a number for the treshold (80) then select Forecasted.
Then enter email information.

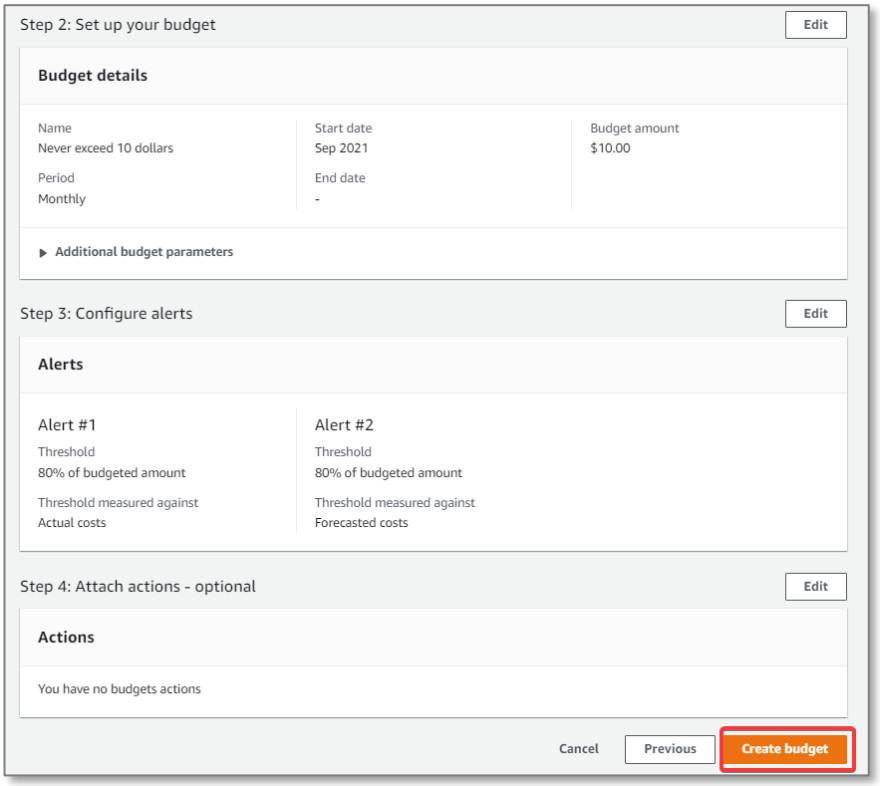


Click *Next*

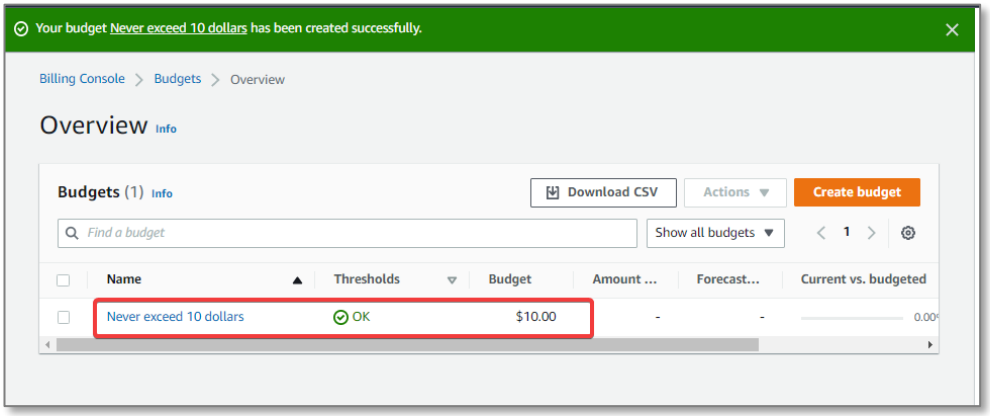
Click *Next*



Then Click Create budget



You will see now the created Budget



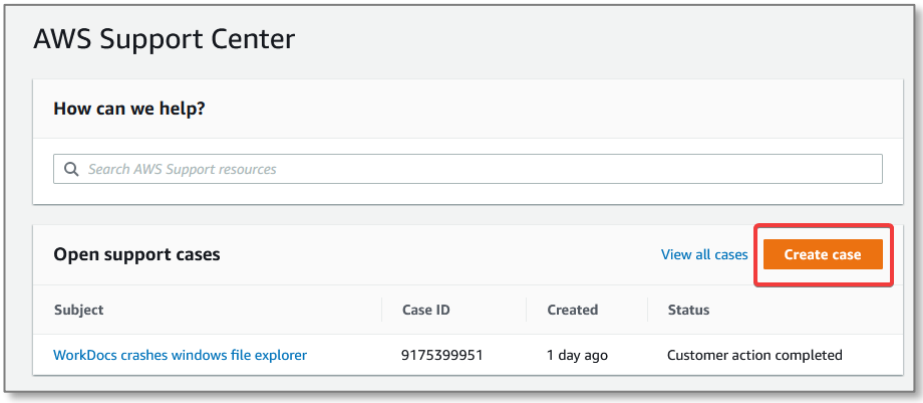
Troubleshooting

Make sure you leave *Amazon SNS Alerts* and *Amazon Chatbot Alerts,* we are not using them.

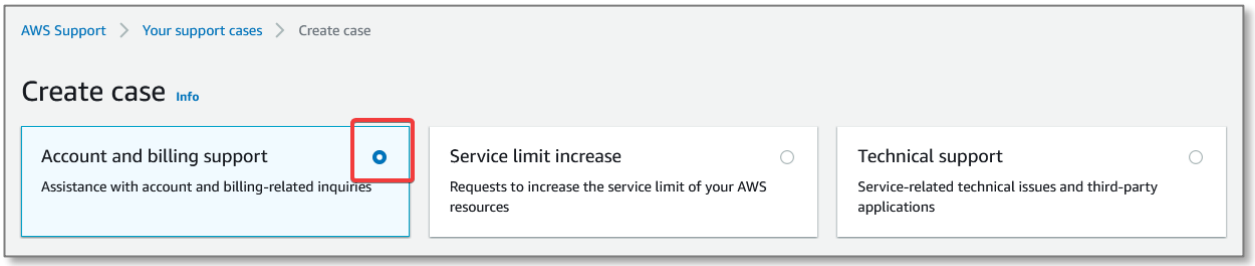
If you have followed the lab notes carefully and still cannot create your budget then there may be a problem with your account such credit card information. AWS Support is free for billing and account issues.

Go to https://console.aws.amazon.com/support

Click *Create case*

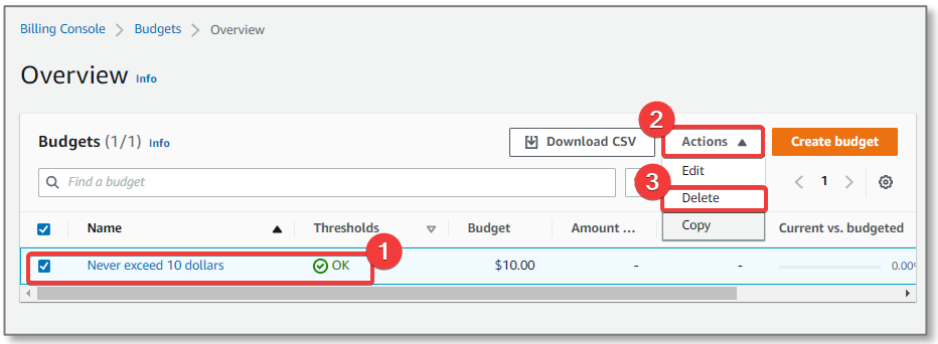


Select *Account and billing support*



Clean Up

Select the *Budgets* you created
Then Click *Actions* then *Delete*



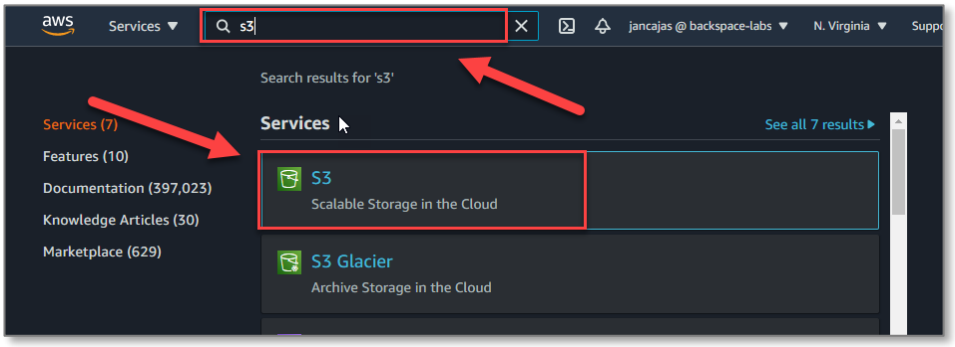
Creating an S3 Bucket and Uploading Files

In this section we will create an S3 bucket, upload files to it and download files from it.

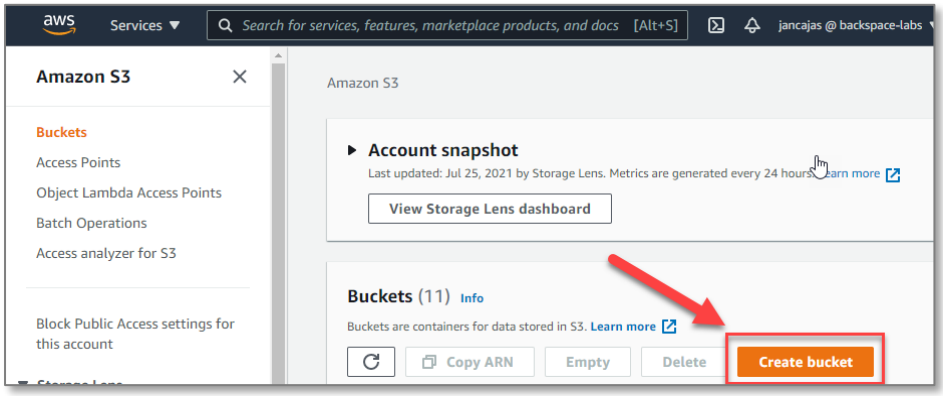
Create a Bucket

Click on the *services* menu and search *S3.*

Select S3



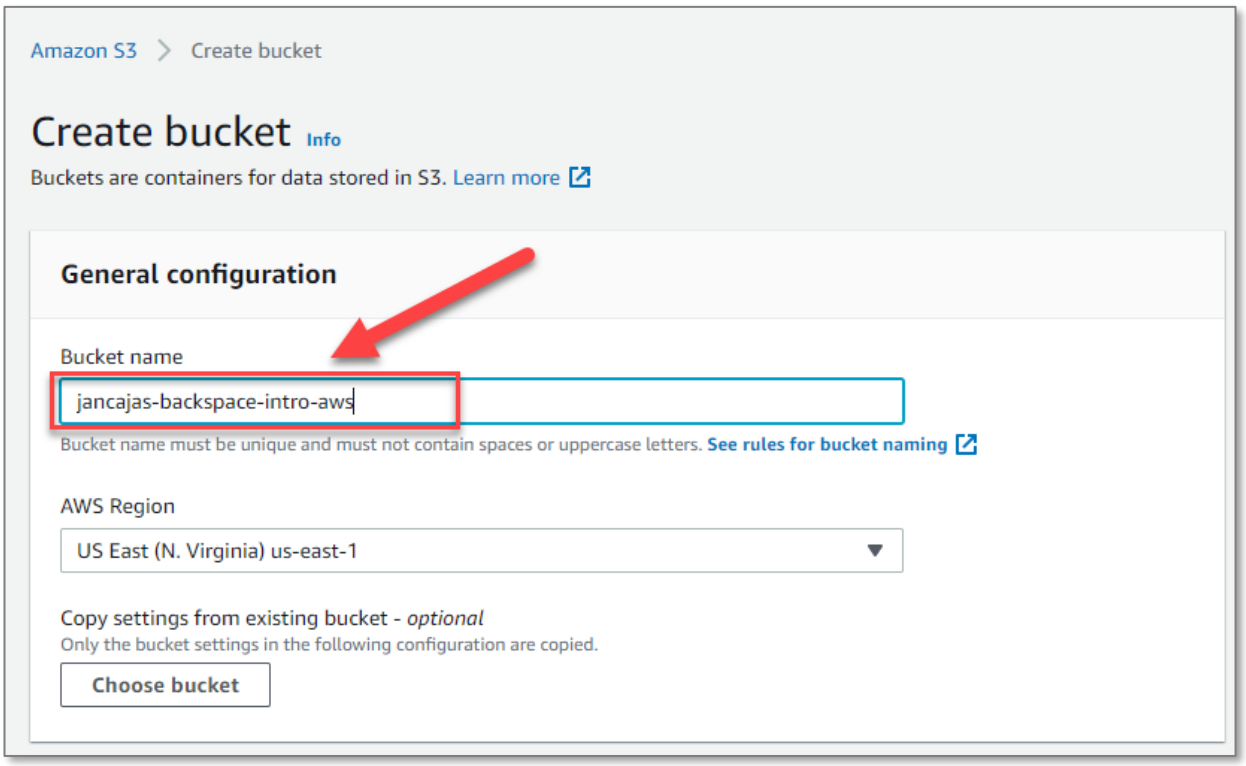
Click on *Create Bucket*



The *Create bucket* dialog box will appear.

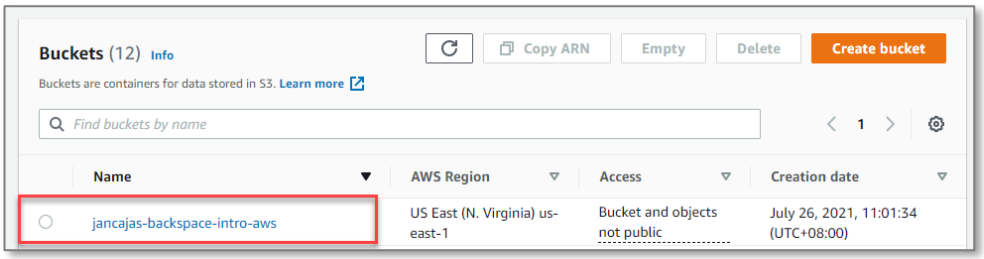
Enter a unique name for your bucket (it will need to different from the one below)

Click *Next*



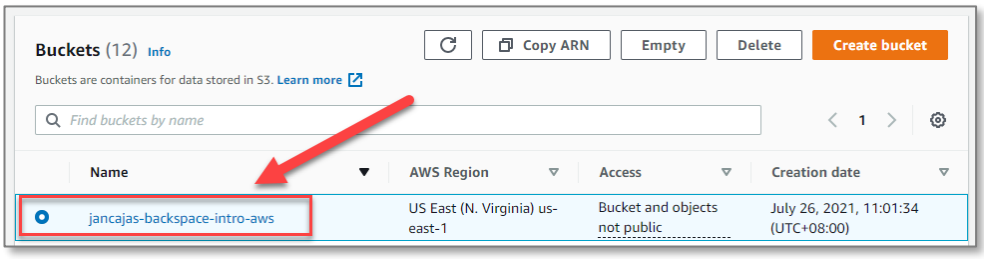
Leave other settings as is and click *Create bucket* (by default the bucket is private)

You will now see your bucket has been created.

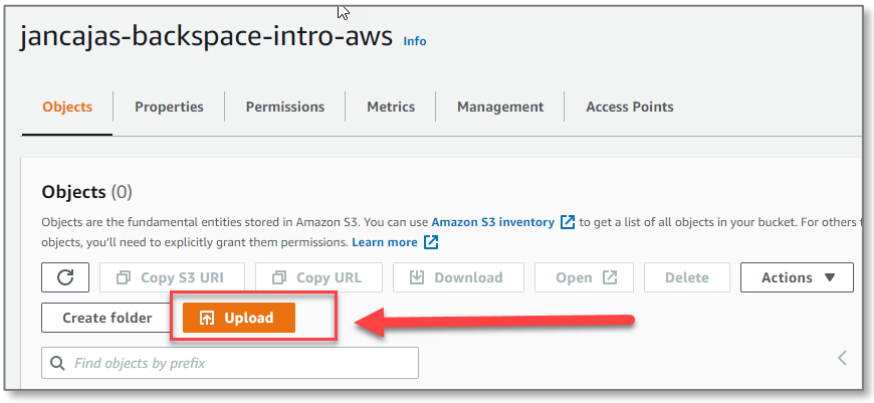


Uploading Files to your Bucket

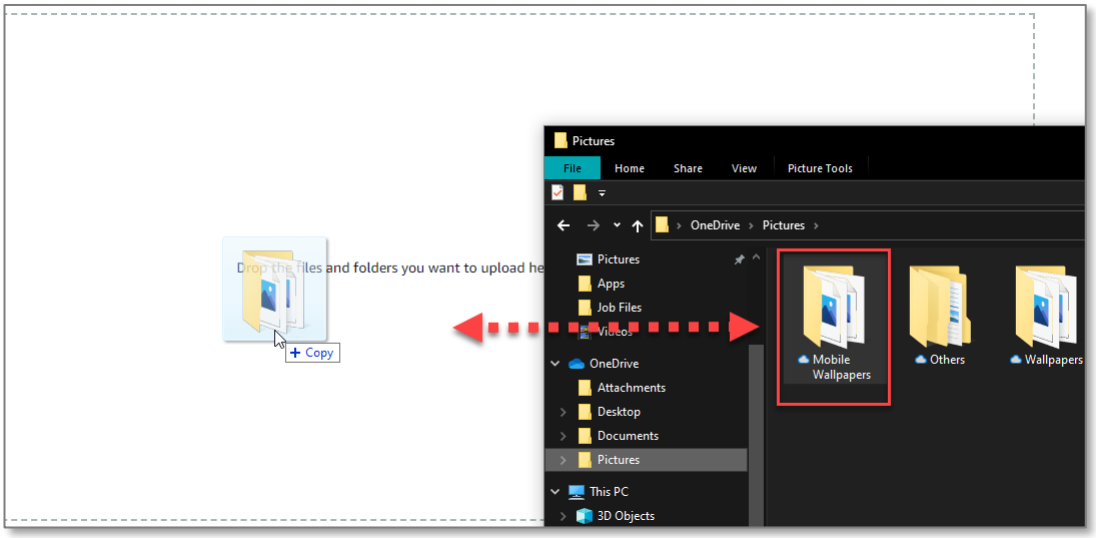
Click on the link to the bucket



Click *Upload*

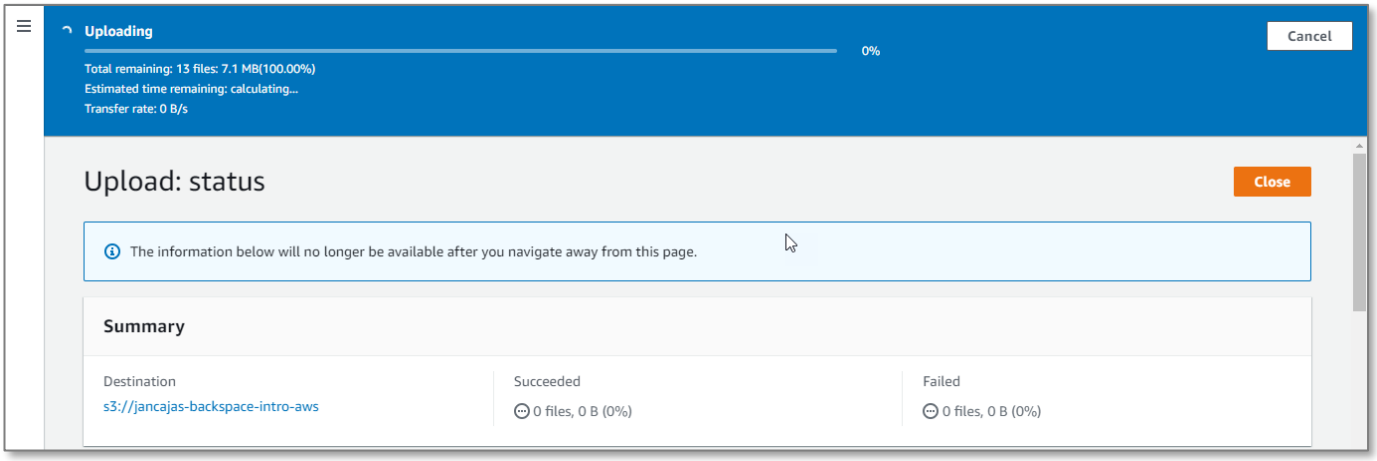


Drag a folder with files onto the form.

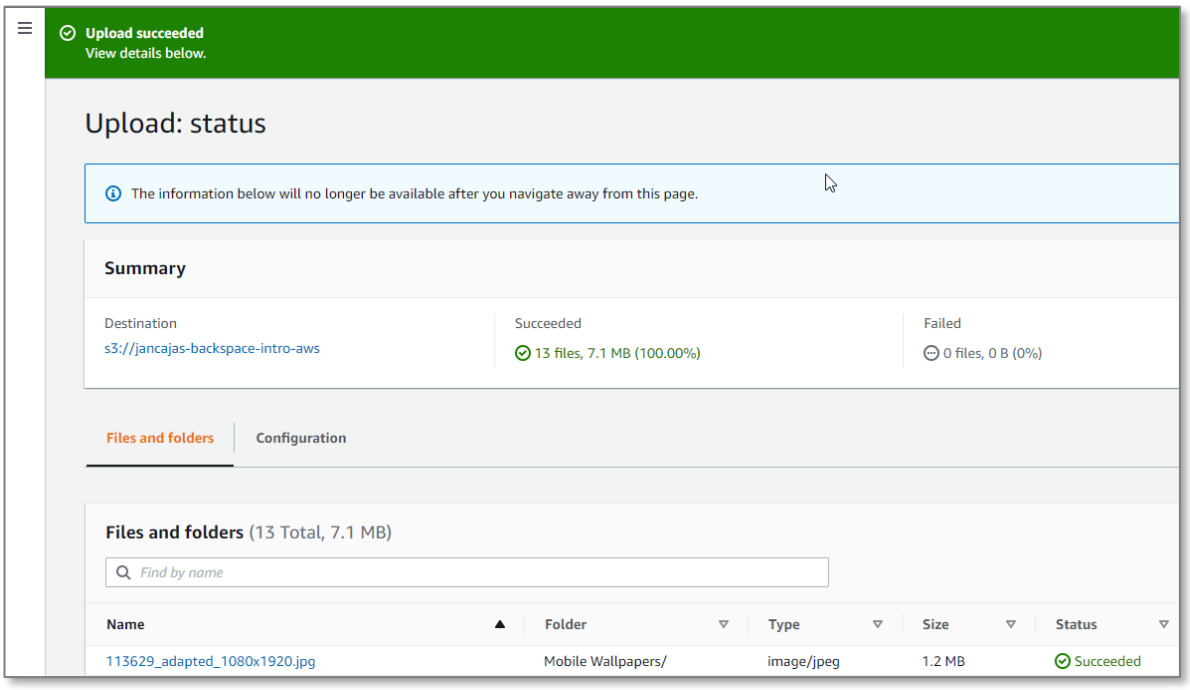


Scroll down and click *Upload*

Your files will begin *uploading*



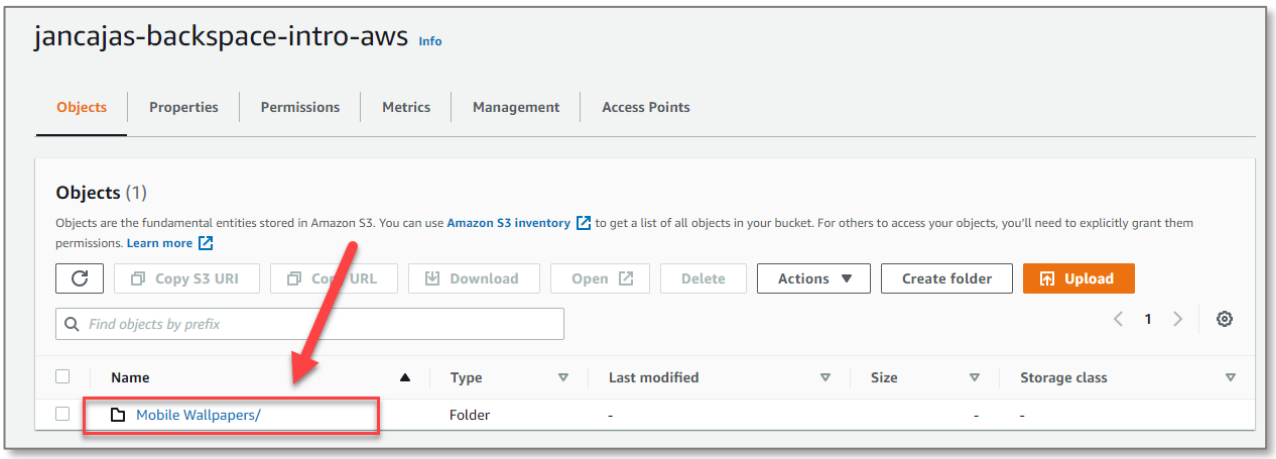
Your upload will eventually complete.



Downloading files from your bucket

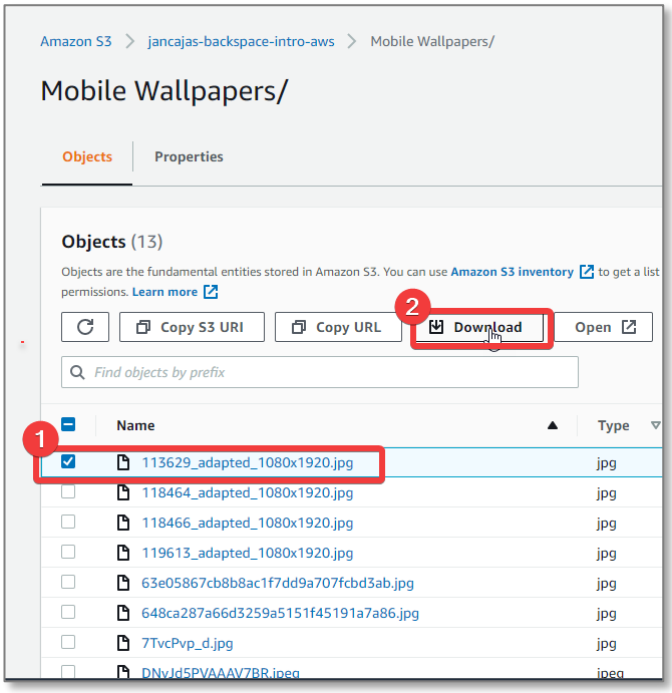
Click *Exit* to navigate back to the bucket details.

Click on the *folder* to view its contents



Select one of the files

Select -> *Download*



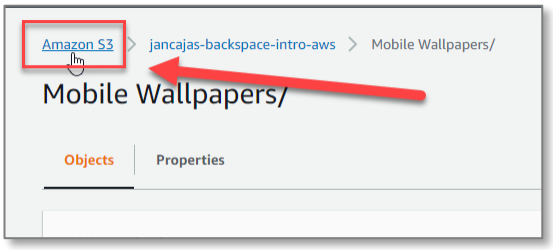
Troubleshooting

If you get the following screen, it means you have clicked on the S3 URL and not the download link as detailed above. You cannot access files directly from a URL as they have private access.

Clean Up

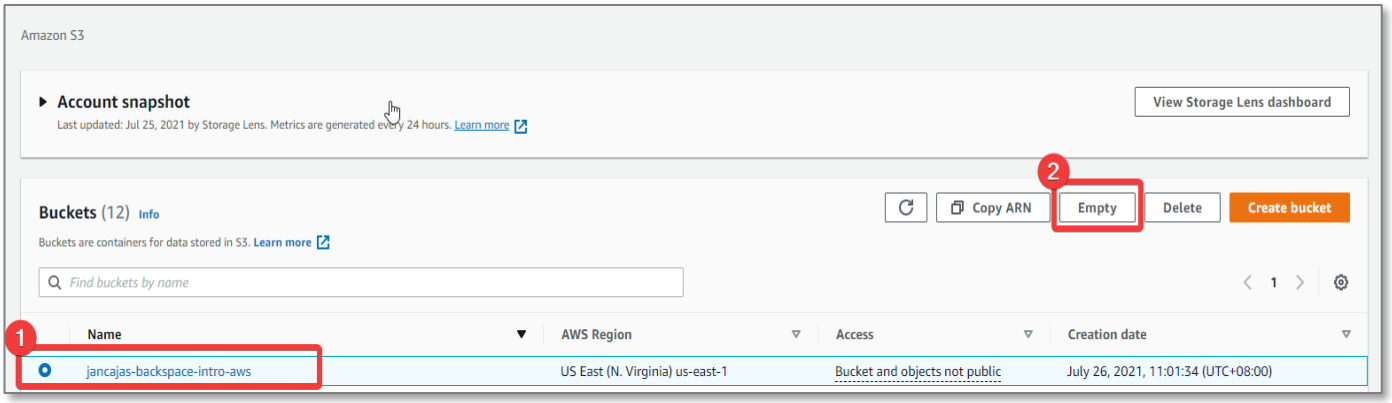
We will now delete the files and bucket so that you will not be billed by AWS.

Go back to the S3 dashboard.



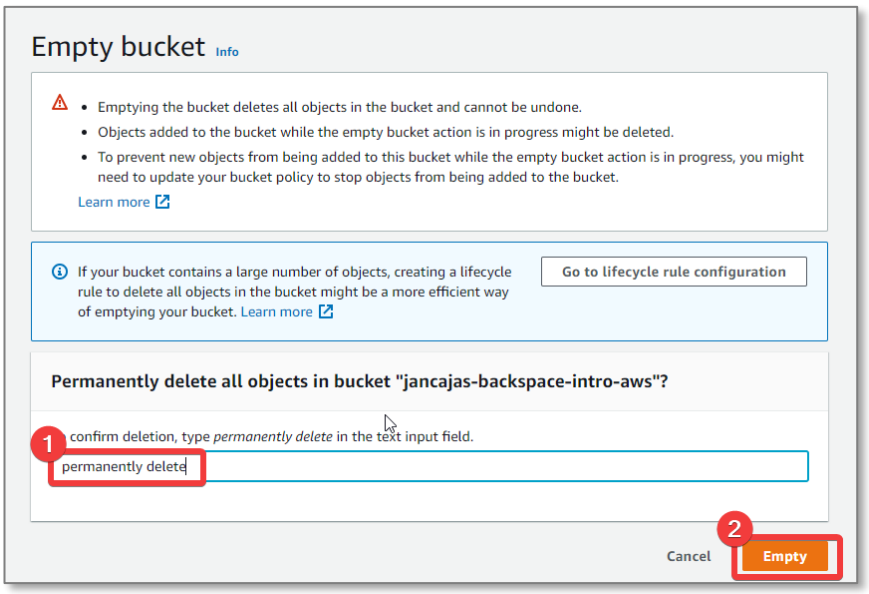
Select the *bucket*

Click *Empty*



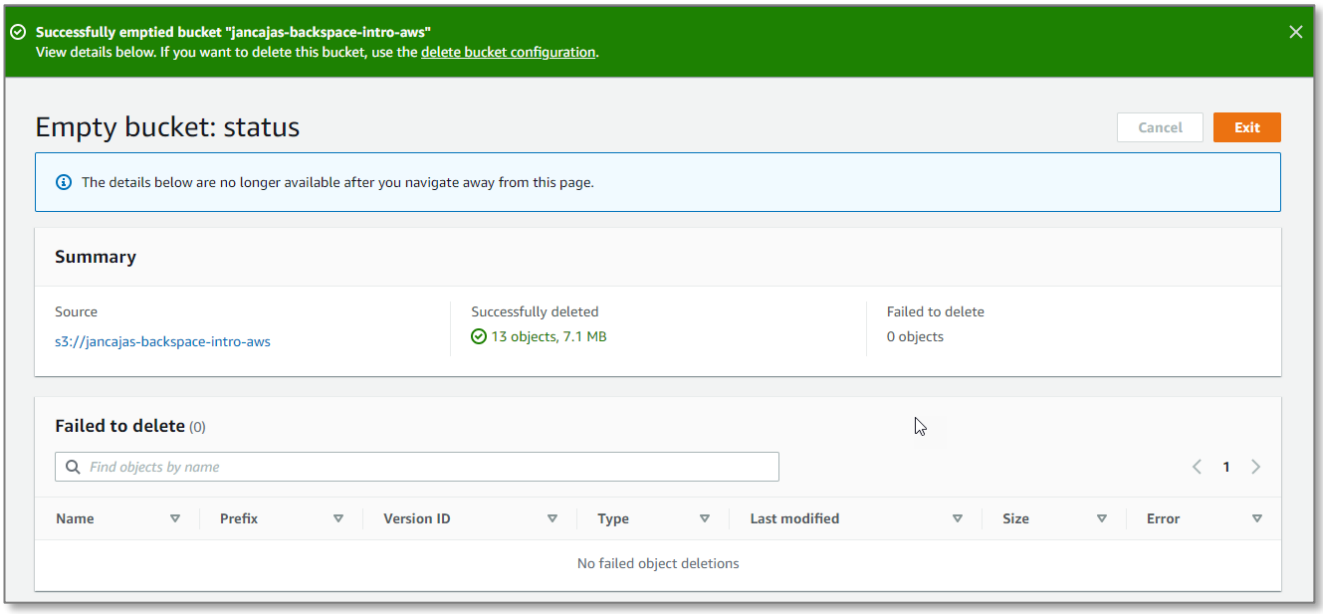
Enter *permanently delete*

Click *Empty*

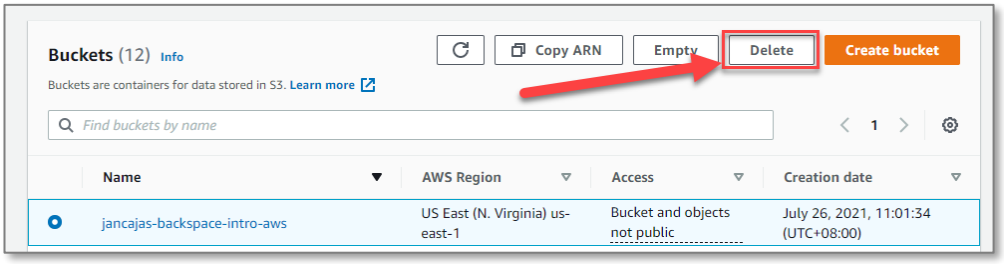


You will see a green success message

Click *Exit*

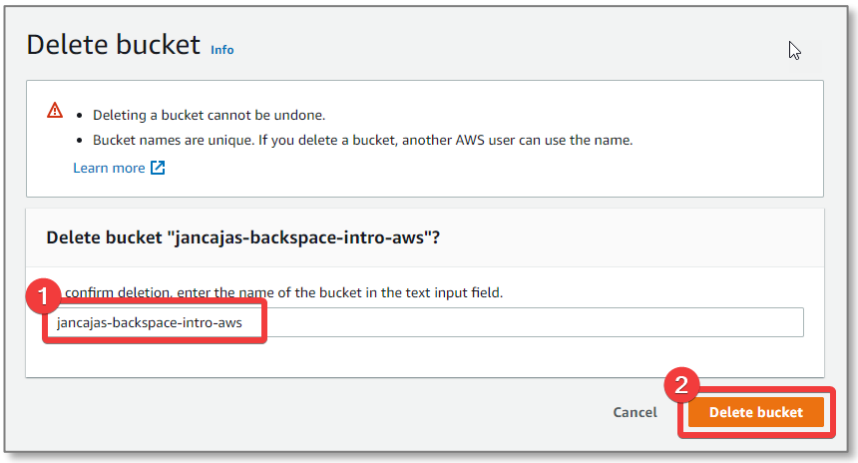


Click *Delete*



Confirm the name of the bucket to delete

Click *Delete bucket*



Creating a SQL Database with RDS

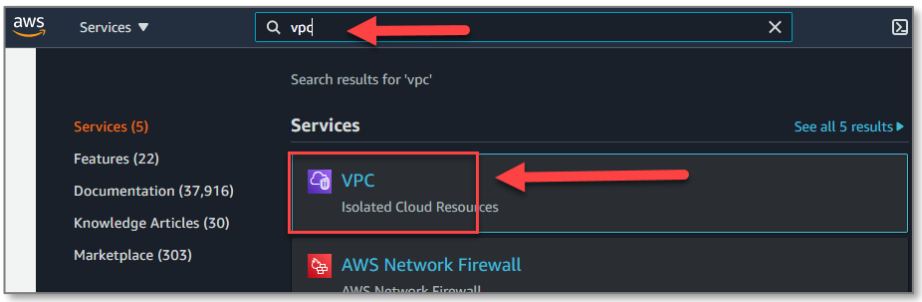
In this section, we will use the Relational Database Service to create a database. We will also connect into the database.

Creating a Security Group

By default, inbound access from the Internet to our database instance is blocked. We will create a security group that defines an inbound rule that allows access from the Internet. We can then associate this security group to our database instance.

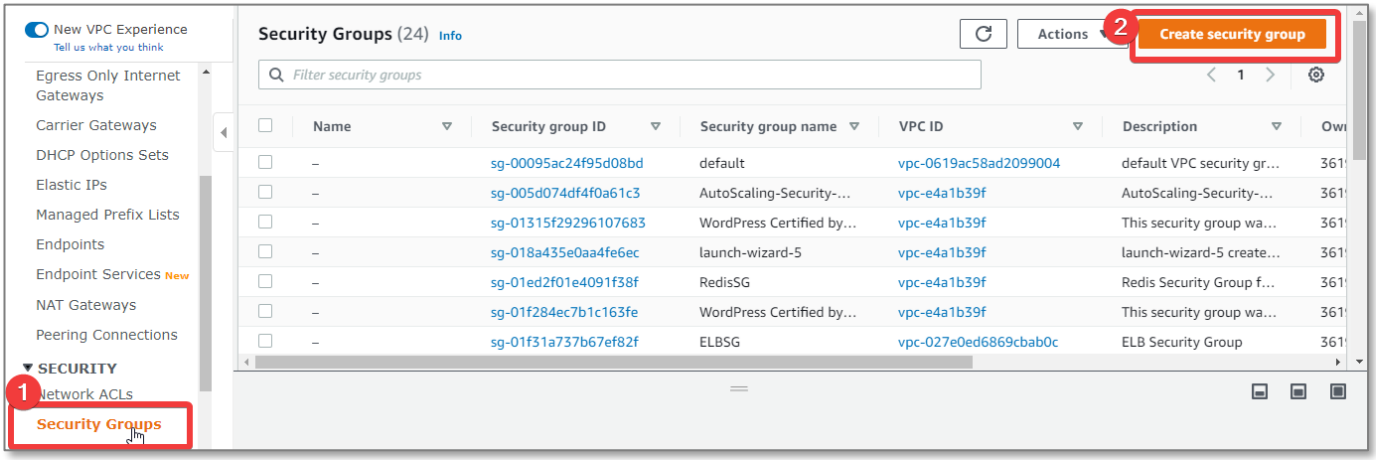
From the AWS console search *VPC.*

Select *VPC*



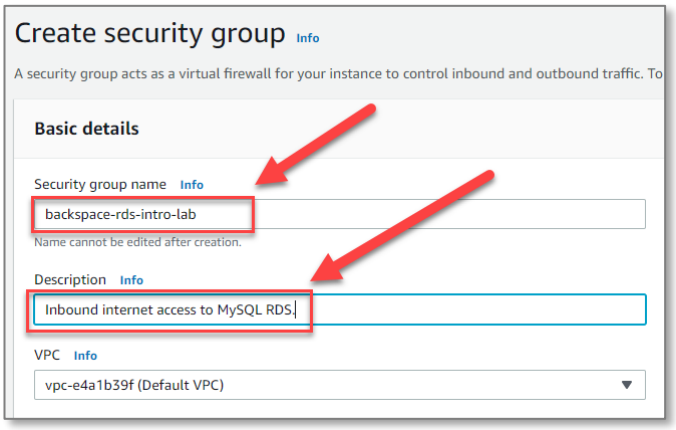
Scroll down and select *Security > Security Groups*

Click *Create security group*



Give it the name *backspace-rds-intro-lab*

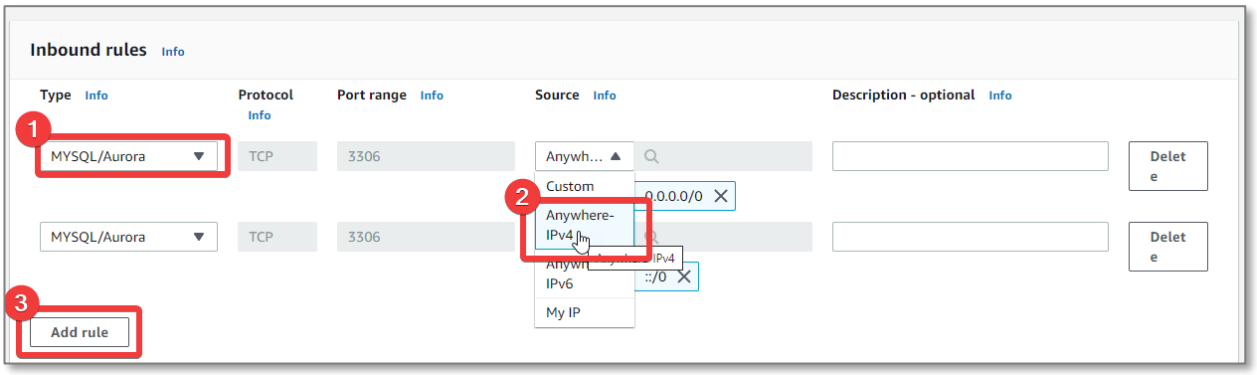
Give it a description



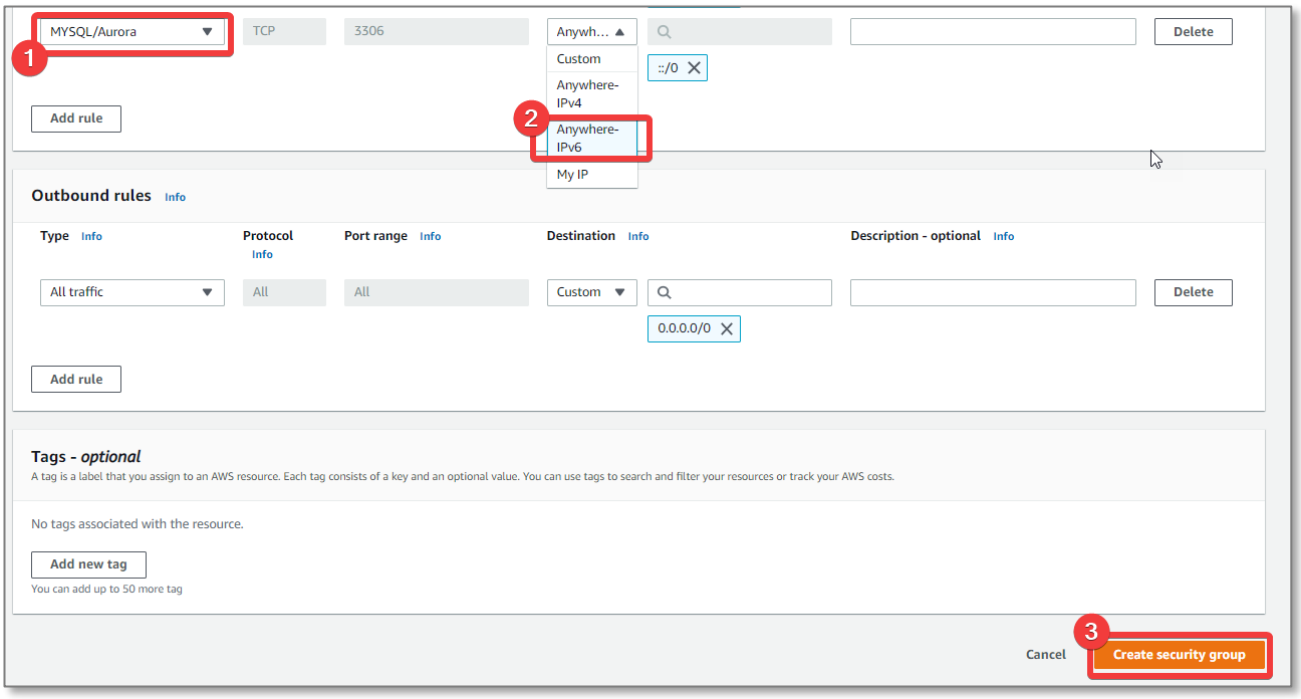
Click *Add rule* for *Inbound rules*

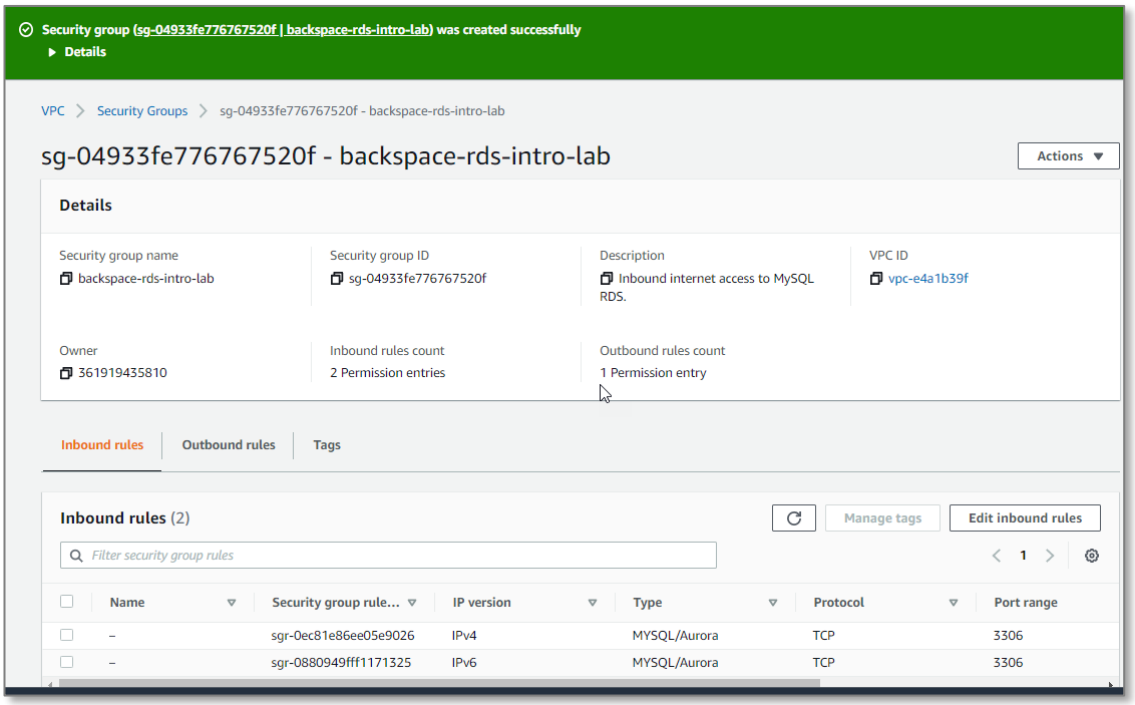
Select type *MySQL/Aurora*

Select source *Anywhere-IPv4 0.0.0.0/0* then click *add rule*



Select type *MySQL/Aurora*
Select source *Anywhere-IPv6 ::/0* then click *Create security group*

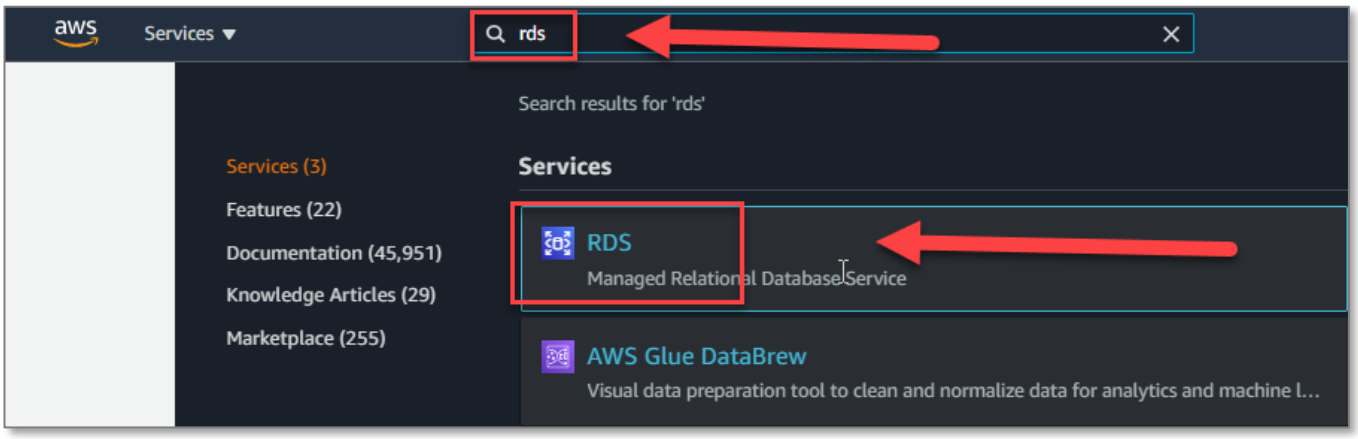




Creating an RDS Database

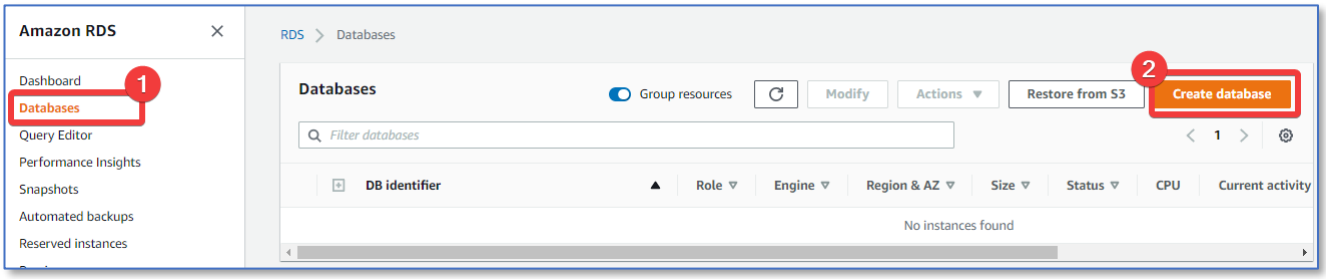
From the AWS console search *RDS*

Click on *RDS*

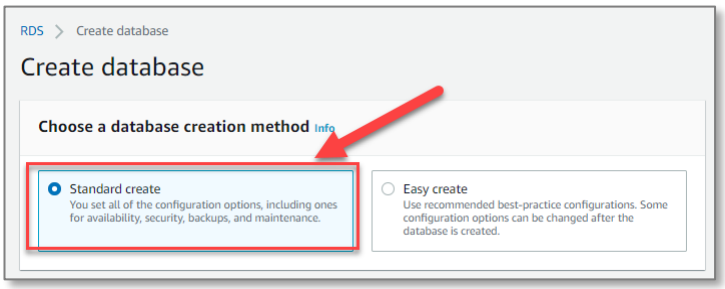


Select *Databases*

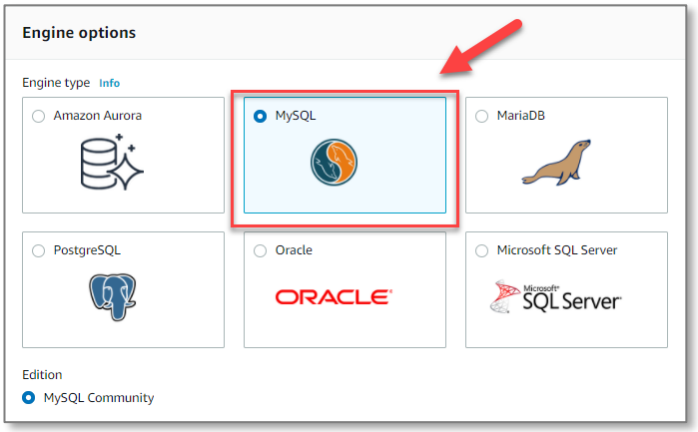
Select *Create database*



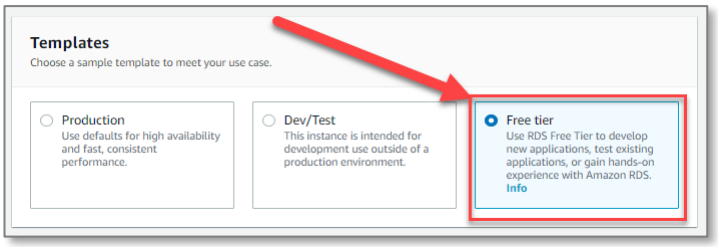
Select *Standard Create*



Select *MySQL*

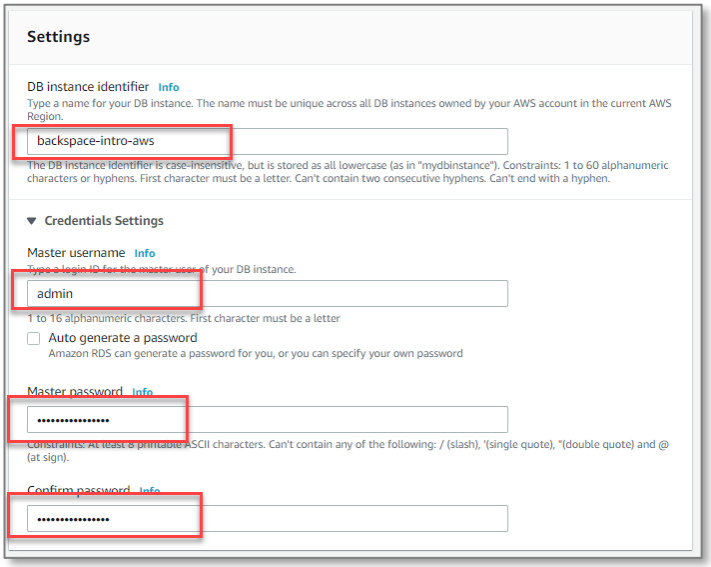


Select *Free Tier*

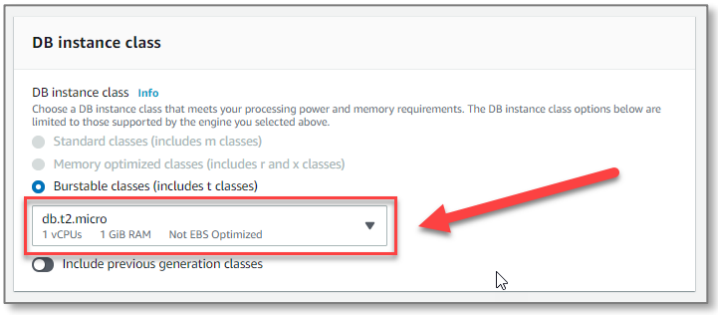


In the *Settings* section give your instance a name/identifier.

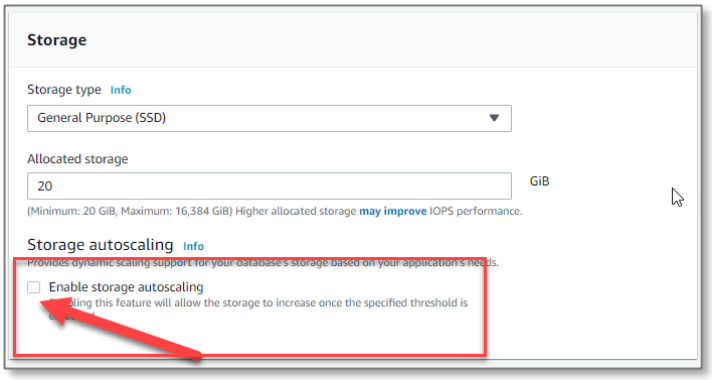
Fill in a master *username and password* (remember this we will need it later)



In the *DB Instance size* section select db.*t2.micro instance class*



Uncheck *Enable storage autoscaling*



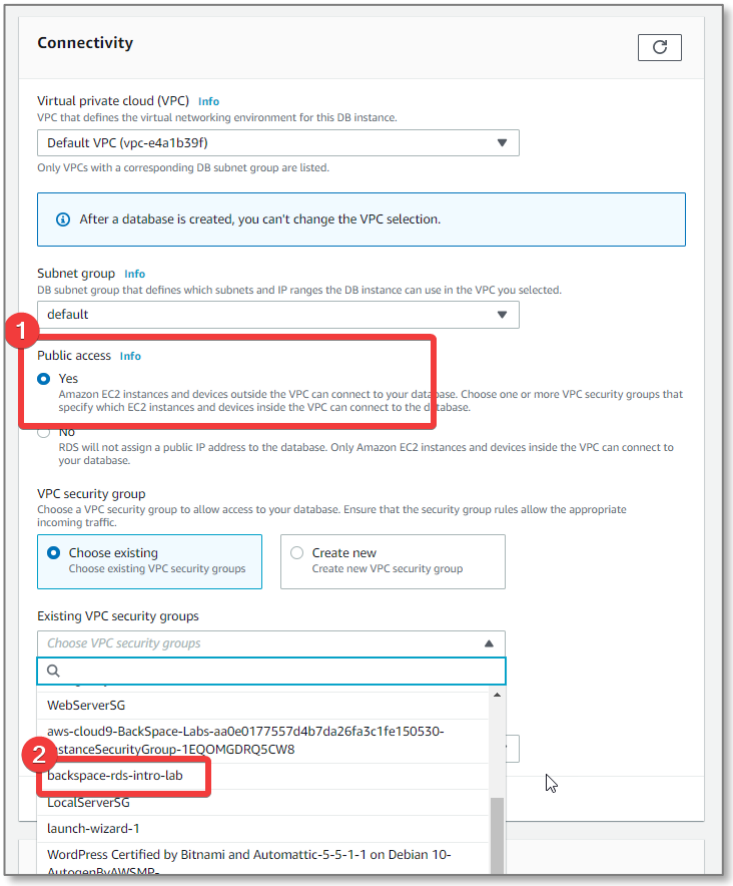
Scroll down to *Connectivity*

Expand *Additional connectivity configuration*

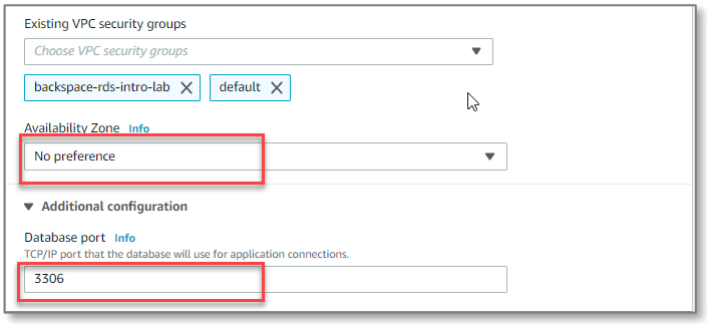
Select *yes* for *publicly accessible* (we will look at security later in the course)

Select *Choose existing for VPC security group*

Select the *backspace-rds-intro-lab* security group we created previously (click outside the list after selecting to close the list)

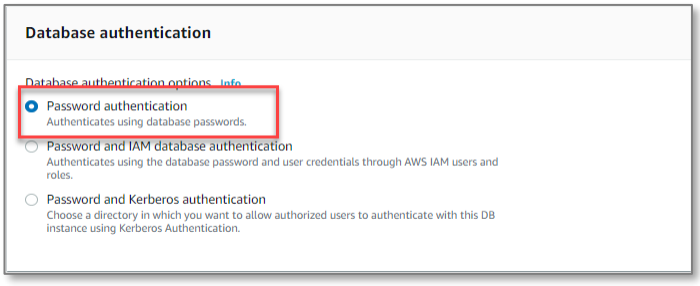


Click outside the list to add the *security group*. You should then see the *security group* added.



Scroll down to *Database authentication*

Leave as *Password authentication*

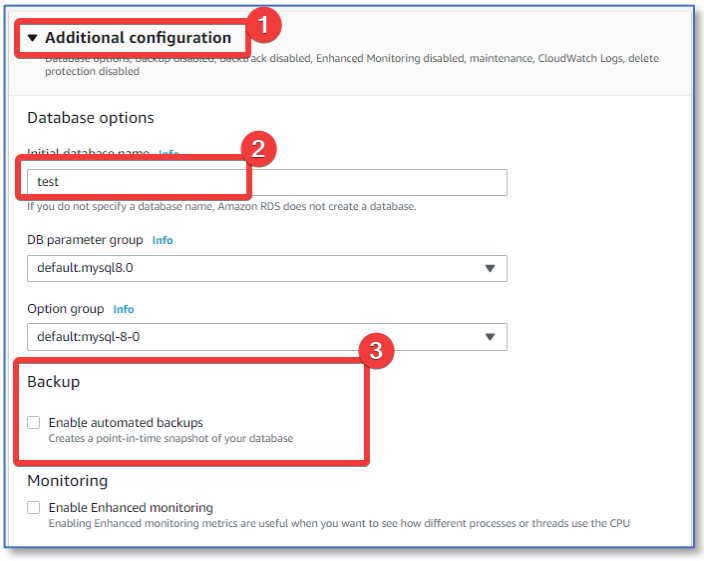


Scroll down and expand *Additional configuration*

Enter a database name.

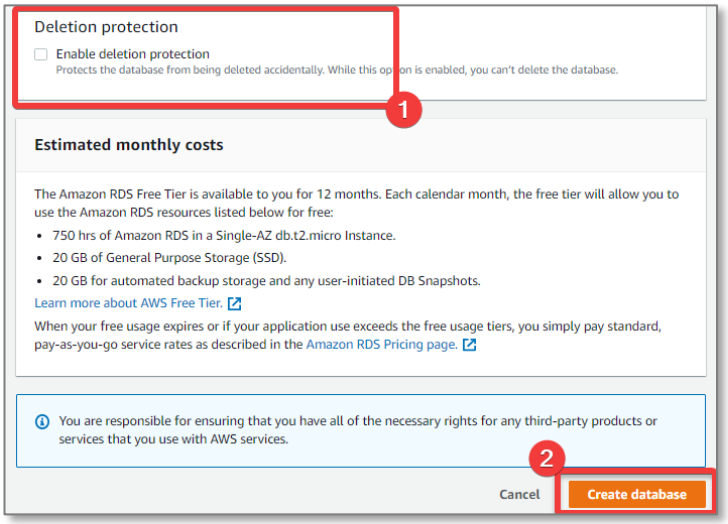
Uncheck *Enable automatic backups*

Leave all other options default.

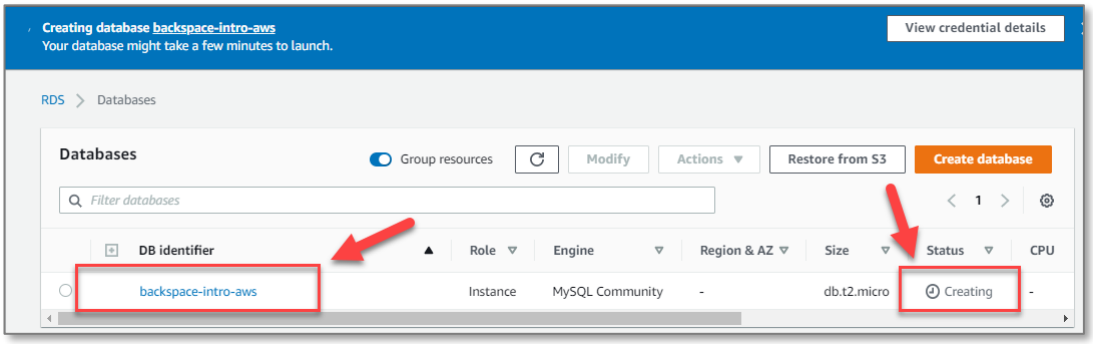


Uncheck *Enable deletion protection* (we want to delete it easily when finished)

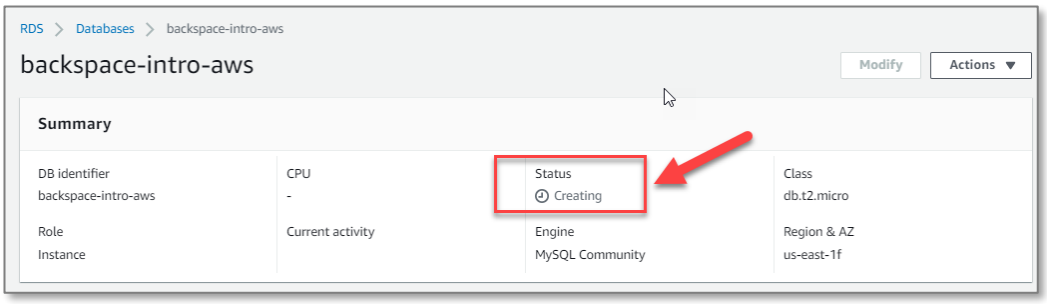
Click *Create database*



Click on the database details link



Your instance will show status *Creating*



Connecting to your RDS Instance

To connect to your MySQL Database, you will need to download and install the MySQL Workbench.

Instructions for Windows:

https://dev.mysql.com/doc/workbench/en/wb-installing-windows.html

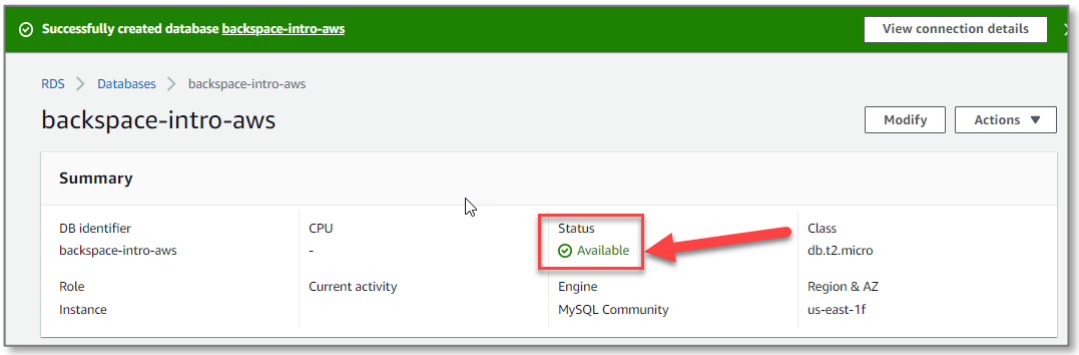
Instructions for Mac:

https://dev.mysql.com/doc/workbench/en/wb-installing-mac.html

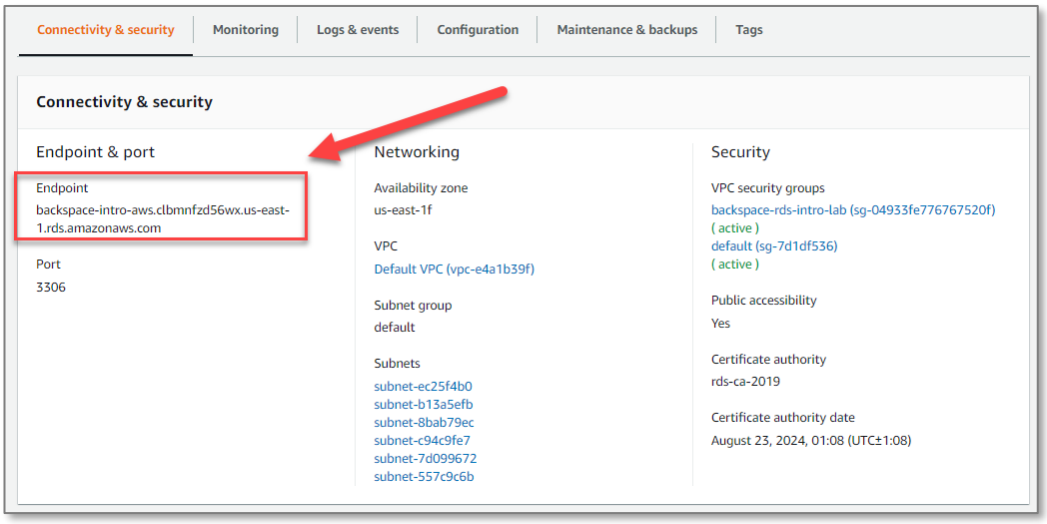
Instructions for Linux:

https://dev.mysql.com/doc/workbench/en/wb-installing-linux.html

Wait for your instance status to be *Available*



Scroll down and copy the database server *endpoint*



Open the *MySQL Workbench* application click to add a new *connection*

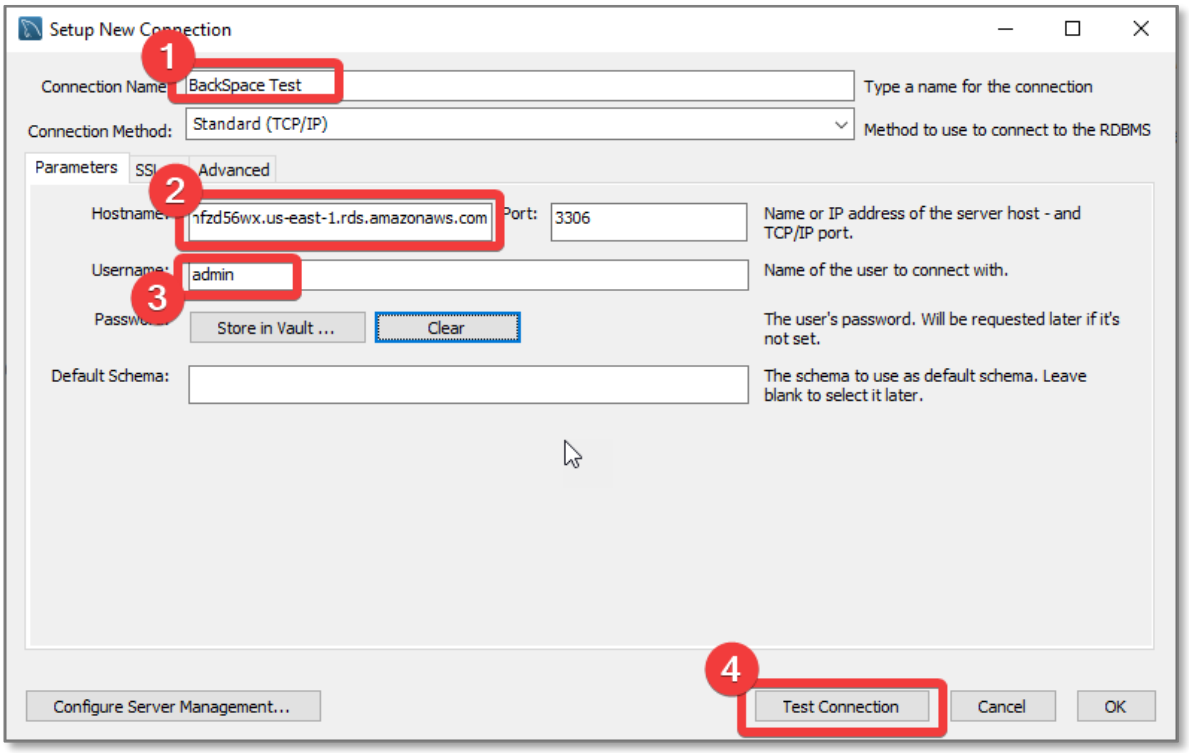
Give the connection a name.

The Hostname will be the RDS server endpoint.

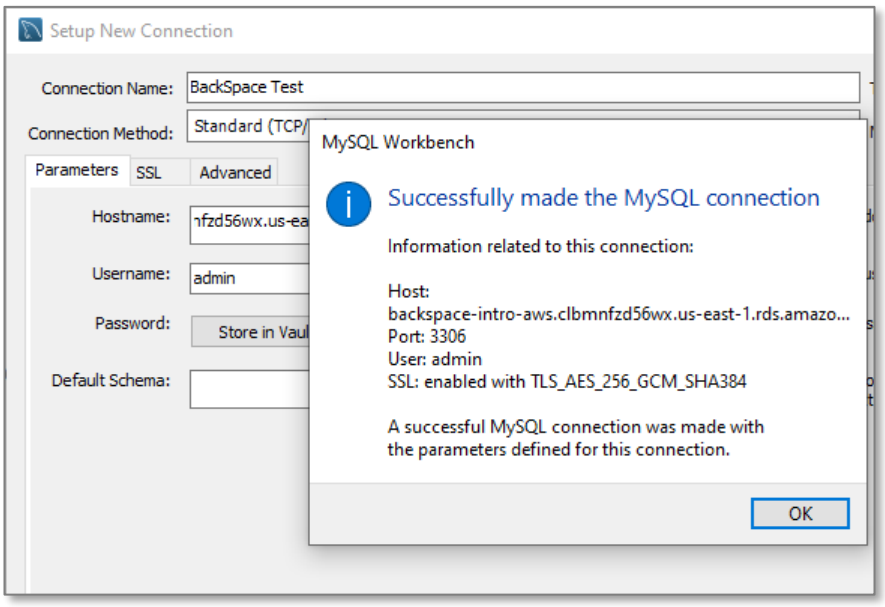
The port will be *3306.*

The Username will be the master username we created in RDS (i.e., admin)

Click *Test Connection*

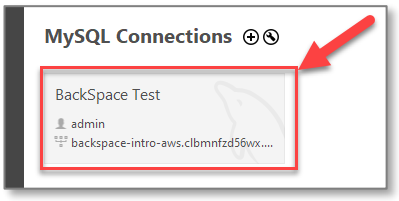


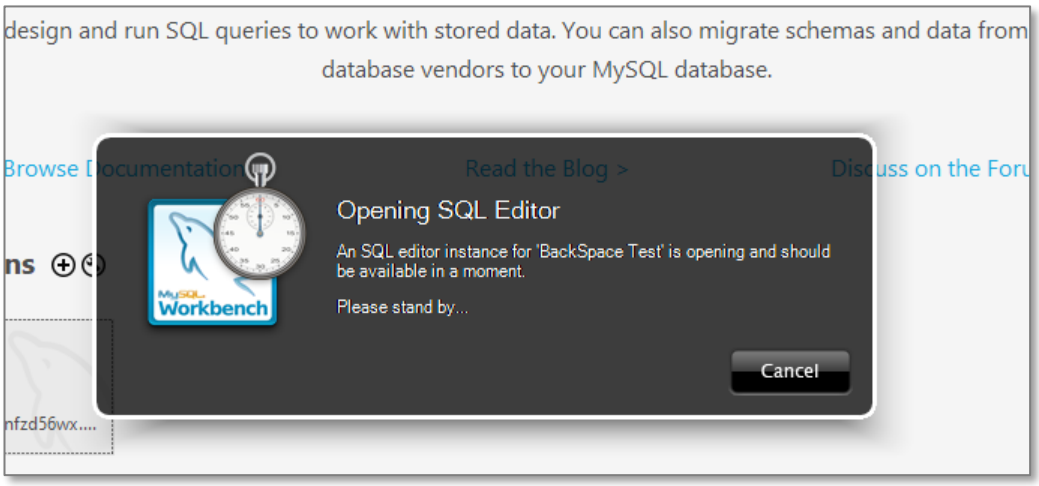
Enter the *password* you created in RDS for your master username



Click *OK*

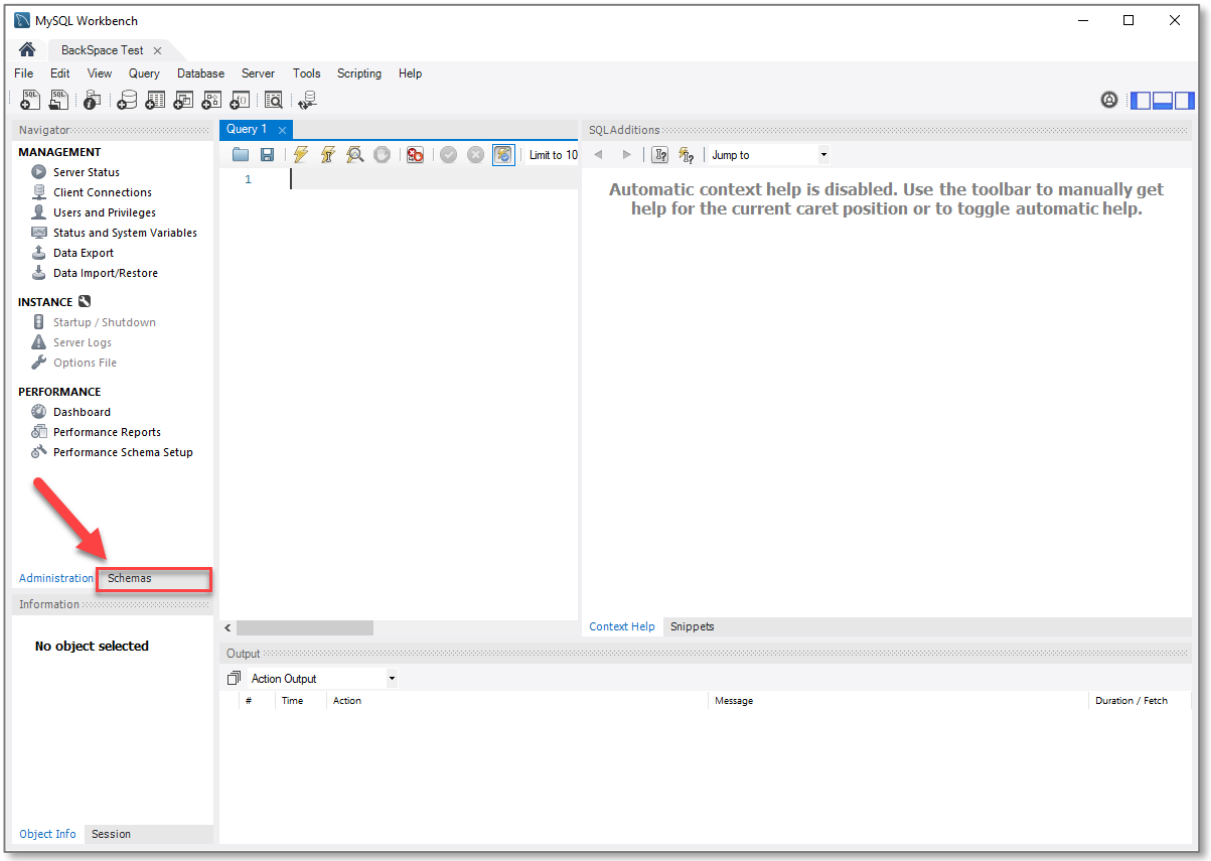
Click on the *Connection*



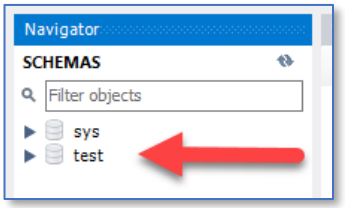


You will soon be connected to your database server

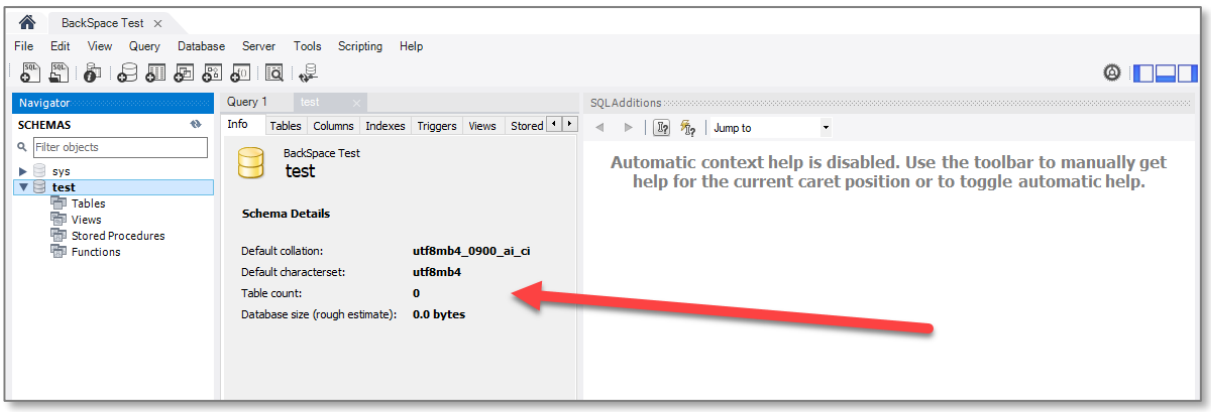
If you cannot connect then please see the “Troubleshooting Connection Issues” below.



Hover over the *"test"* database under *"SCHEMAS"* and click the information icon to get information about the database that was created by us in RDS.



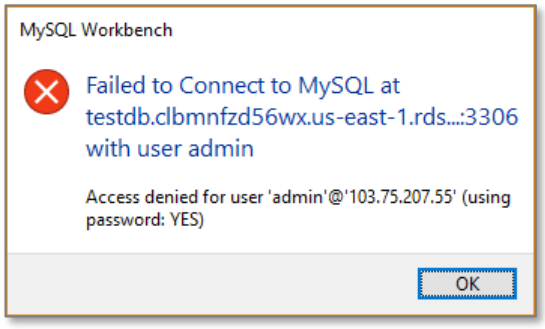
You then get an information screen for the database.



Troubleshooting Connection Issues

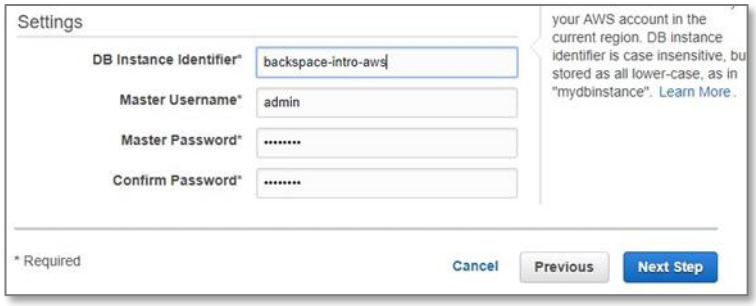
If you are getting connection errors then check the following:

Wrong Username / Password

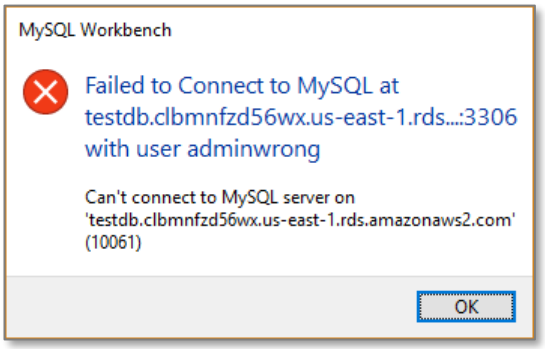


Make sure you use the correct username and password.

The username and password must be the one created when the RDS instance was created.



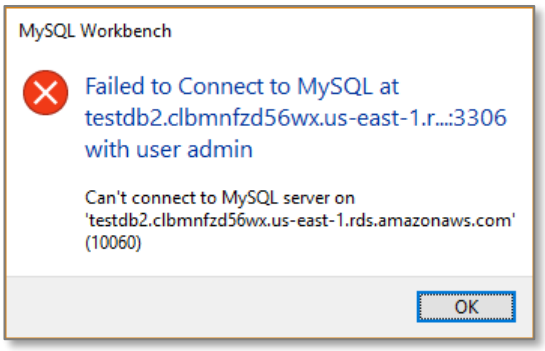
Bad Connection String



This error means nothing exists at the endpoint. Check the connection endpoint and port are correct.

The hostname will be the RDS Instance Connection Endpoint without *:3306* on the end.

No Connection



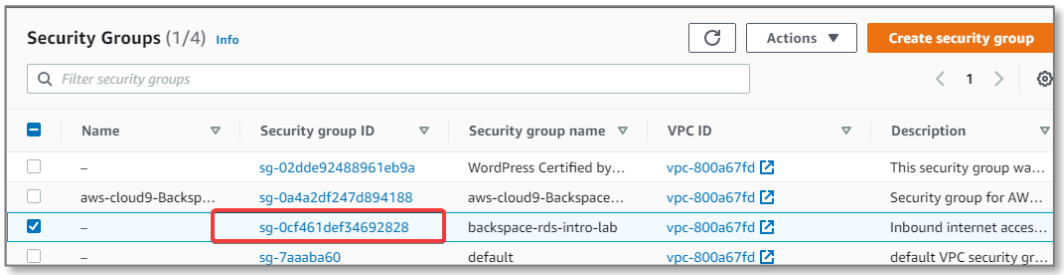
This error means your server exists but you are unable to connect to it. This can be caused by:

- You have not selected 'public' when creating instance and the security group inbound rules will be incorrect. This will block traffic to your instance. See *Security Group Inbound Rules* below.
- You have a dynamic IP address or multiple IP addresses passing through a load balancer. See *Security Group Inbound Rules* below.
- Firewall at your end is blocking access to port 3306. See *Client-side Firewall* below.

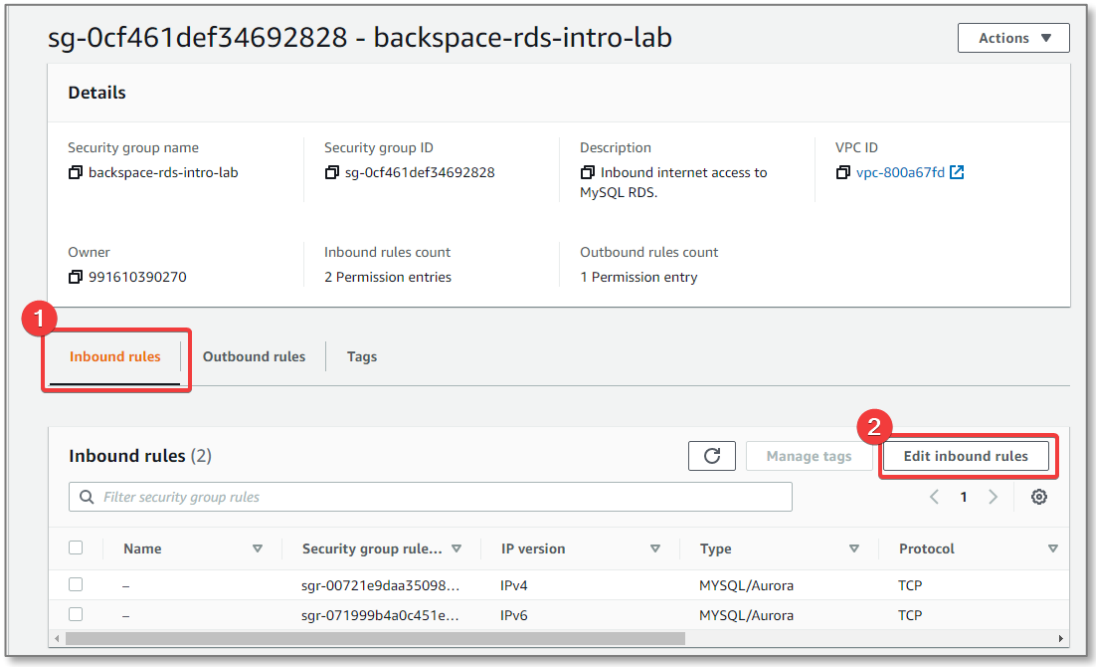
Security Group Inbound Rules

If you did not select *yes* for *publicly accessible* as detailed, your security group will block remote access.
The security group may have an inbound rule for your IP address. If you are using a dynamic IP address or you are connecting from different networks then this will need to be changed to "anywhere" for the lab.

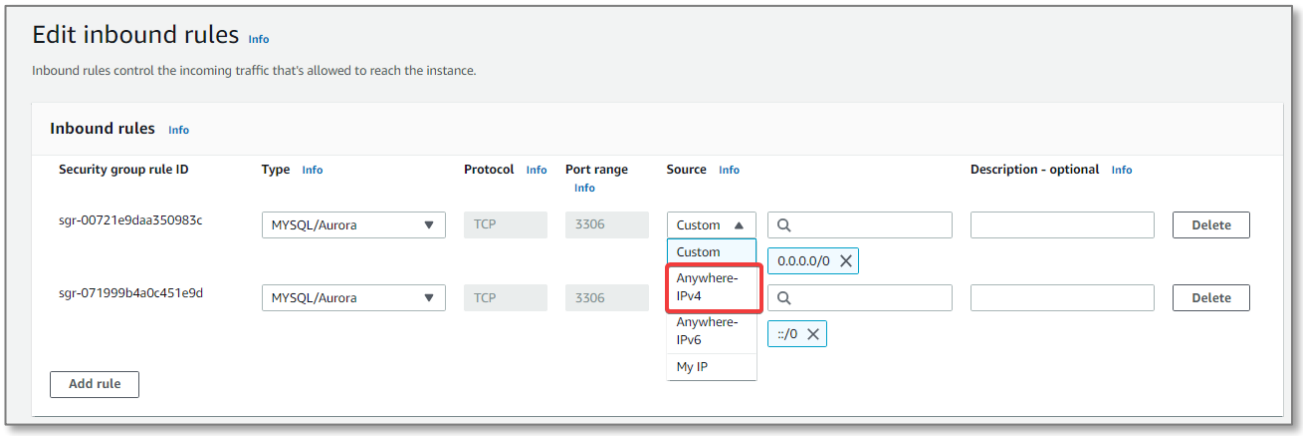
Click the *security group*



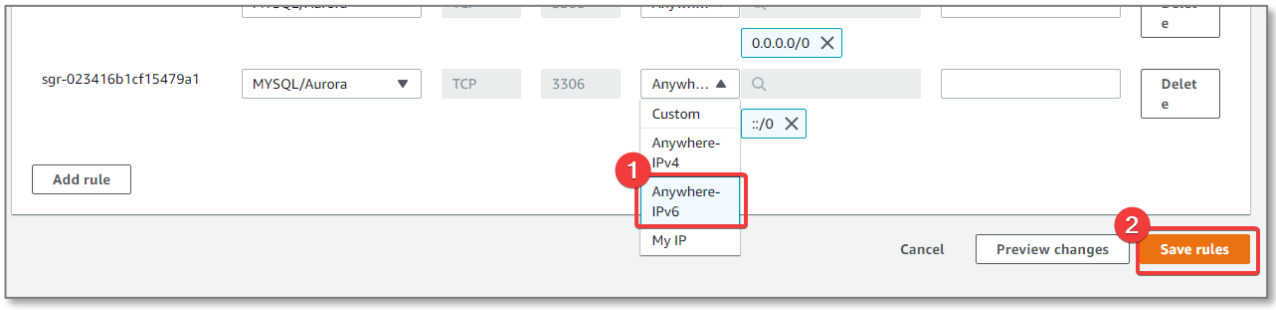
You will be taken to the EC2 console
Select the *Inbound rules* tab
Click *Edit inbound rules*



Change inbound rule to *Anywhere IPv4* 0.0.0.0/0



Change inbound rule to *Anywhere IPv6* ::/0 , click *Save rules*



Client-side Firewall

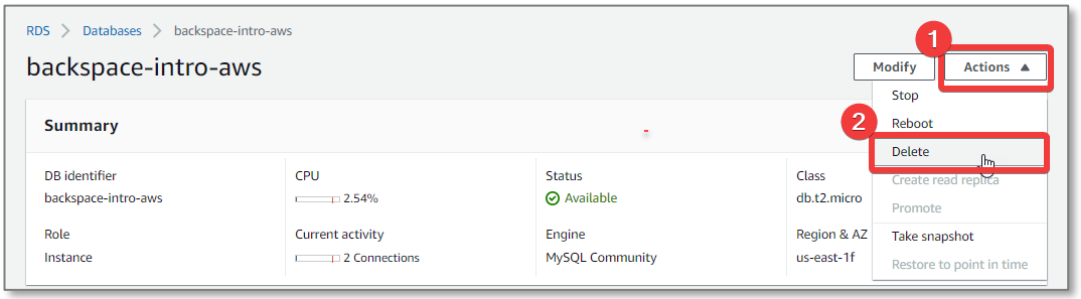
If you are still having problems connecting, a firewall at your end may be preventing access on port *3306.* This is common if you are connecting from your work environment as port *3306* traffic may be blocked.

Clean Up

To avoid incurring charges from AWS we will terminate the instance.

Go back to the *RDS console.*

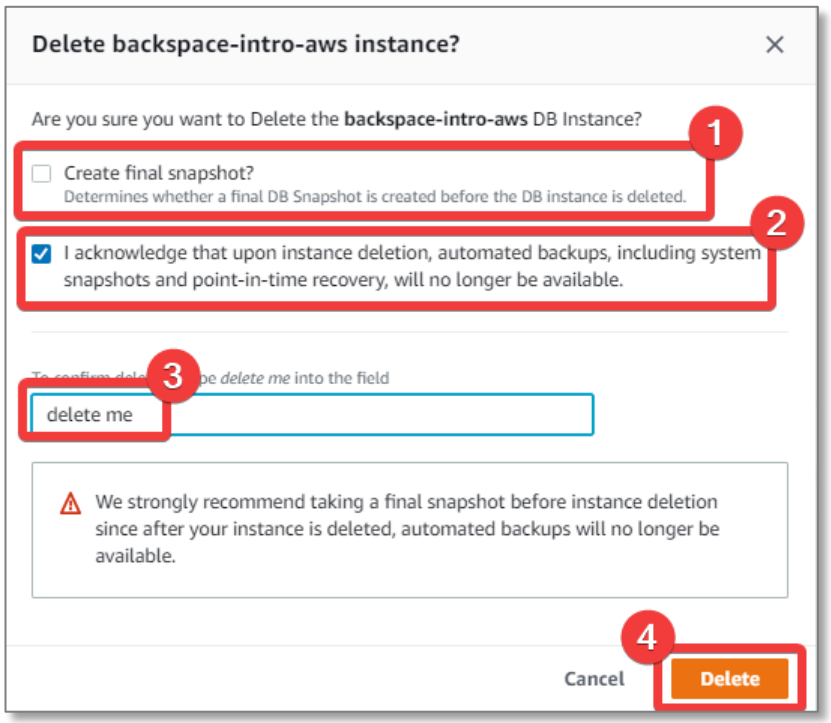
Click *Actions -> Delete* to terminate the instance



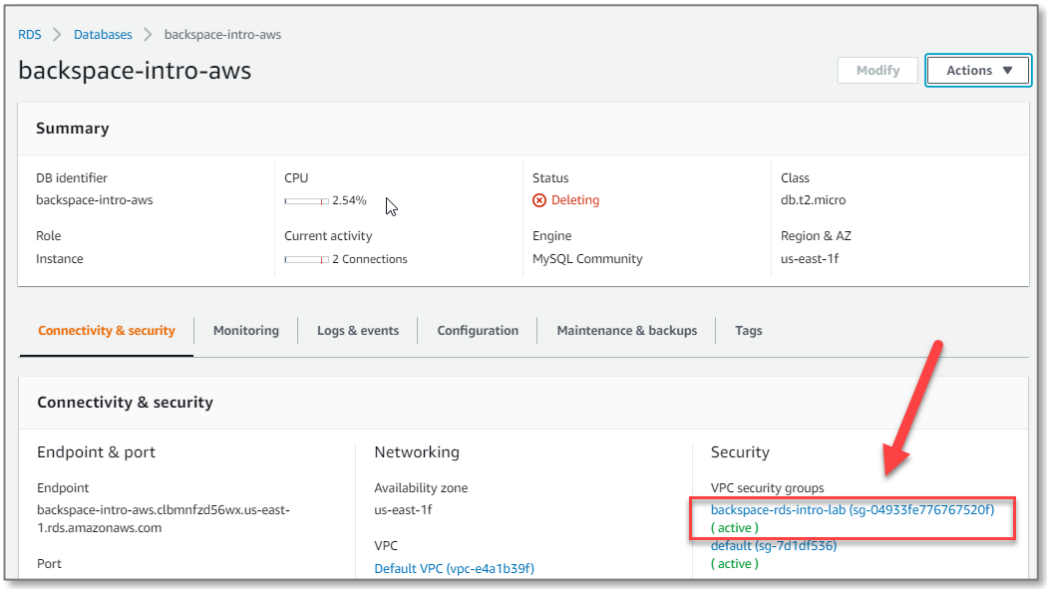
Select *No* for *Create final snapshot*

Check *"I acknowledge that upon instance deletion, automated backups, including system snapshots and point-in-time recovery, will no longer be available."*

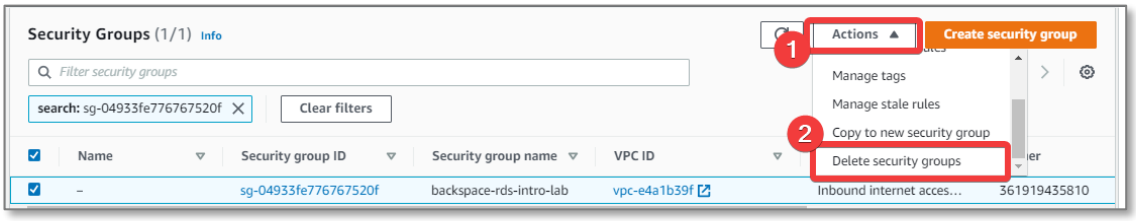
Click *Delete*



Click on the *VPC security group* we created previously



Select *Actions -> Delete security group*



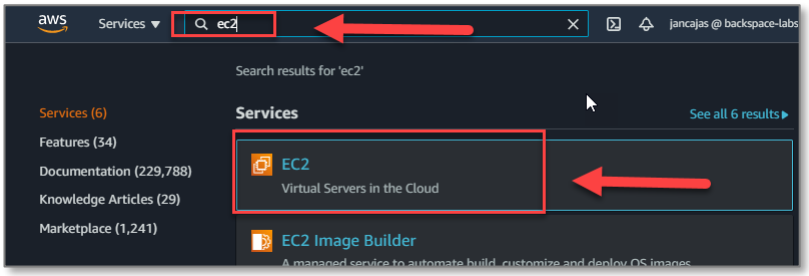
Creating a Web Server with EC2

In this section, we will launch a publicly accessible WordPress application on Amazon EC2.

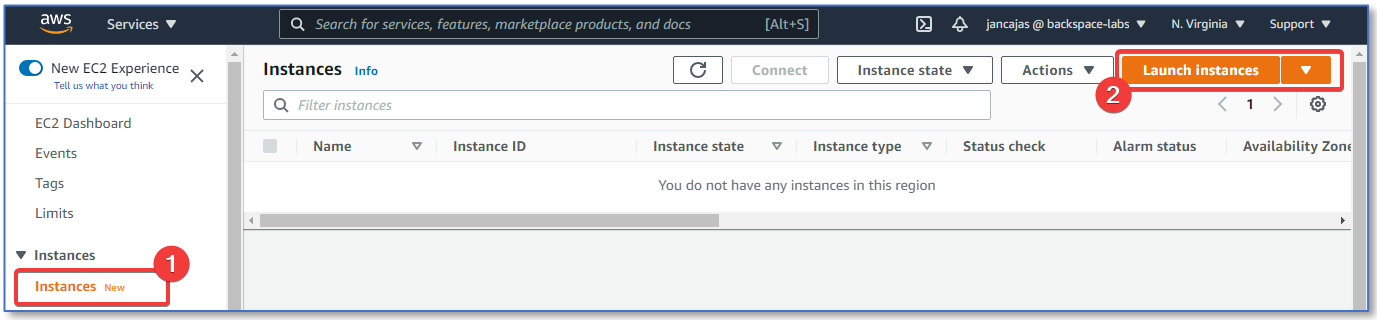
Launch an EC2 Instance

From the AWS console search *EC2*

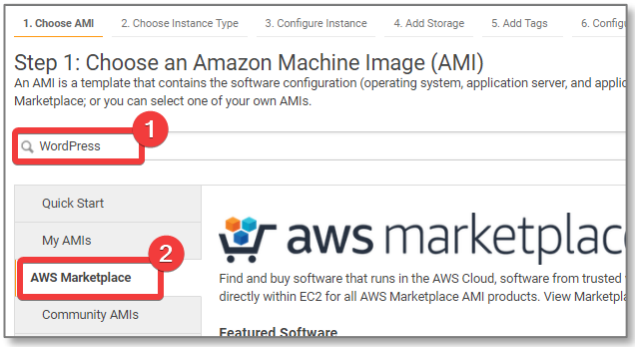
Click *EC2*



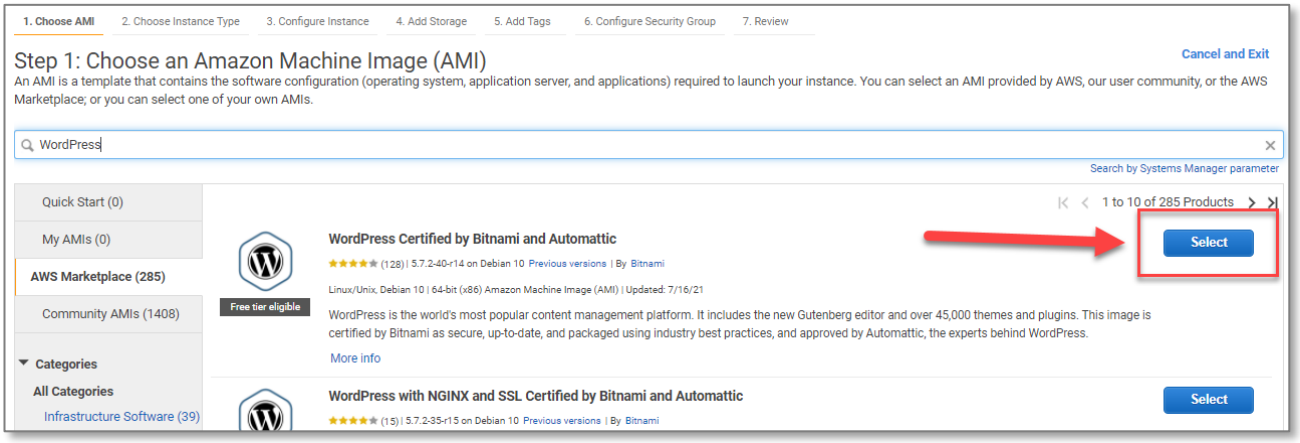
Select *Instances -> Launch Instances*



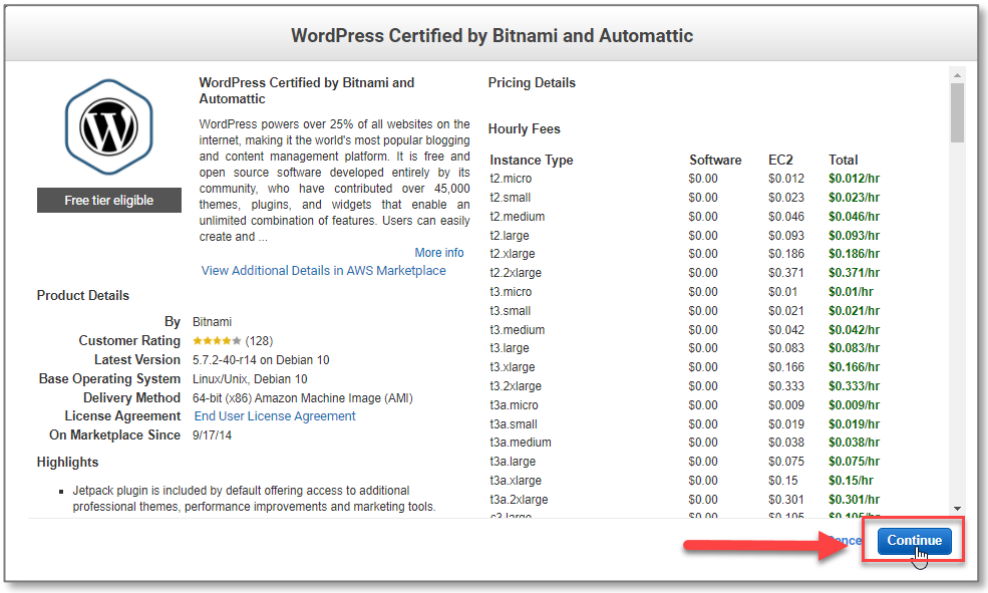
Select the *AWS Marketplace* and search for *WordPress*



Select the *Bitnami AMI*

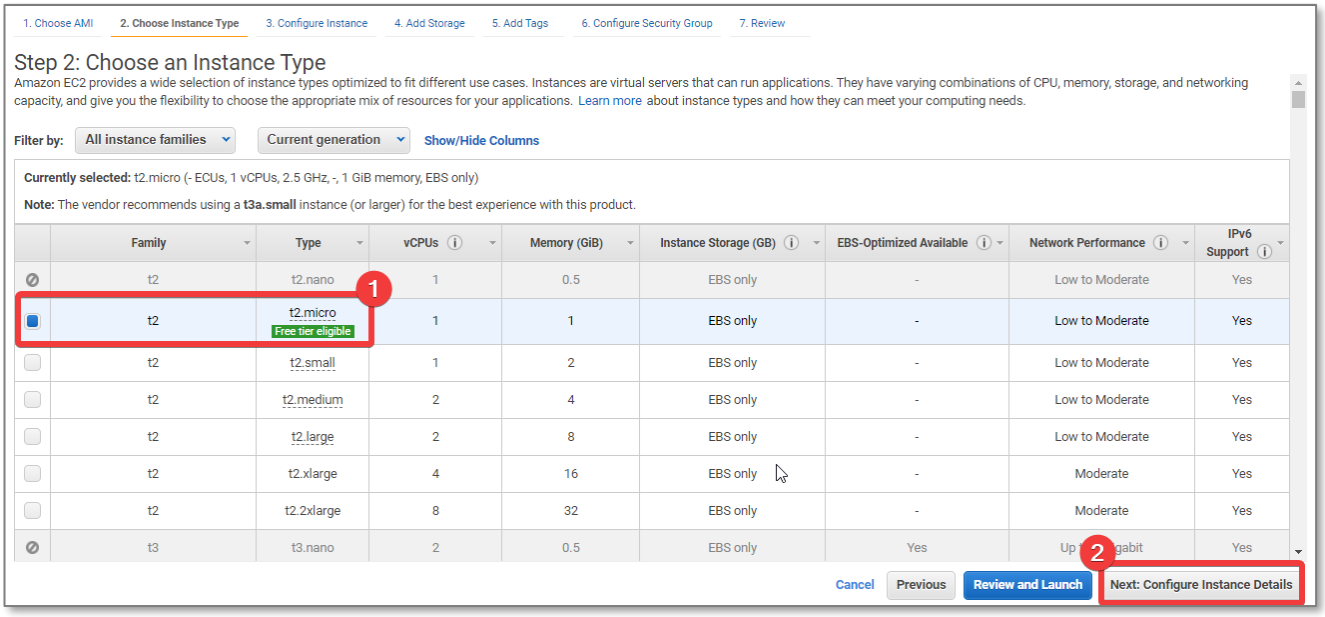


Click *Continue* (This lab will be covered under the AWS free tier for accounts less than 12 months old)



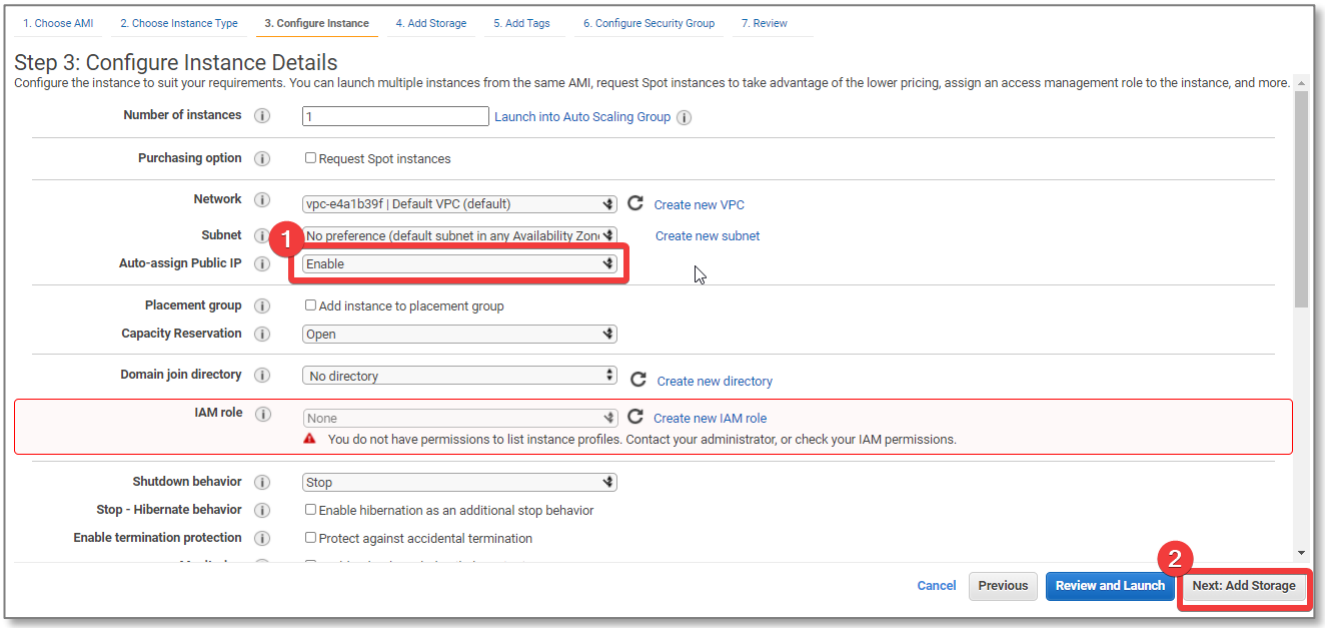
Choose the *t2 micro instance.*

Click Next: *Configure Instance Details*

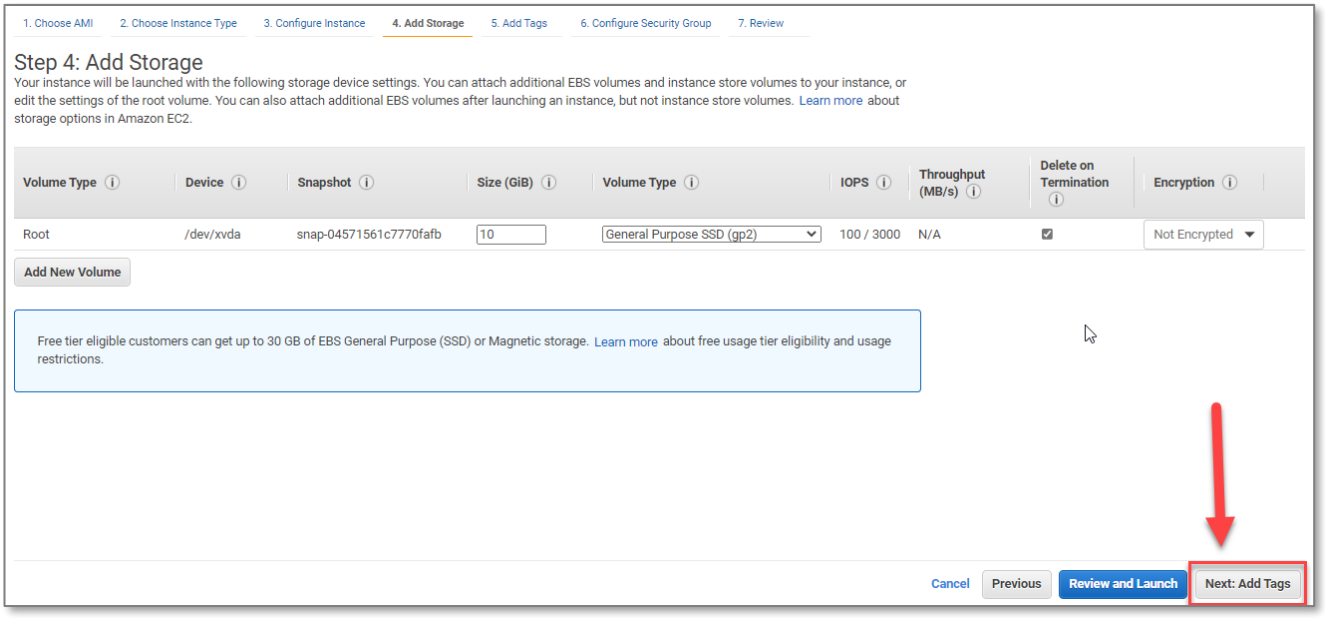


Select enable for *Auto-assign Public IP*

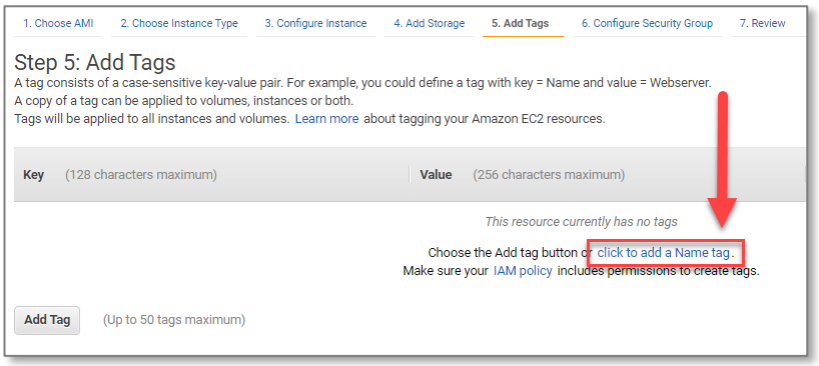
Click Next: *Add Storage*



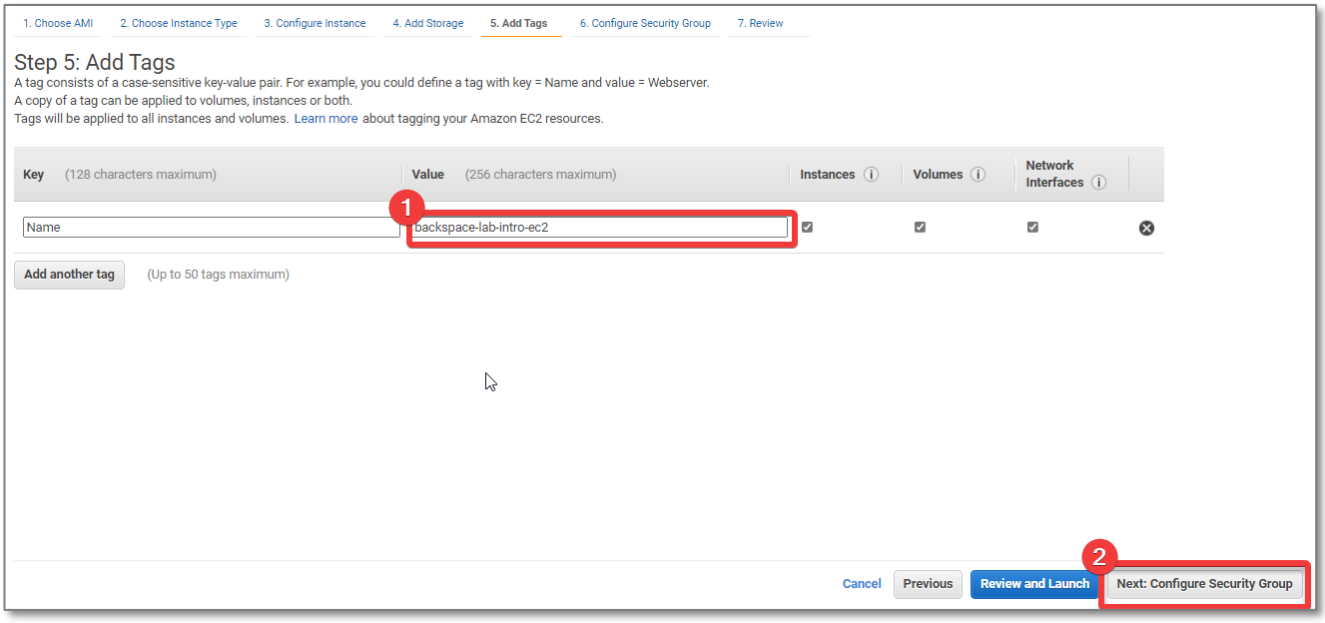
Click Next: *Add Tags*



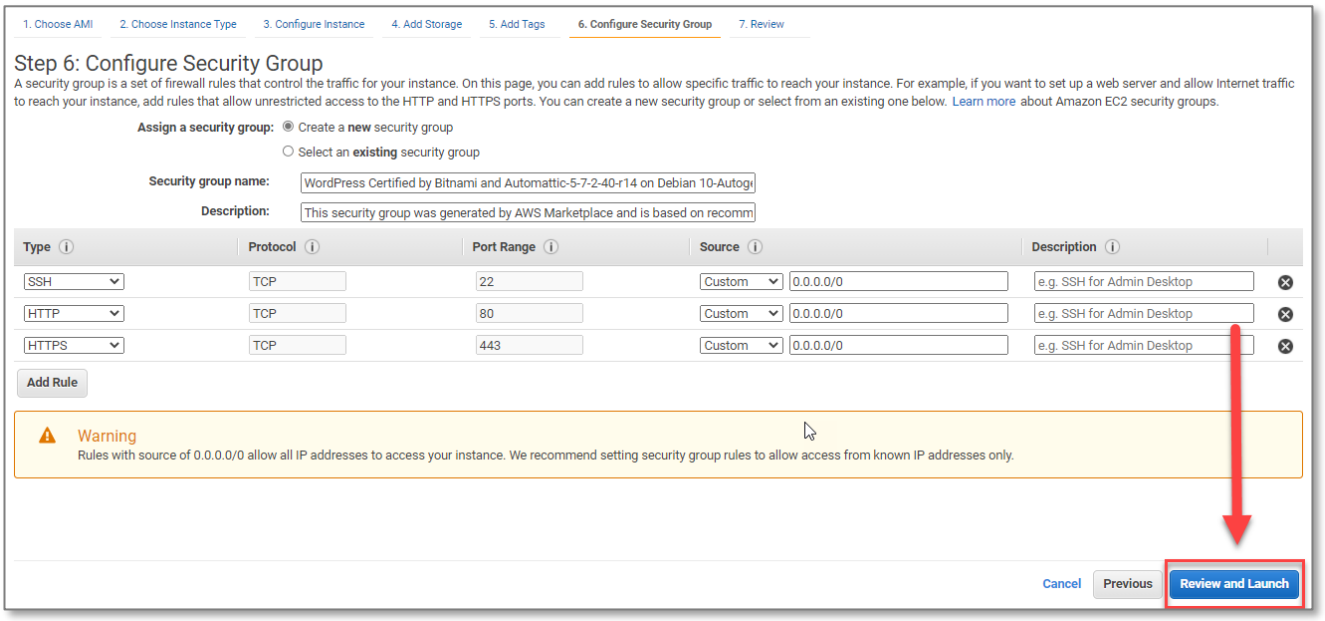
Click to *add* a Name tag



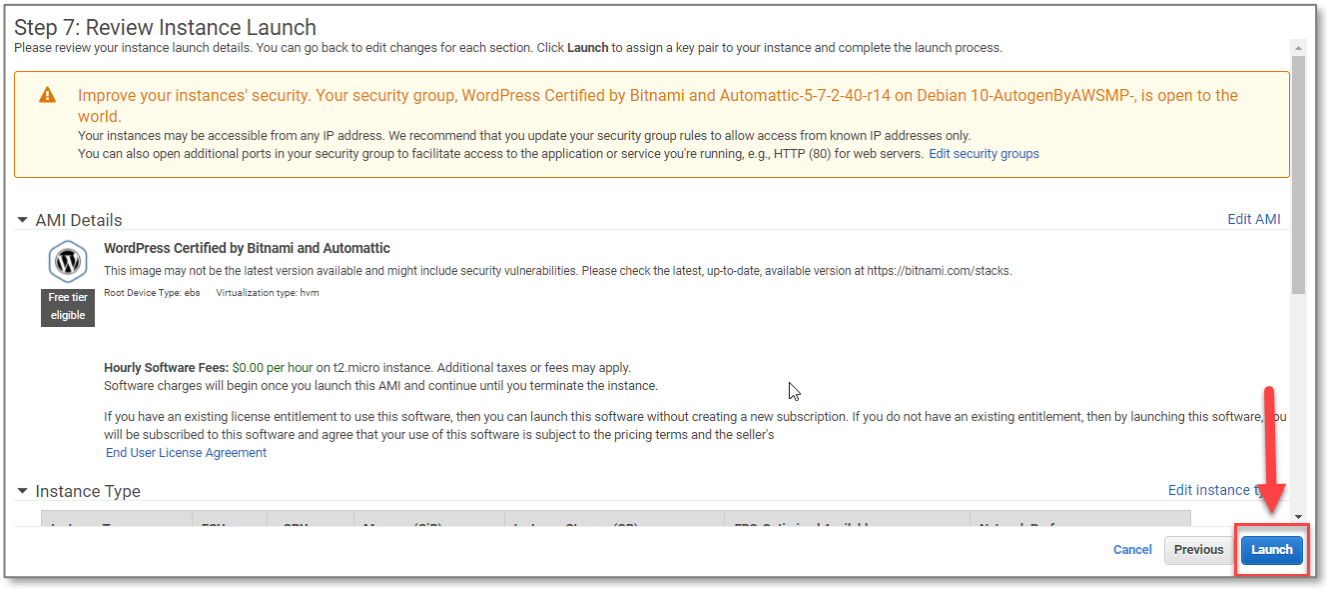
Give it a name and click Next: *Configure Security Group*



Click *Review and Launch*



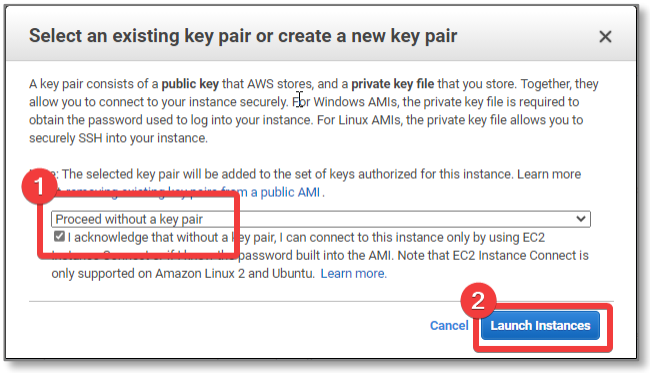
Click *Launch*



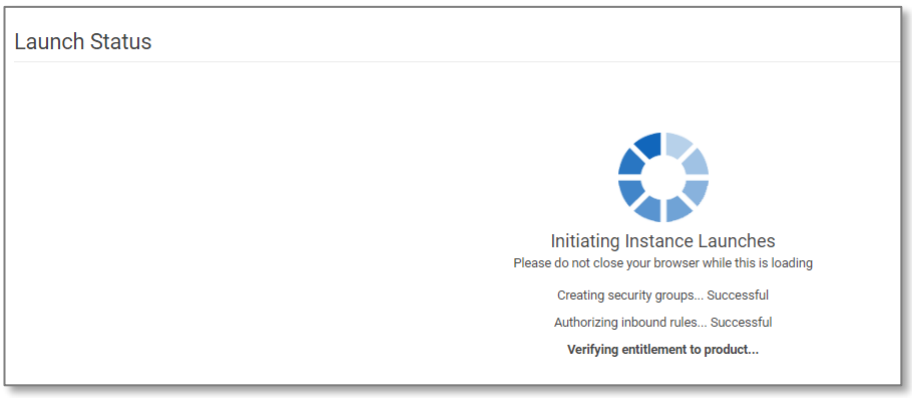
Select *Proceed without a key pair*

Select *"I acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI."*

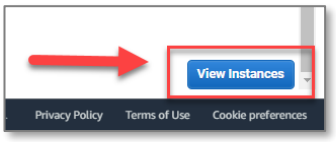
Click *Launch Instances*



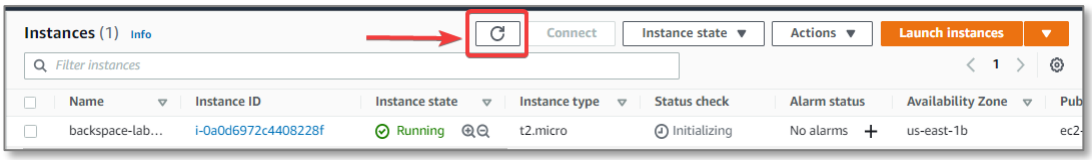
Wait for launch to initiate



When the launch process has started scroll to the bottom of the page and click *View Instances*

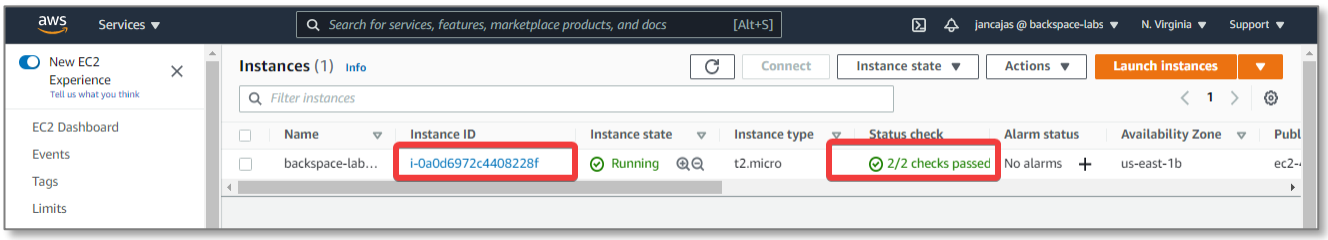


After a few minutes, the status of the instance will change to running and status checks will be completed (you will need to *refresh* the screen to see any changes).

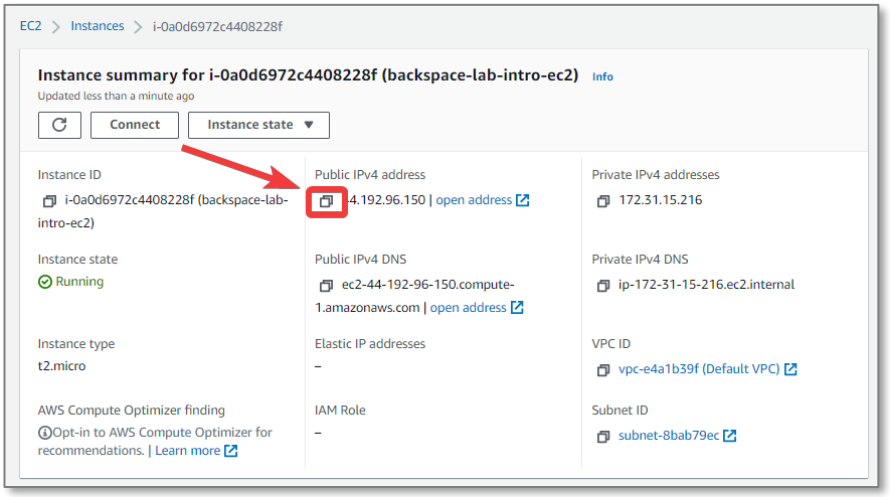


Viewing your web server

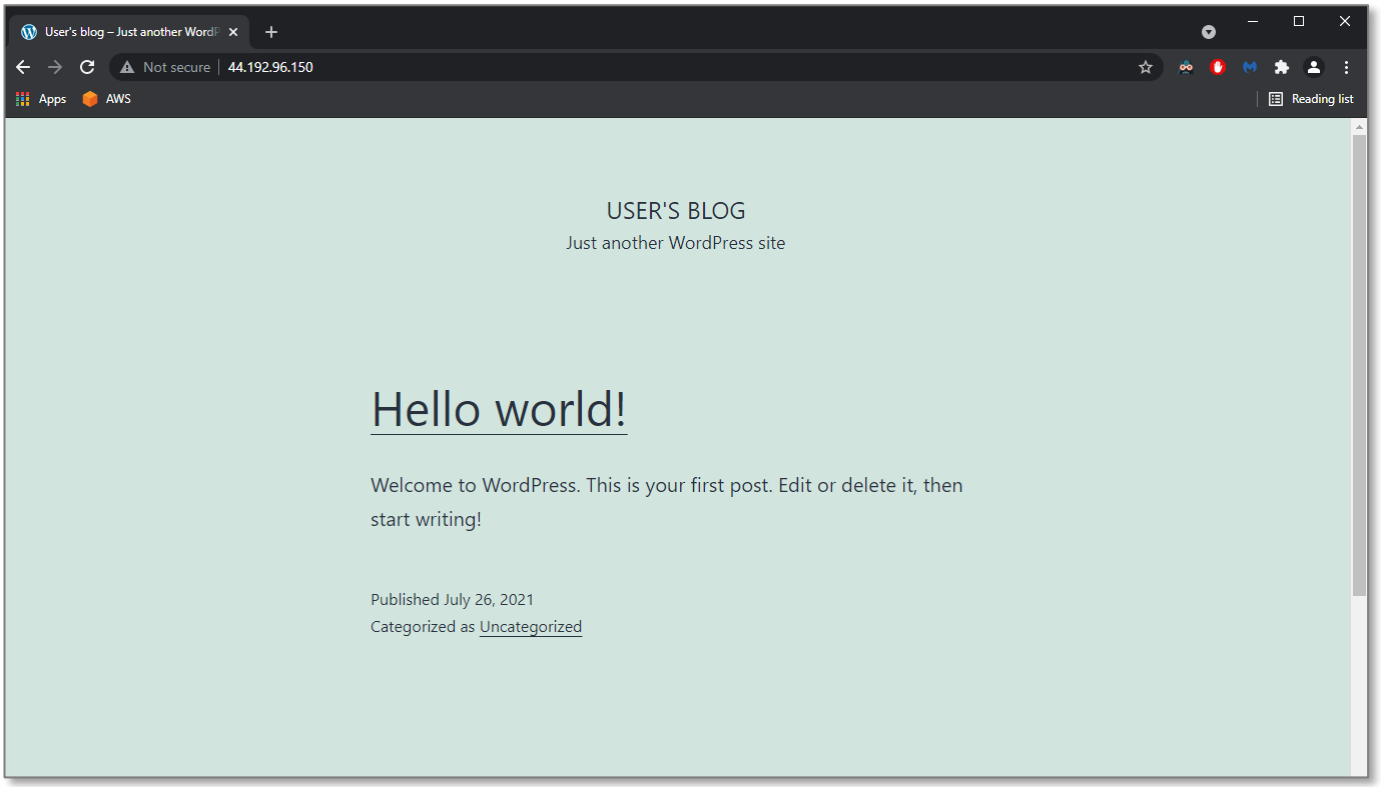
After the Status checks have completed click on the *Instance ID* to select the instance.



Copy the public *IP address* of your web server (don't click on *open address* this will open it in HTTPS not HTTP).



Paste the I*P address* in your browser.



Troubleshooting viewing your WordPress application

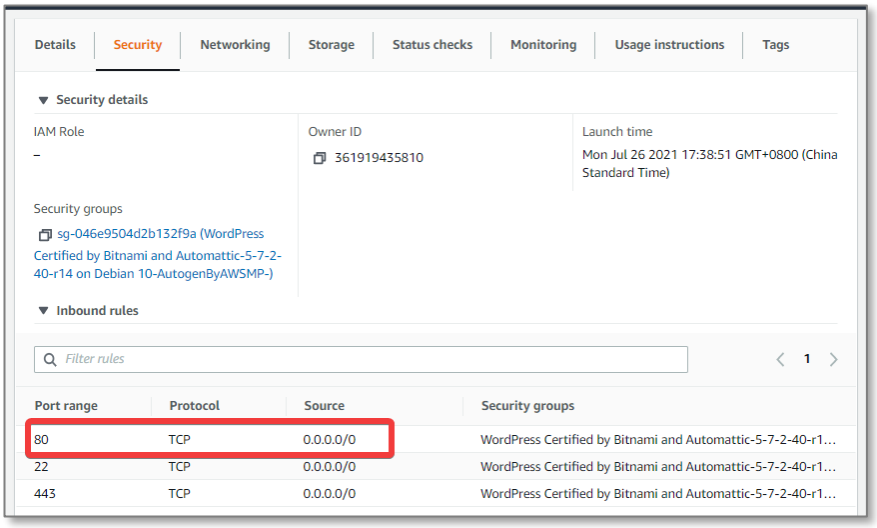
If you cannot view your website, it probably hasn't finished the launch process completely.

If you navigate to your website and it displays a security message, you have tried to open with HTTPS not HTTP.

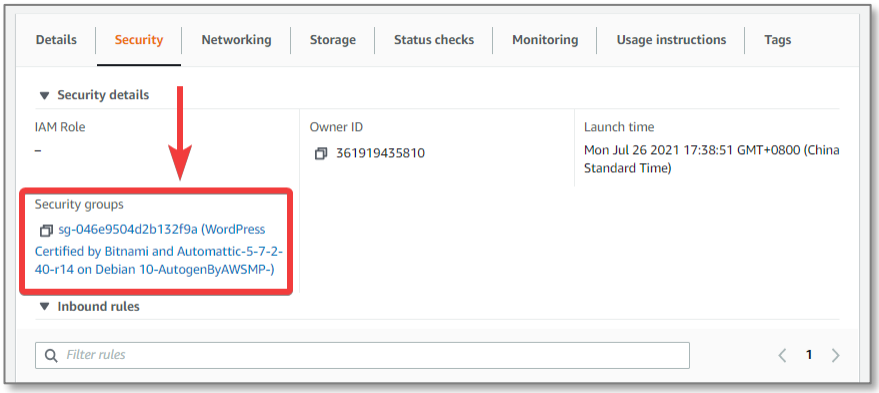
If after quite some time you still can't view your website, it may be that your security group does not allow inbound requests on port 80 (http). The inbound rules should include:

80 tcp 0.0.0.0/0

Scroll down and click on the *Security tab*

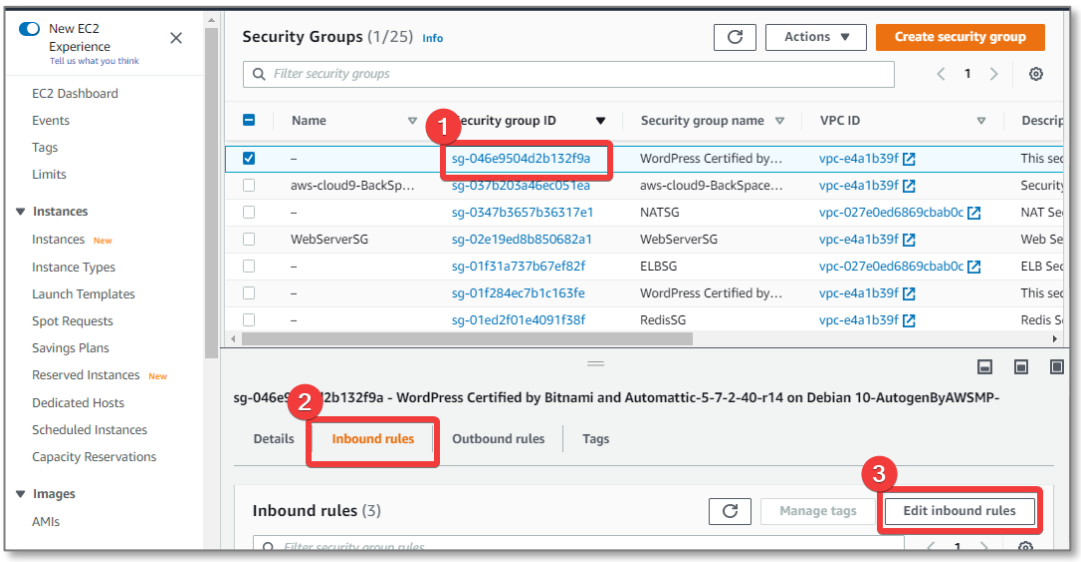


If the rule is not present you will need to *add* it by clicking on the *security group* to open it:



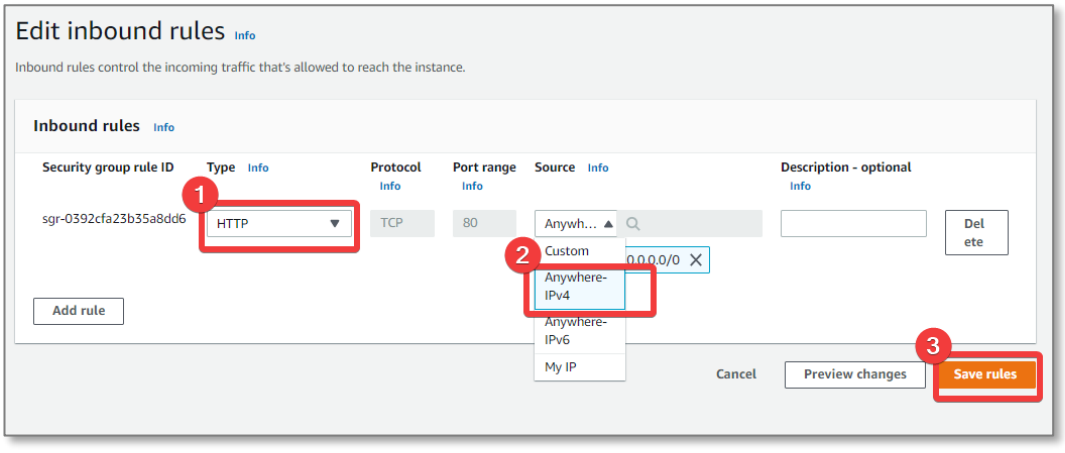
Click on the *Inbound rules tab*

Click on *Edit inbound rules*



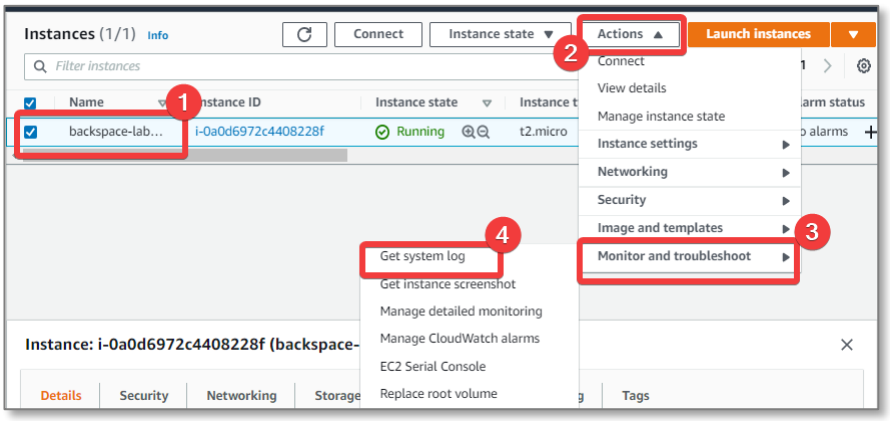
Add a rule for *HTTP* and *Anywhere IPv4 0.0.0.0/0*

Click *Save rules*

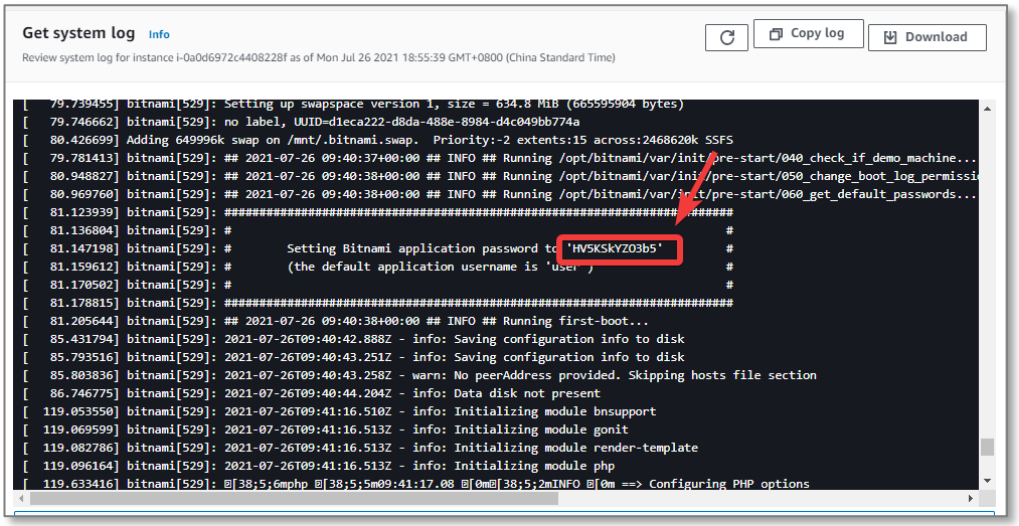


Finding the Username and Password for your WordPress application

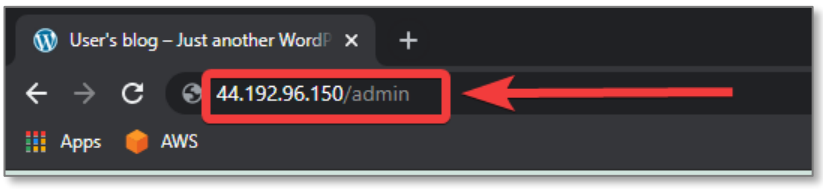
Go back to the EC2 console and select *“Monitor and troubleshoot”*, *“Get System Log”*. Do not click on connect.



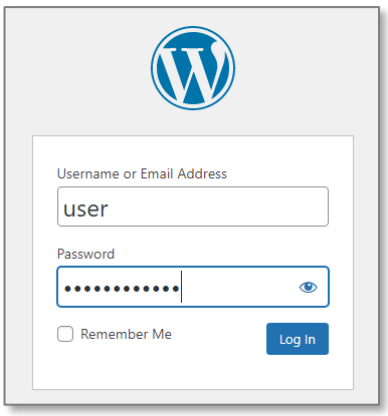
Scroll up until you find the log entry for the application *username and password* and copy it.



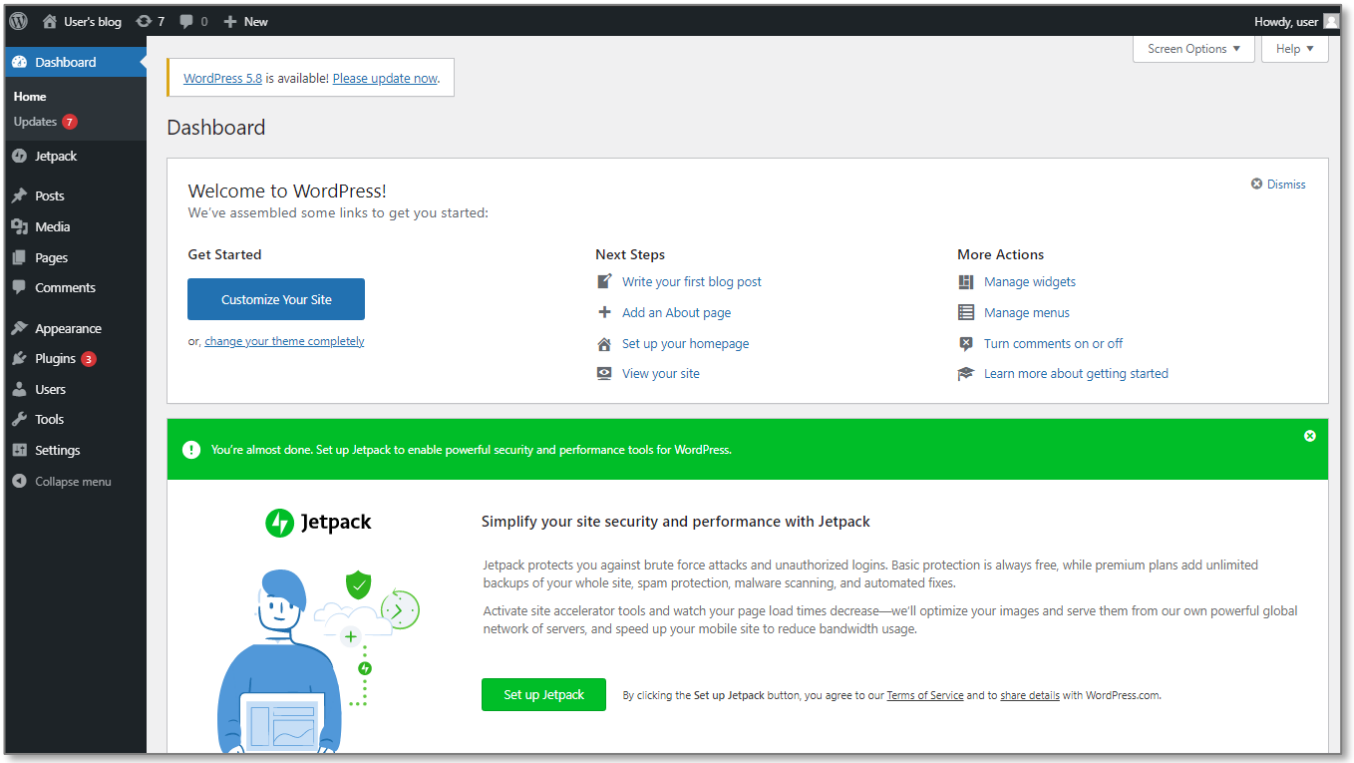
Go to the *admin subdirectory* of your website in your browser



Enter Username *user* and paste in the password

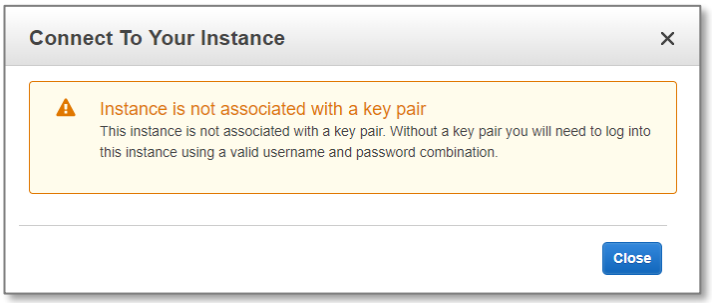


You will now be in the *admin* section of your WordPress application



Troubleshooting logging in to the WordPress application

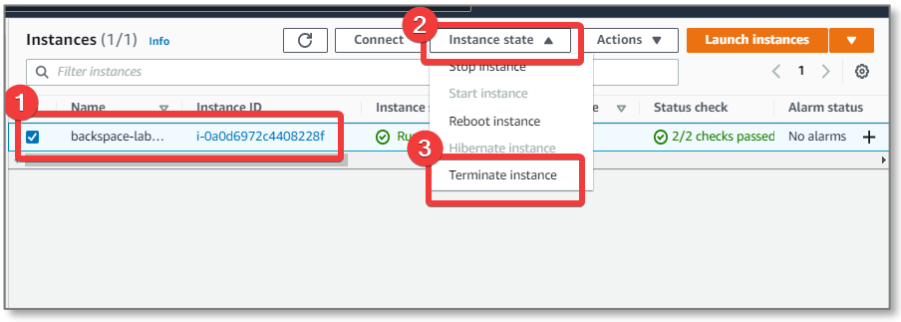
If you get the following message:



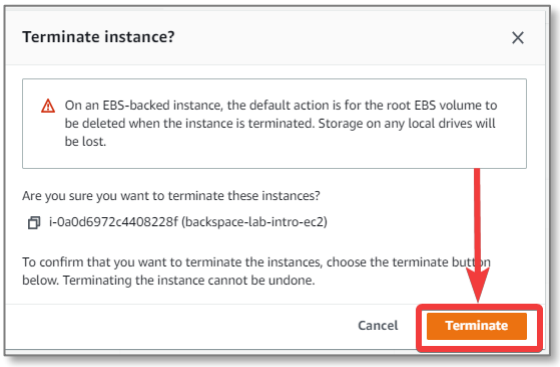
You have tried to connect to the Linux operating system by clicking on *Connect.* Do not click on connect, select *Actions – Instance settings - Get System Log* as detailed previously.

Clean up

Select *Actions -> Instance State -> Terminate*



Make sure you *terminate* the instance so that you are not billed for it anymore.

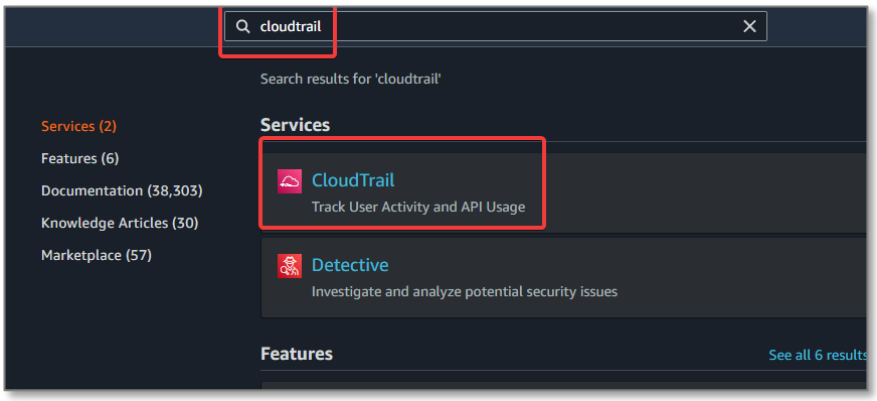


Monitoring User Activity with AWS CloudTrail

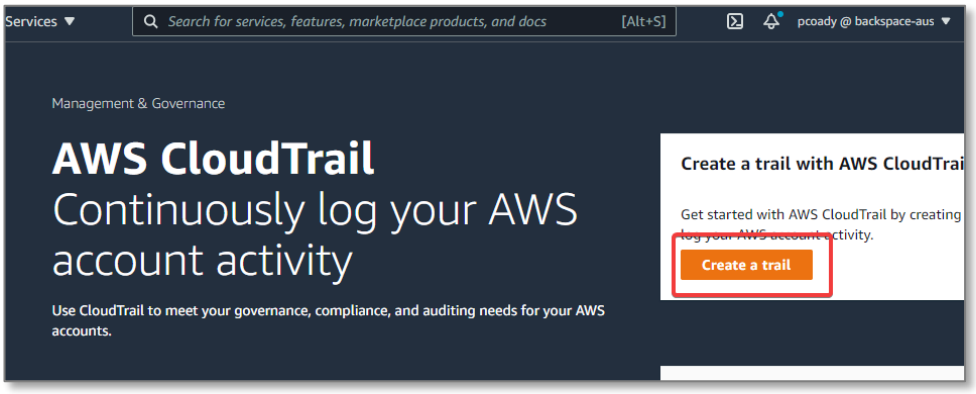
In this section, we will use the AWS CloudTrail service to create a trail and monitor user activity on our account.

From the AWS console search *CloudTrail*

Click *CloudTrail*

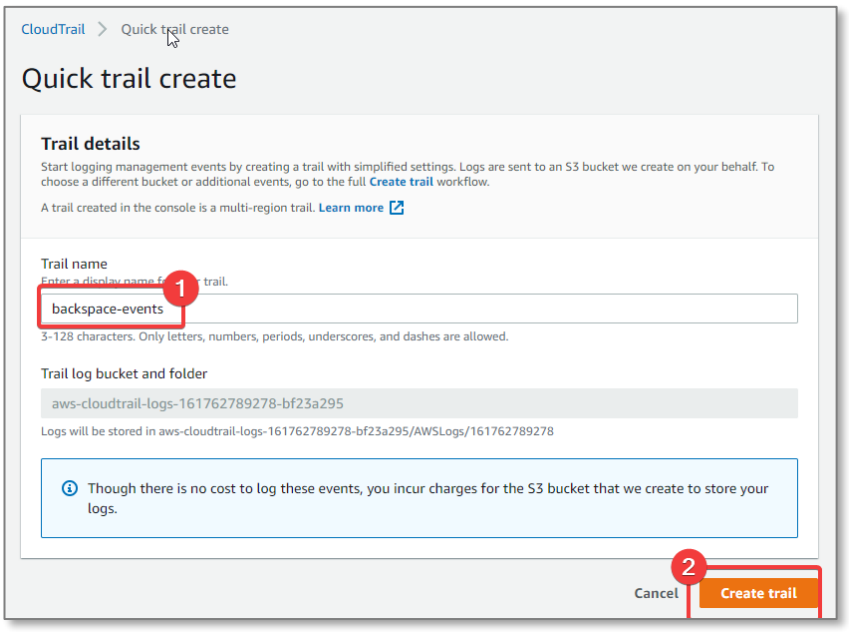


Click on *Create a trail*

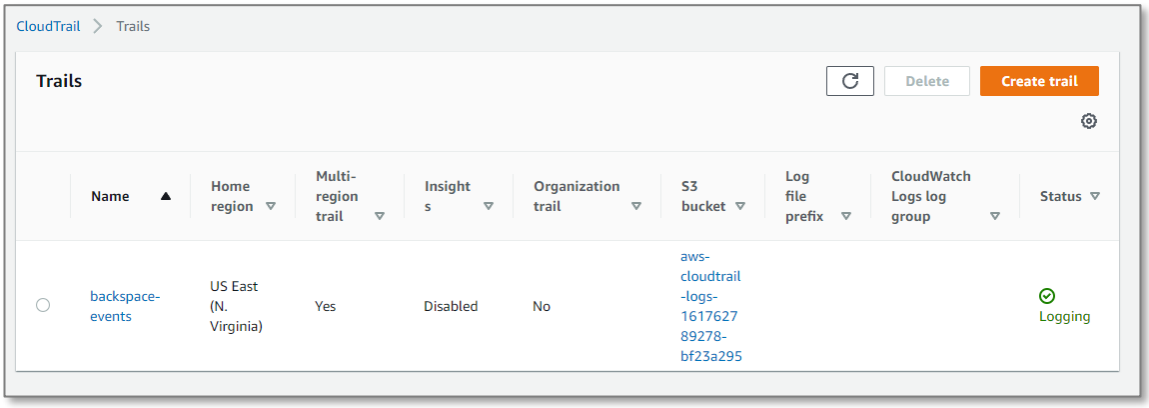


Give the trail a name

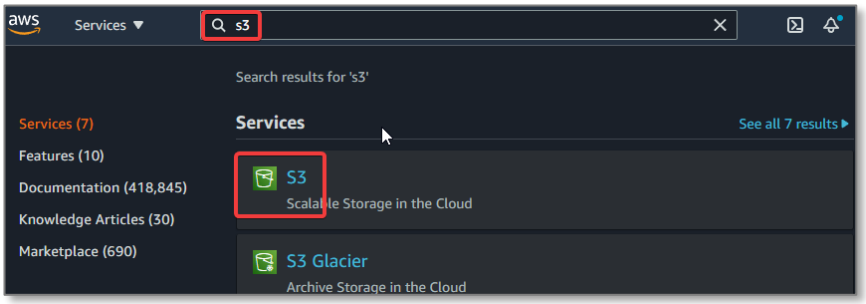
Click on *Create trail*



Your trail is now active

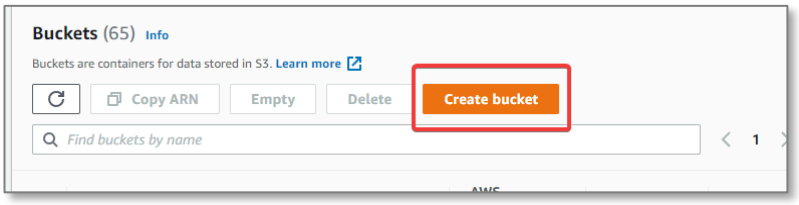


Go to the S3 console

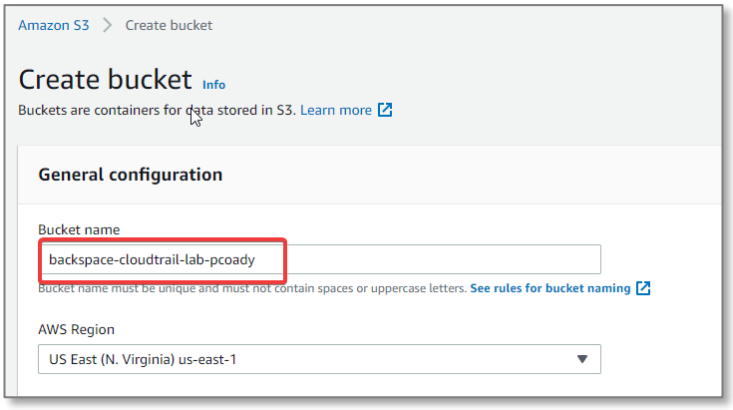


You will now see a bucket for storing the CloudTrail events has been created.

Click *Create bucket*

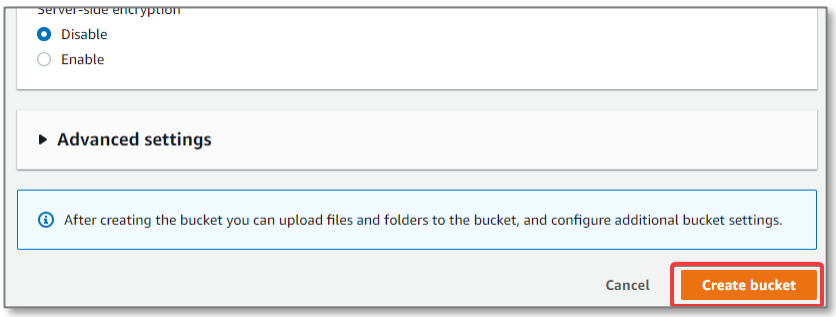


Give the bucket a unique name

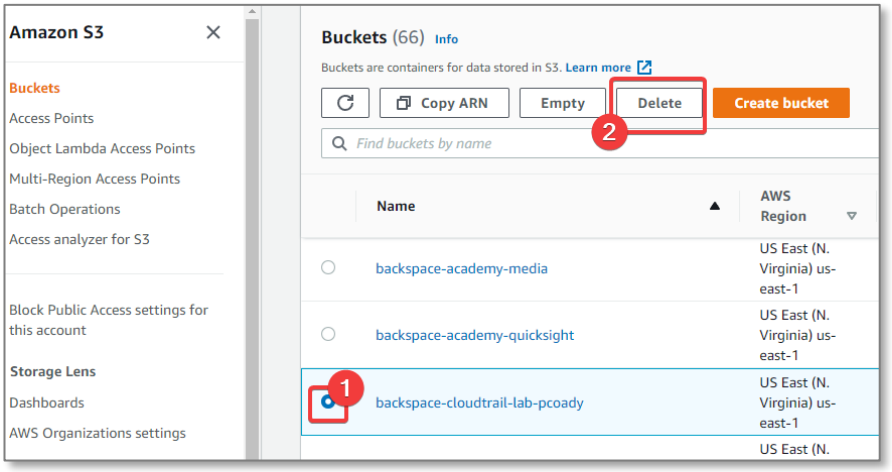


Leave default settings

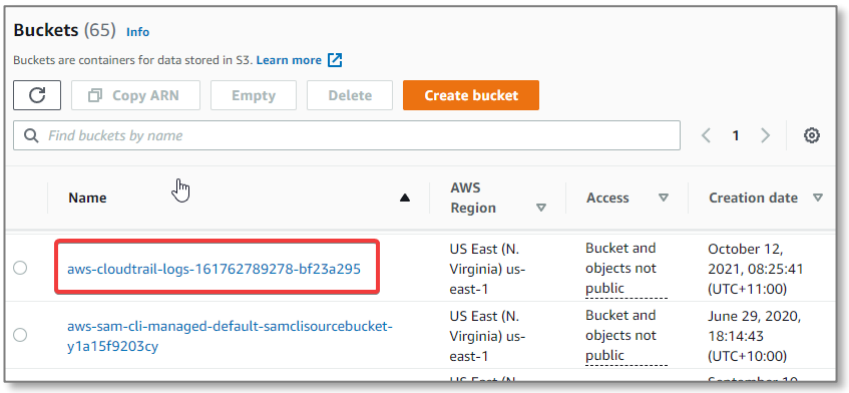
Click *Create bucket*



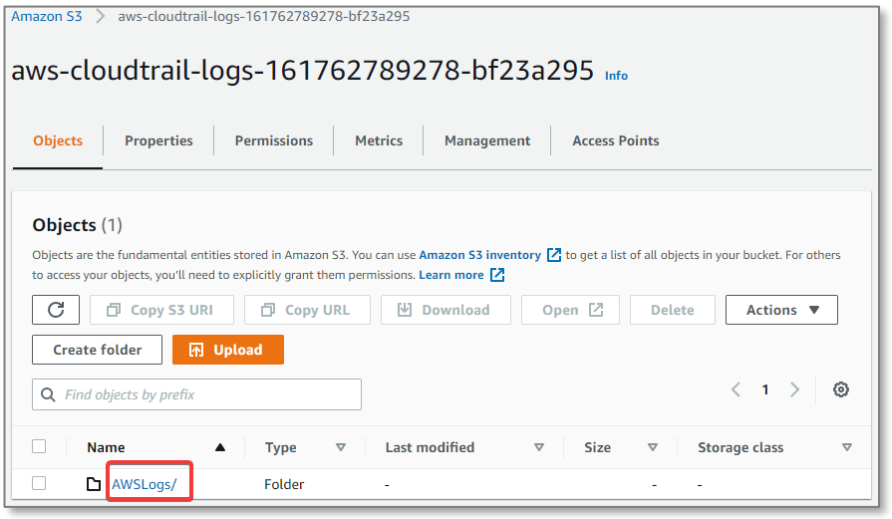
Now delete the bucket



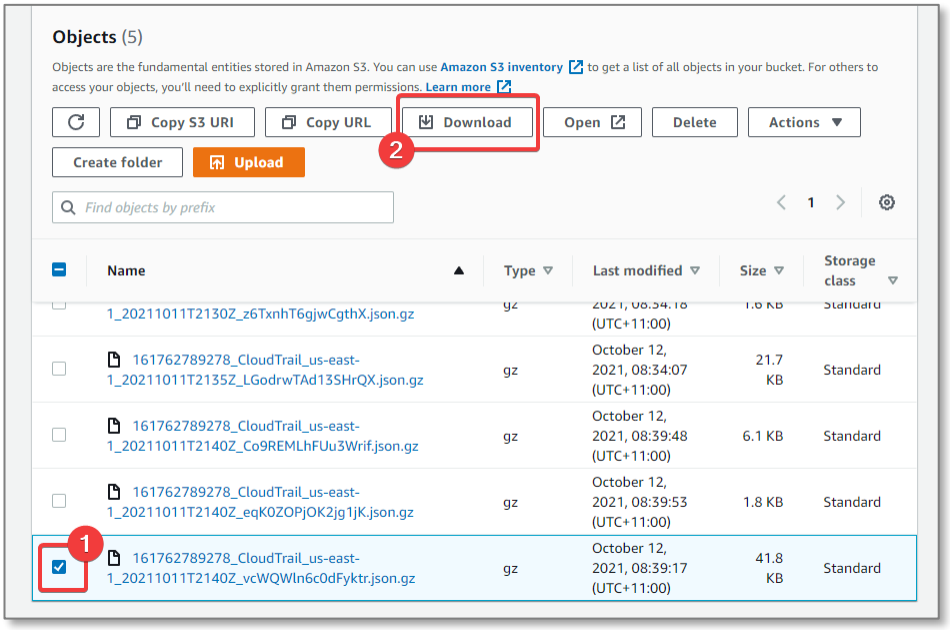
Now open the CloudTrail logs bucket



Navigate to find the logs in your region



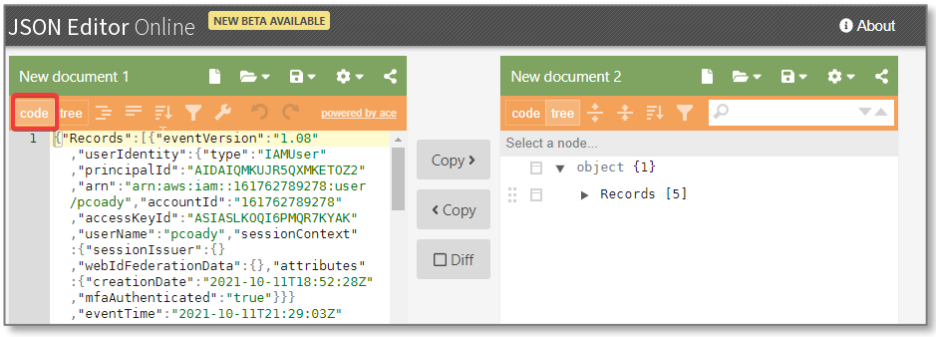
Select the latest log file and click *Download*



Go to https://jsoneditoronline.org

Select *Code* on the left pane

Paste in the contents of the file

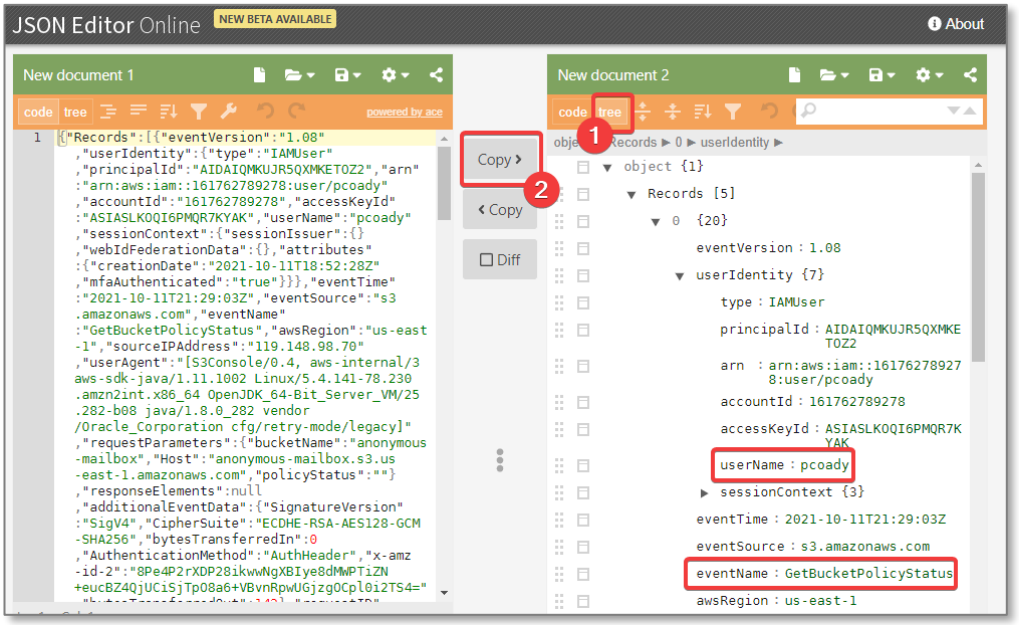


Select *Tree on the right pane*

Click *Copy>*

You will now be able to see the list of event records.

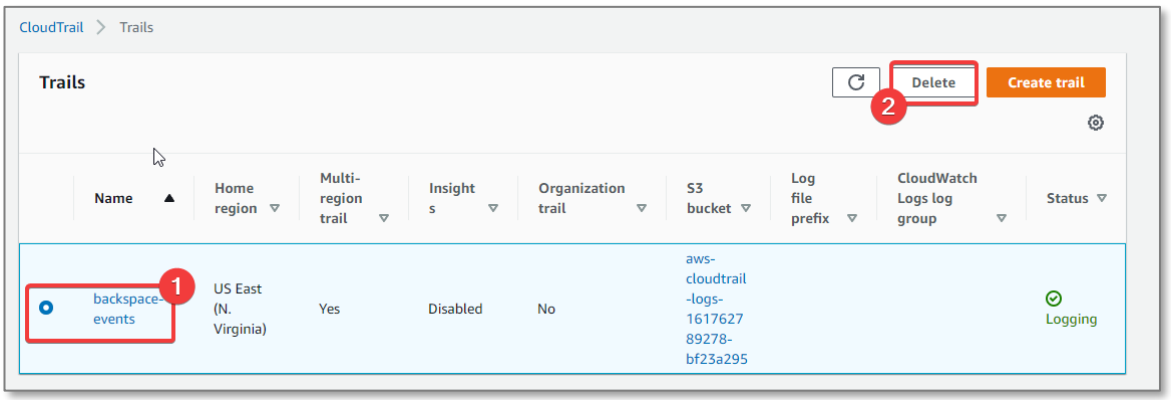
Expand one of the records to see the user details and the call made



Clean Up

Go back to the CloudTrail console

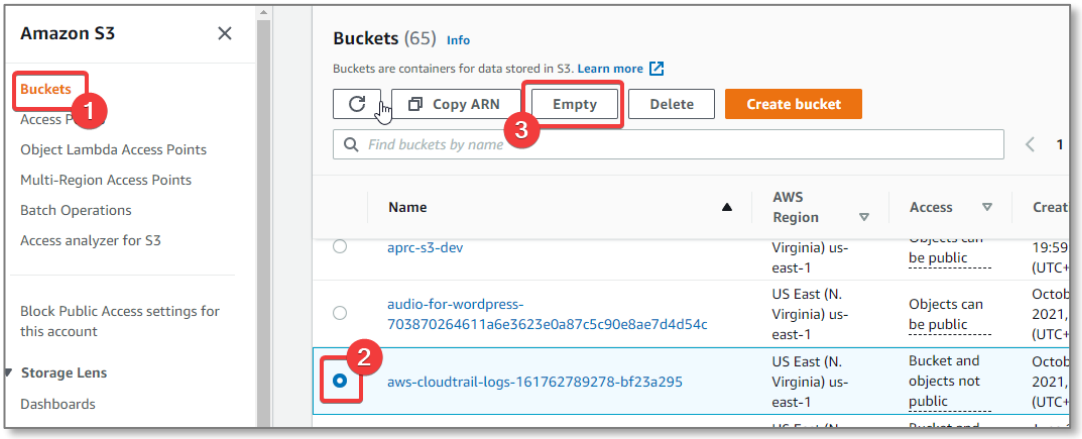
Delete the trail.



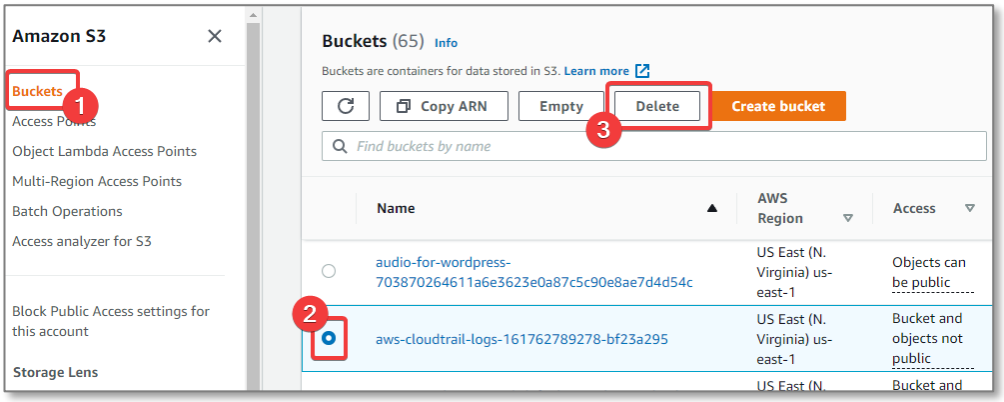
Please note you will not be able to empty and delete the S3 bucket unless the CloudTrail trail has been deleted.

Go back to the S3 console

Empty the CloudTrail bucket



Delete the empty CloudTrail bucket

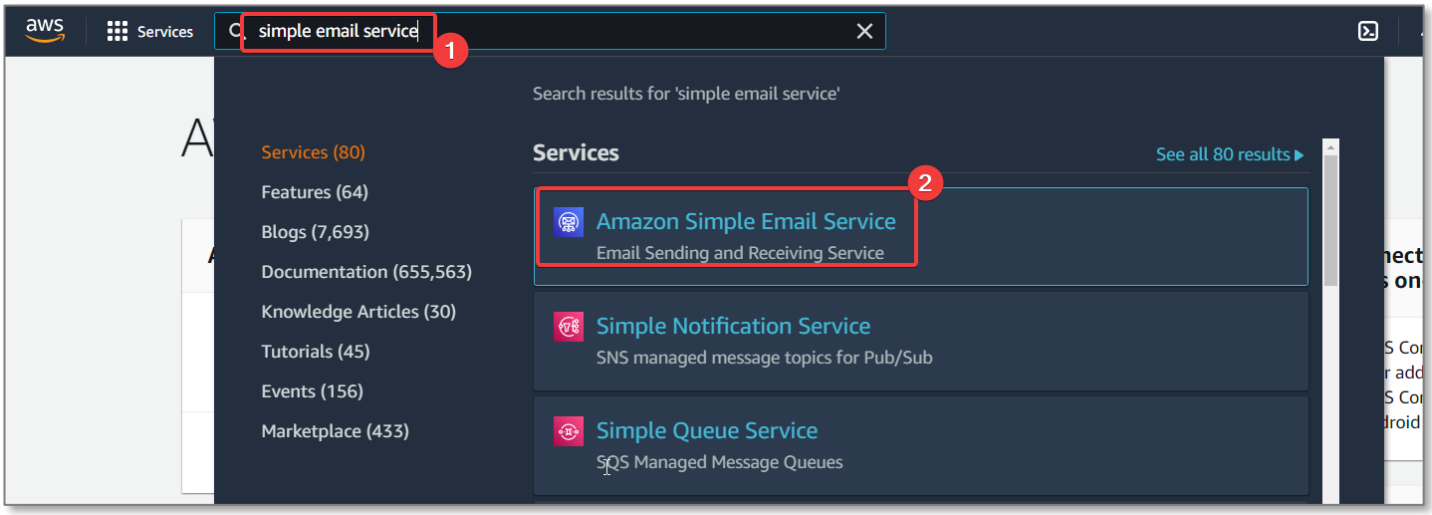


Sending Emails with Amazon SES

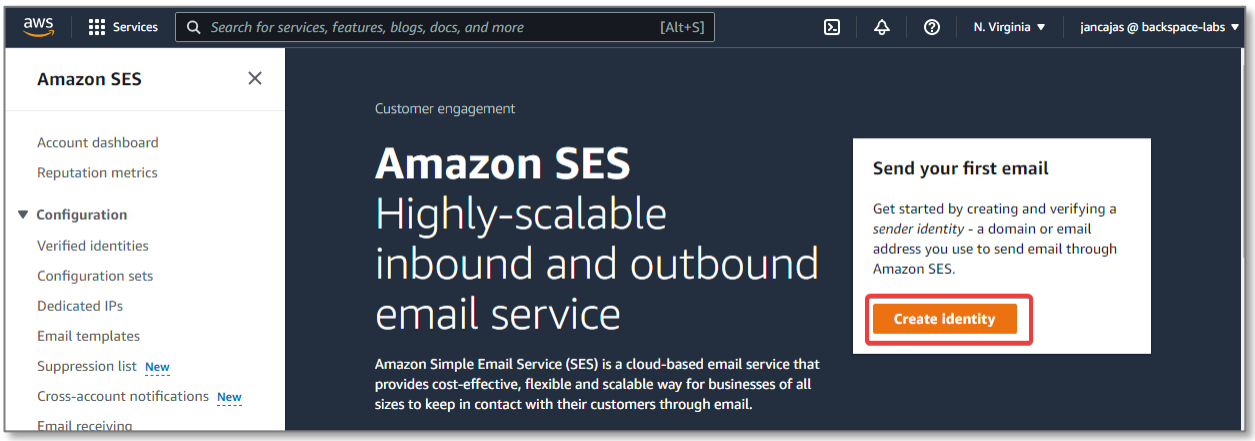
In this section, we will use the Simple Email Service to send an email.

From the AWS console search *SES*

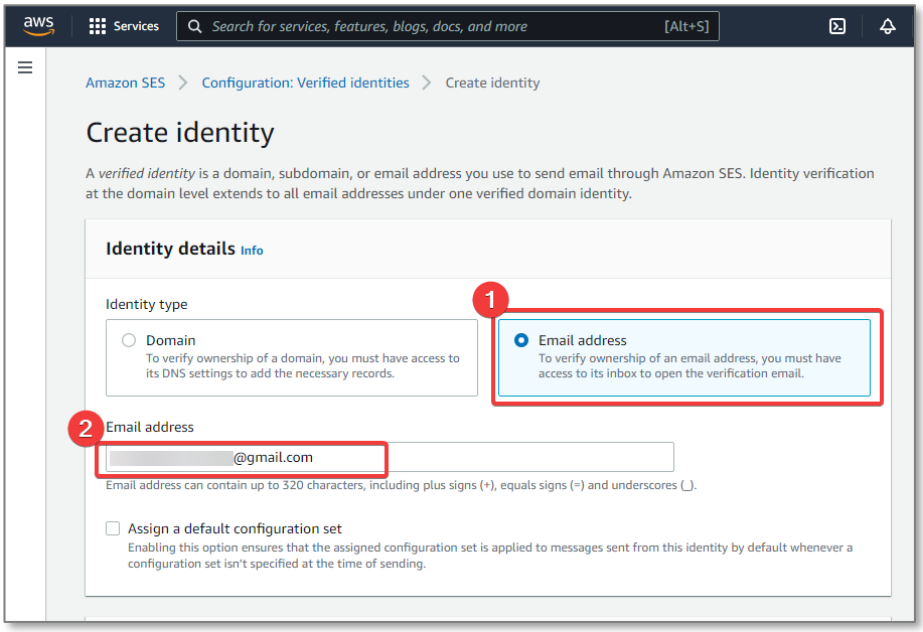
Click *Simple Email Service*



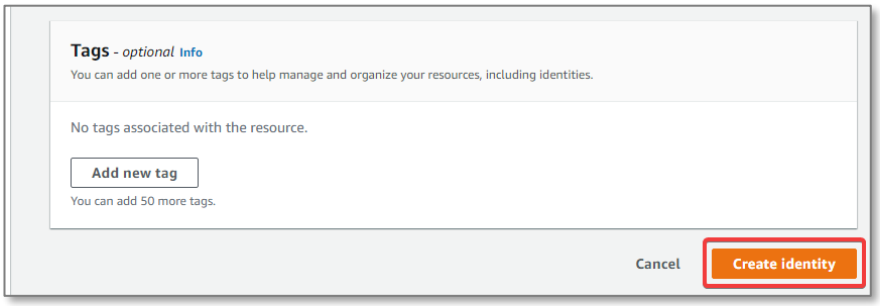
Click on *Create Identity*



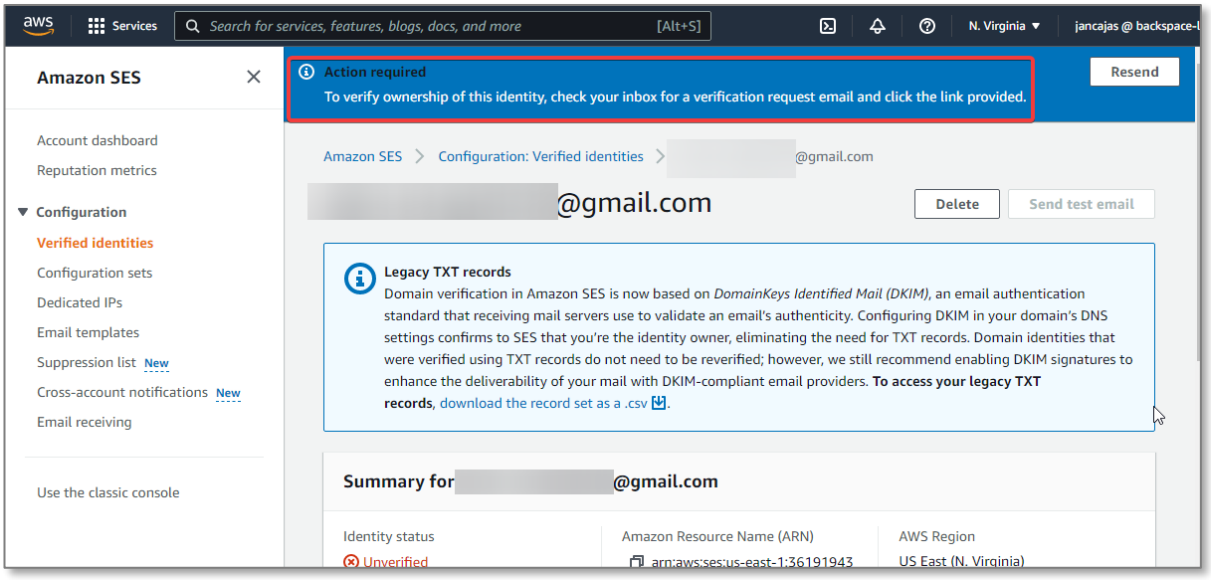
Select *email address* and type-in your *Email Address*



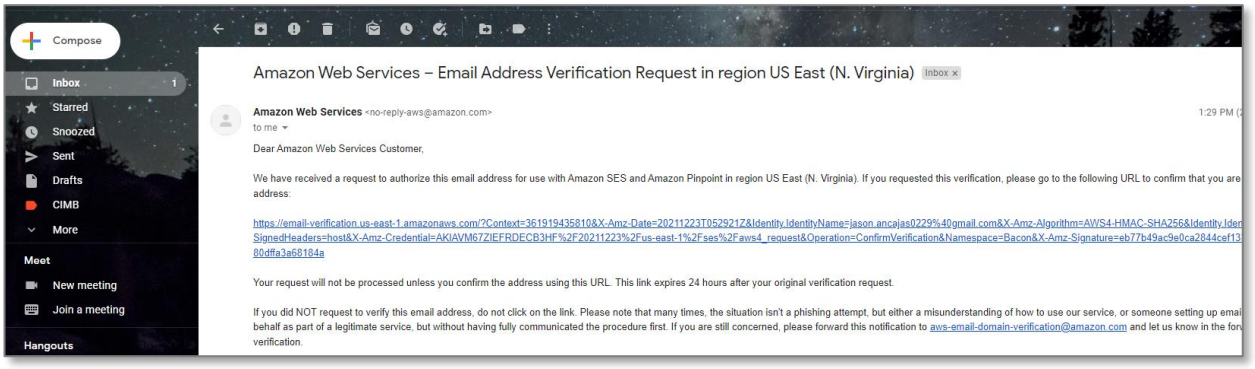
Click *Create Identity*



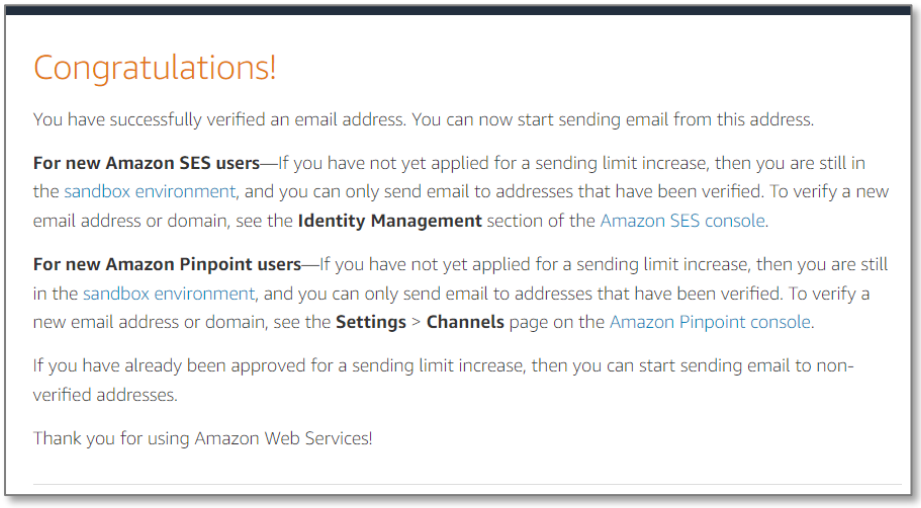
You need to verify it from your entered email address



Check your *email*

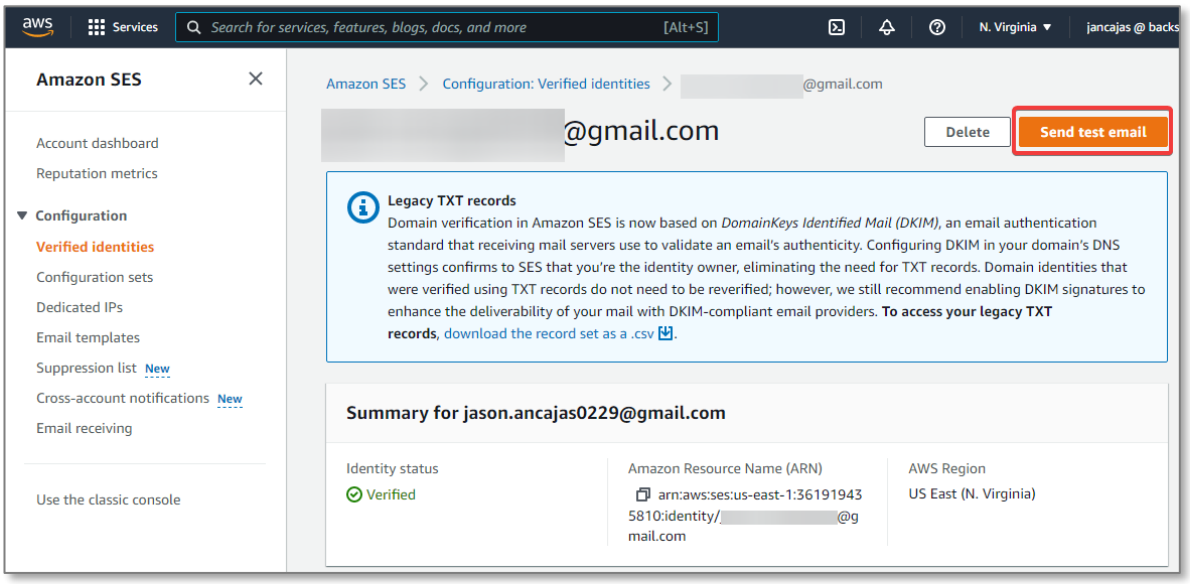


You will now receive a verification status.



Go back to the *SES console page* and *refresh* the information to see the email has been verified

Send a test email



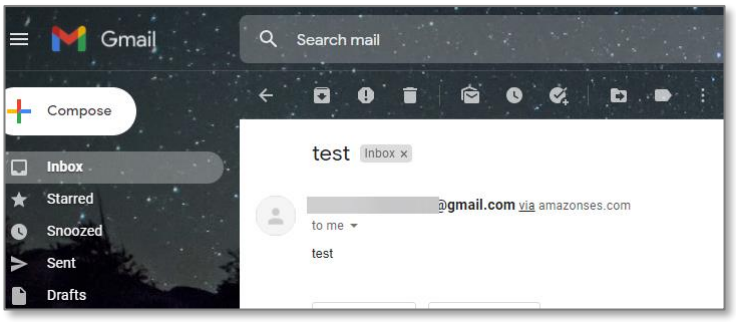
Select *Formatted*

Select *Custom* for the Scenario

Enter the email address *recipient*

Click on *Send test email*

Check your email to see if it worked.

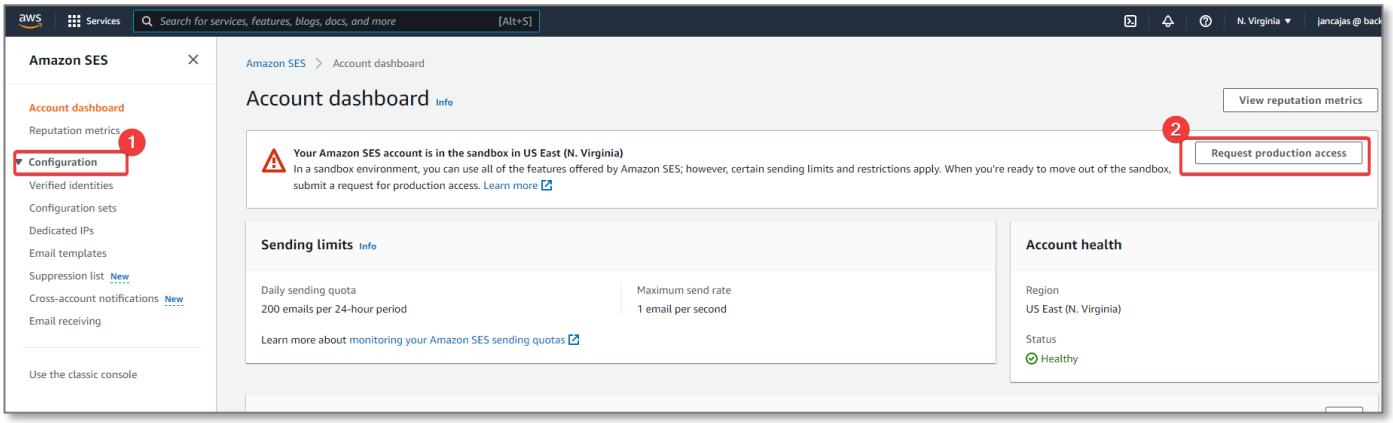


Requesting full access to SES

New accounts only have sandbox access, but this can be changed by applying to AWS.

Click on *Confuguration*

Click on *Request Production access*

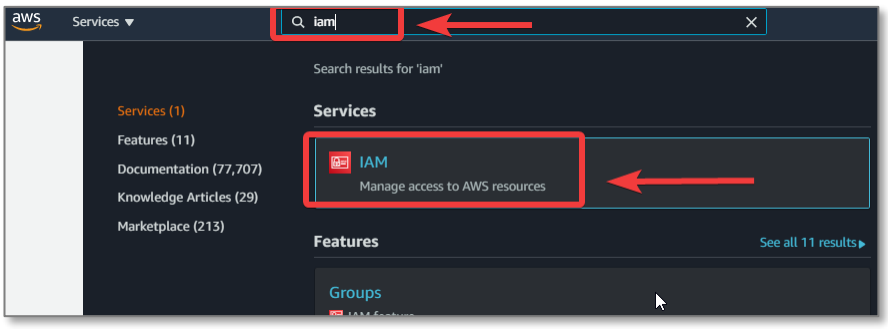


Creating an IAM User

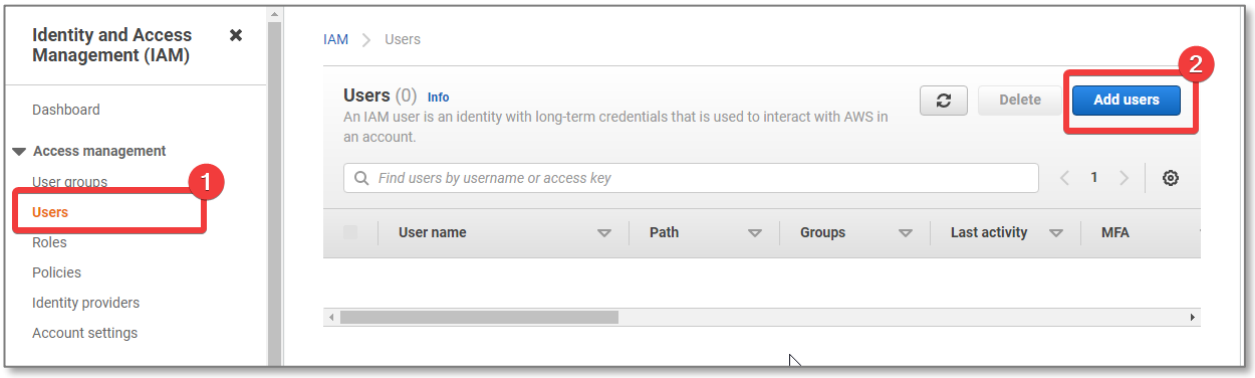
In this section, we will use the Identity and Access Management (IAM) service to create a user with console access and programmatic access.

From the AWS console click search *IAM*

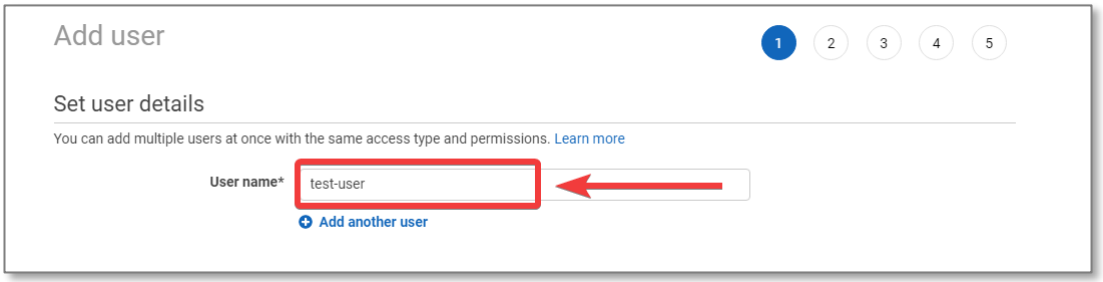
Select *IAM*



Select *Users -> Add user*



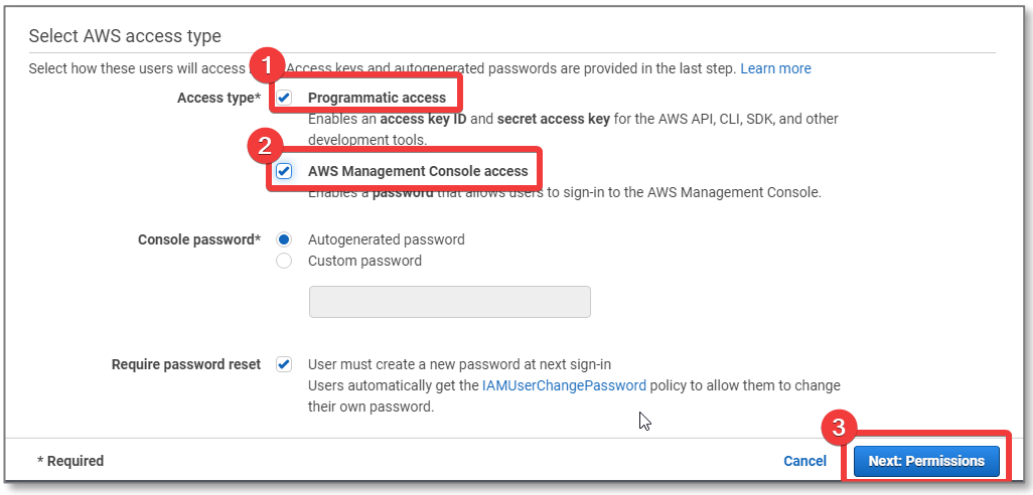
Give the user a name



Check *Programmatic access*

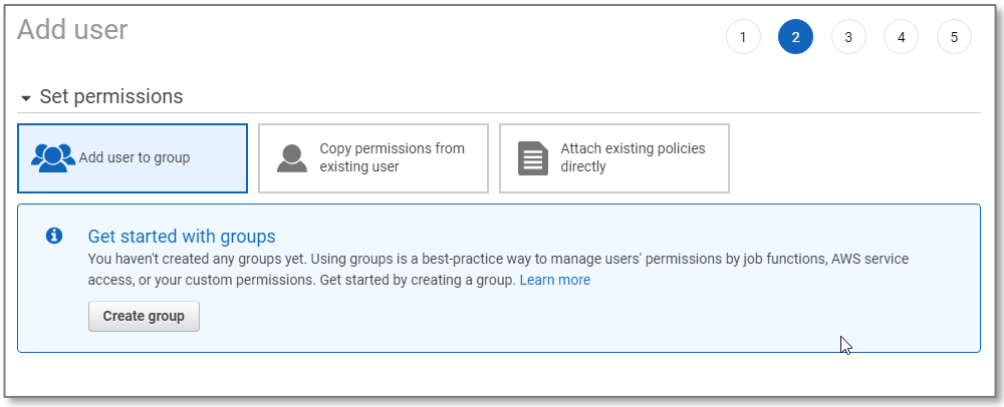
Check *AWS Management Console access*

Click *Next: Permissions*



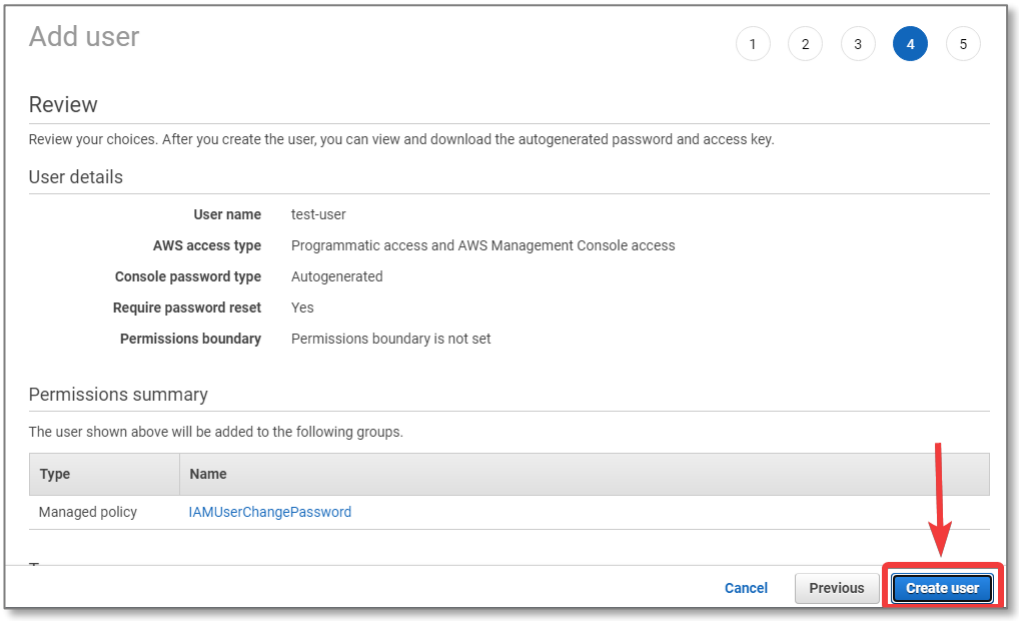
We won't set any permissions for the user at this point.

Click *Next: Tags*
We won't set any tags.

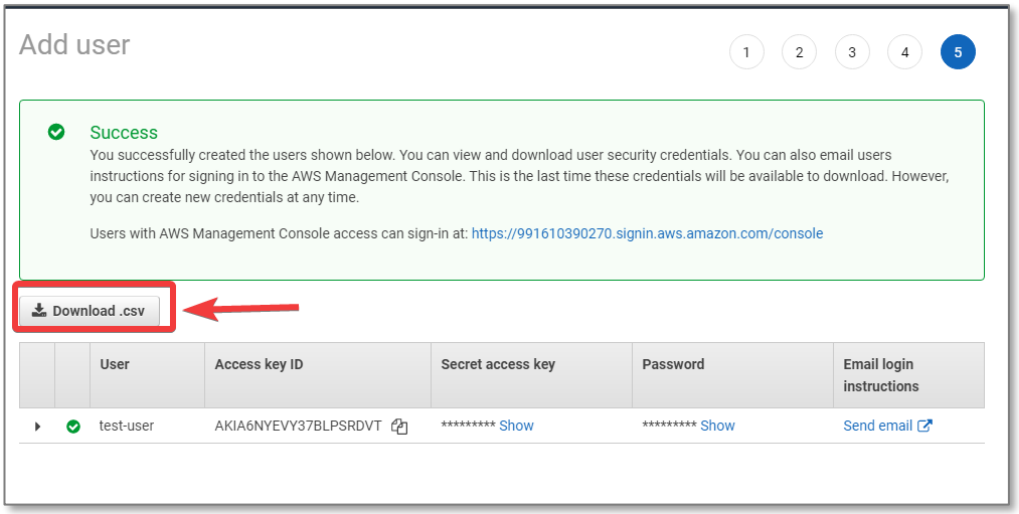


Click *Next Review*

Click *Create user*



You would normally download the csv file containing the user credentials (access key and secret access key) to a safe location.



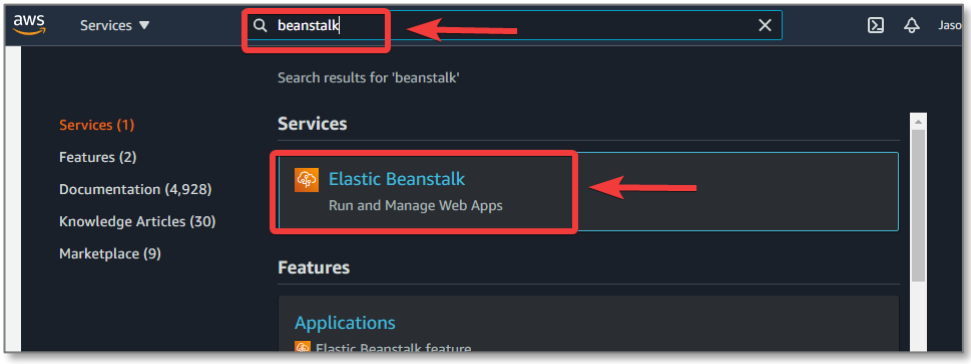
Clean Up

Go back to the *Users* dashboard and delete the user

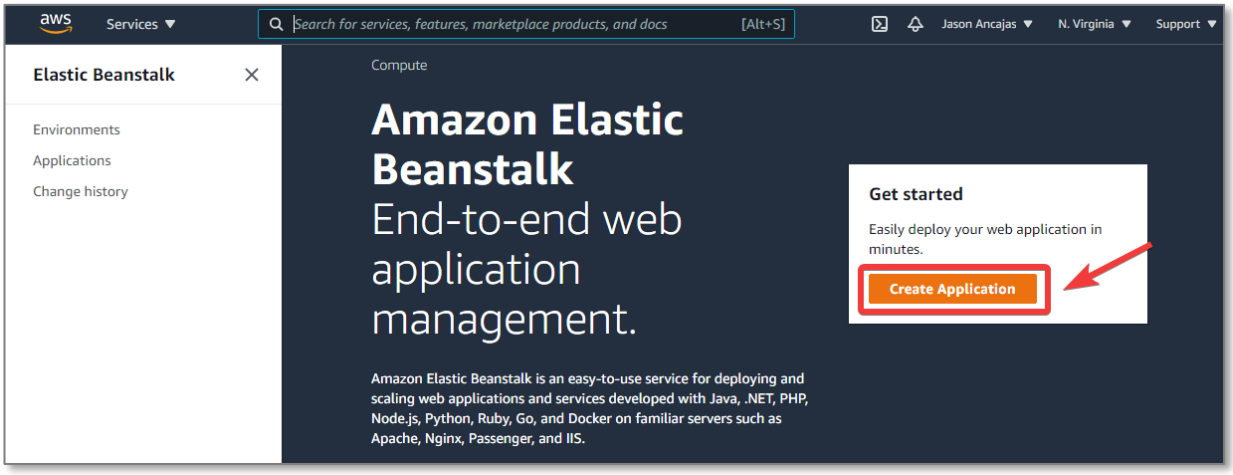
Creating a Highly Available Architecture with Elastic Beanstalk

In this section, we will create a highly available and fault tolerant architecture using the AWS Elastic Beanstalk service.

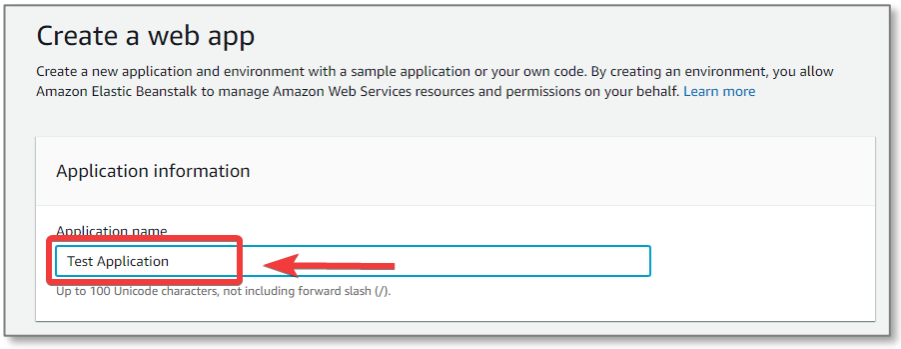
From the Management Console search *Elastic Beanstalk*



Click *Create Application*

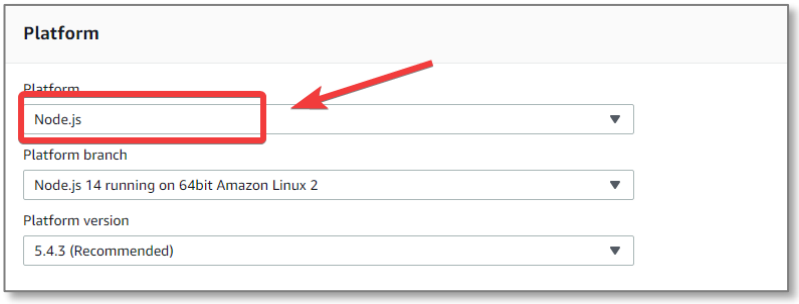


Give your application a name *Test Application.*



Scroll down to *Platform*

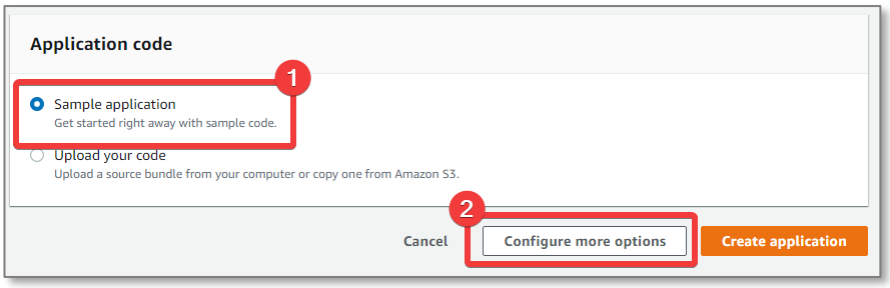
Select *Node.js* as the platform



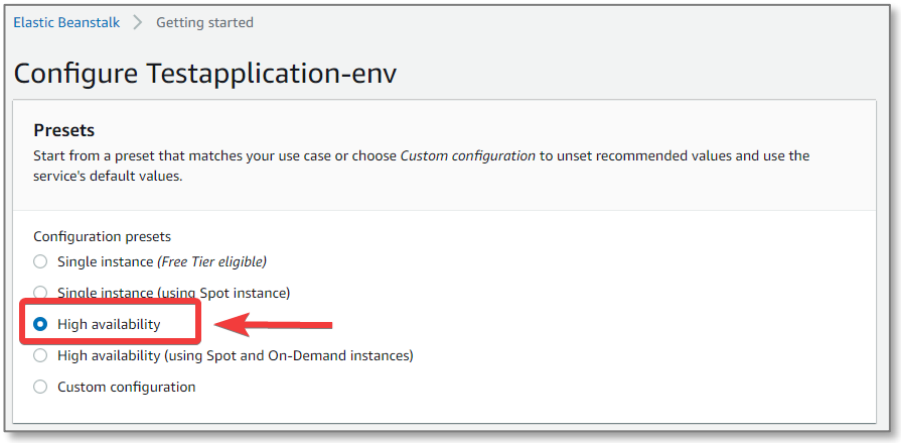
Scroll down to *Application code*

Select *Sample application*

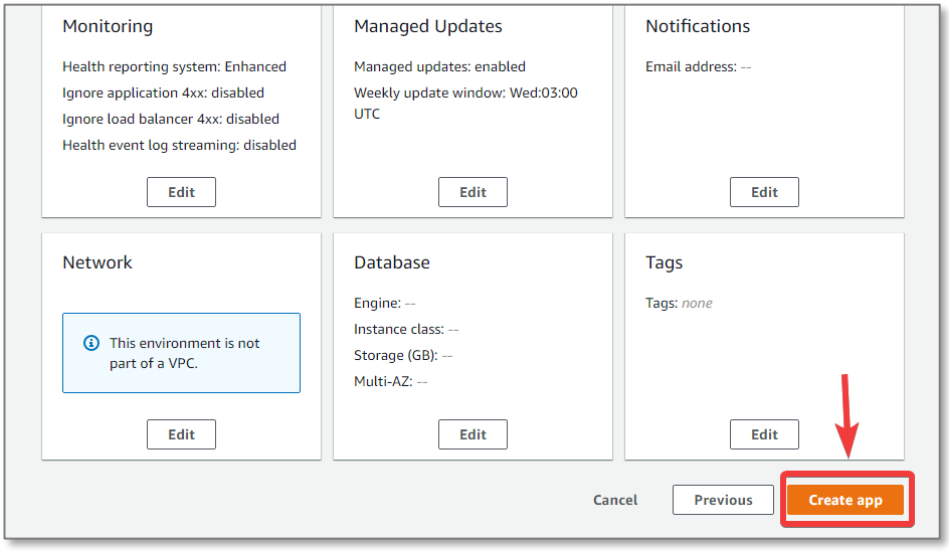
Click *Configure more options*



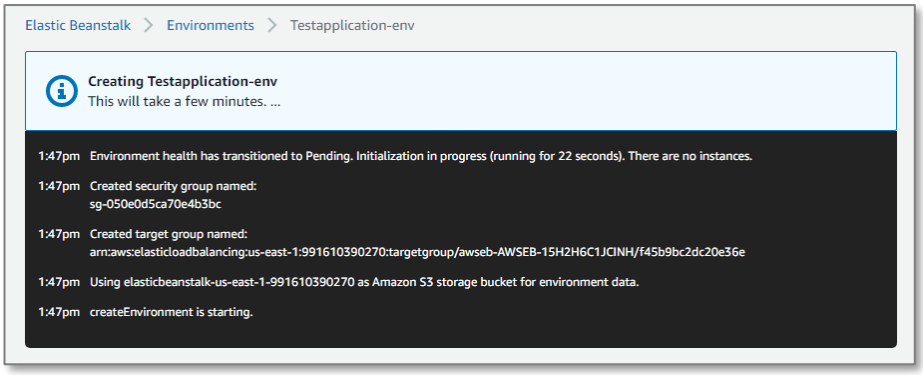
Select High availability



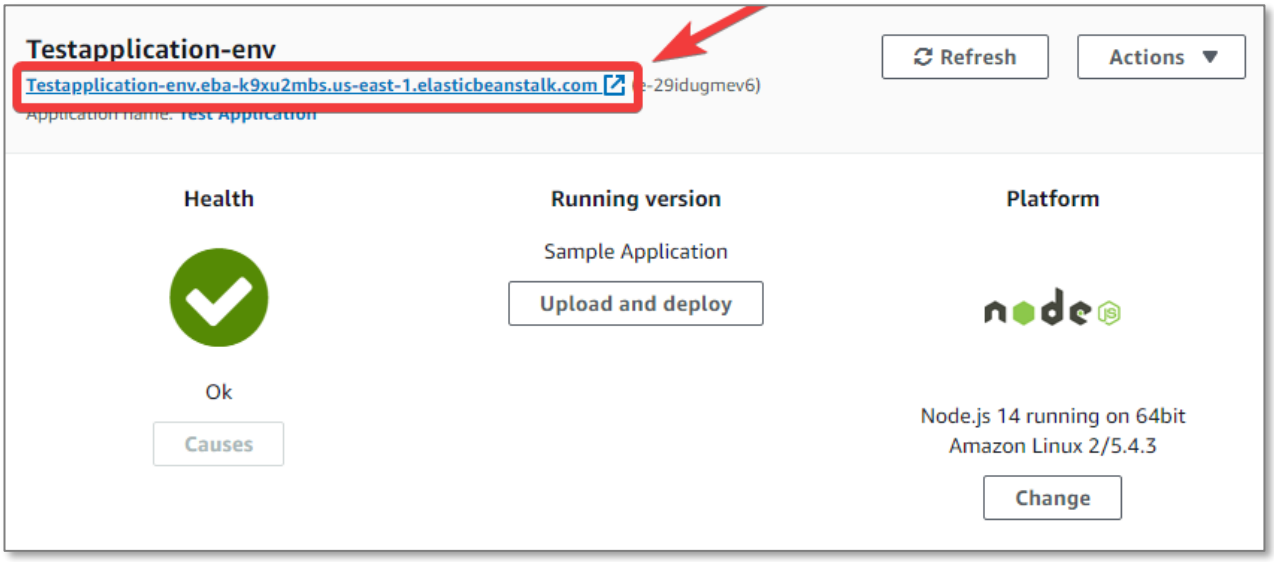
Scroll down and click *Create app*



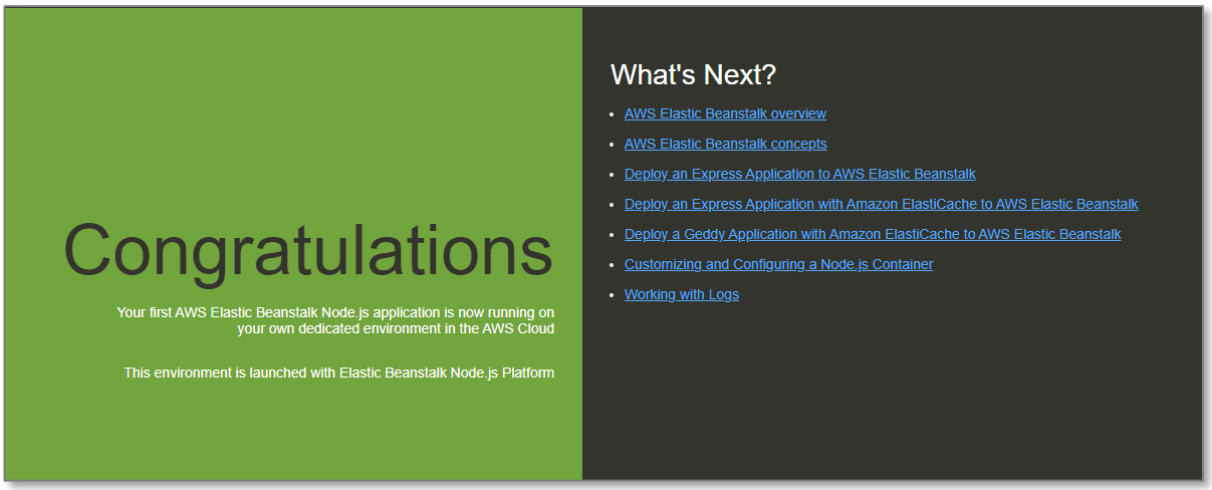
Your application will now start being created.
After some time, your environment will be created.



Click on the website URL



You will now see the Sample Application



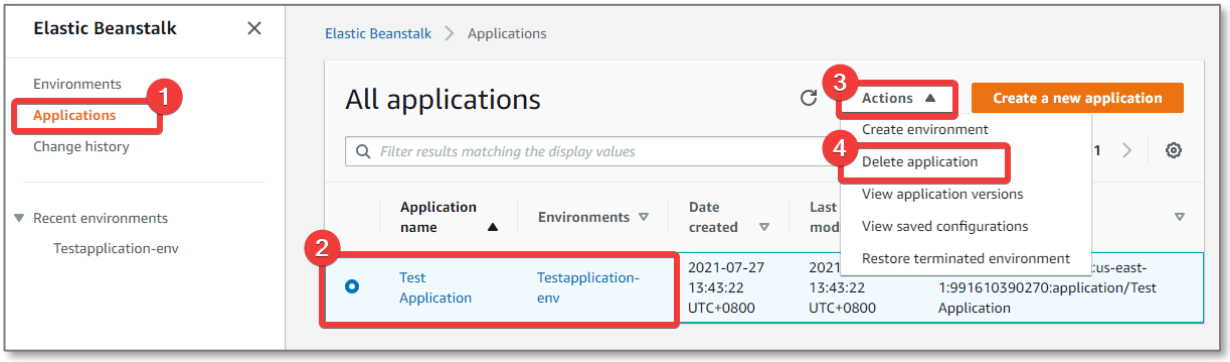
Clean Up

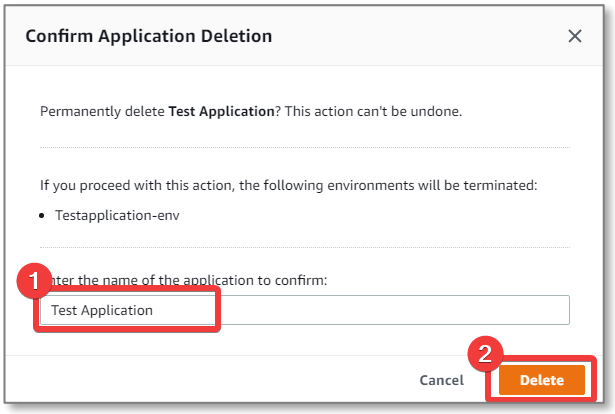
We will now delete the environment so that you will not be billed by AWS.

Click on *Applications*

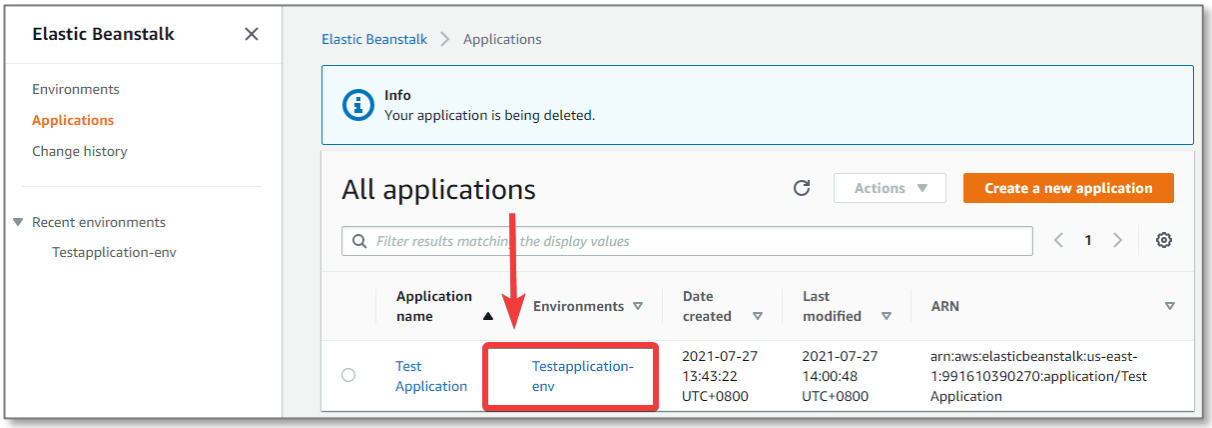
Select the application

Select *Actions -> Delete application*





Click on the *Environments*



You will now see your environment is being terminated.