



lab



lab title

**Introduction to AWS**

**V1.24**



Course title

**BackSpace Academy**  
**AWS Certified Associate**



# ▶ Table of Contents

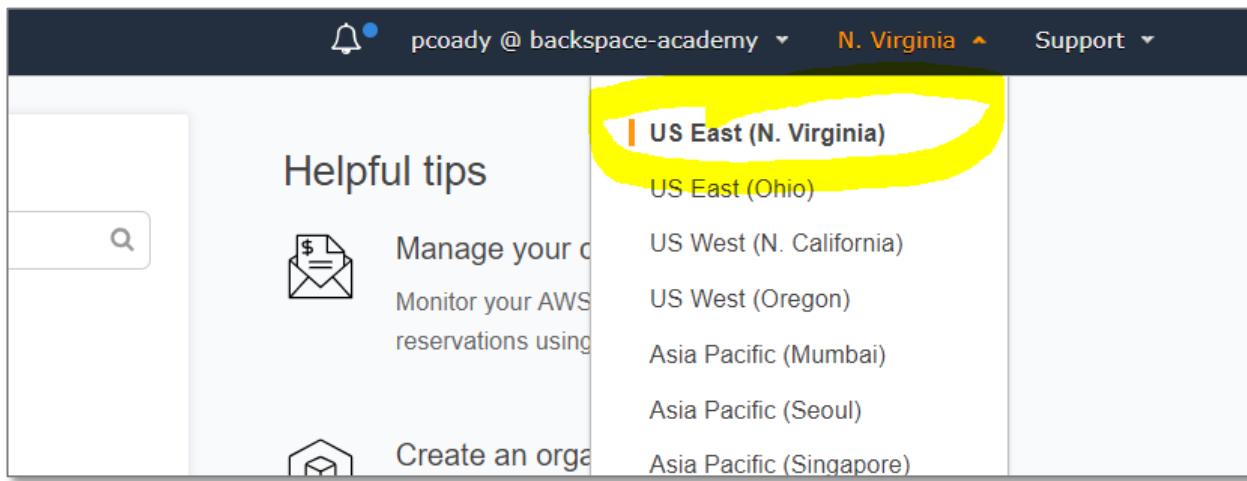
## Contents

<b>Table of Contents .....</b>	1
<b>About the Lab .....</b>	3
<b>Checking your AWS Usage and Monthly Bill .....</b>	4
<b>Creating an S3 Bucket and Uploading Files.....</b>	5
Uploading Files to your Bucket.....	8
Downloading files from your bucket .....	10
Troubleshooting.....	11
Clean Up.....	11
<b>Creating a SQL Database with RDS.....</b>	14
Connecting to your RDS Instance .....	18
Troubleshooting Connection Issues .....	22
Connecting to your RDS Instance using the Command Line .....	24
Clean Up.....	26
<b>Creating a Web Server with EC2.....</b>	28
Viewing your web server .....	32
Finding the Username and Password for your WordPress application.....	35
Clean up .....	37
<b>Sending emails with Amazon SES .....</b>	38
Requesting full access to SES.....	40
<b>Creating a Billing Alert with CloudWatch and SNS.....</b>	41
Enabling Billing Alerts .....	41
Creating a CloudWatch Alarm .....	42
<b>Creating an IAM User .....</b>	45
<b>Creating a Highly Available Architecture with Elastic Beanstalk .....</b>	48
Clean Up.....	52



# ► 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 AWS Certified Associate 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 lastest version with any updates or corrections.**

# ▶ Checking your AWS Usage and Monthly Bill

In this section we will learn how to use the AWS Billing & Cost Management Dashboard to keep track of costs.

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.

**Billing & Cost Management Dashboard**

**What's New in AWS Billing and Cost Management?**

- Manage your spend with AWS Budgets
- Visualize your costs and usage with the newly-optimized Cost Explorer
- Easily upload your Cost and Usage Reports into Redshift and QuickSight

**Spend Summary** **Cost Explorer**

Welcome to the AWS Account Billing console. Your last month, month-to-date, and month-end forecasted costs appear below.

Current month-to-date balance for August 2017

**\$0.01**

**Month-to-Date Spend by Service**

The chart below shows the proportion of costs spent for each service you use.

Service	Amount
SES	\$0.01
S3	\$0.00
EC2	\$0.00
CloudWatch	\$0.00
Other Services	\$0.00
Tax	\$0.00
<b>Total</b>	<b>\$0.01</b>

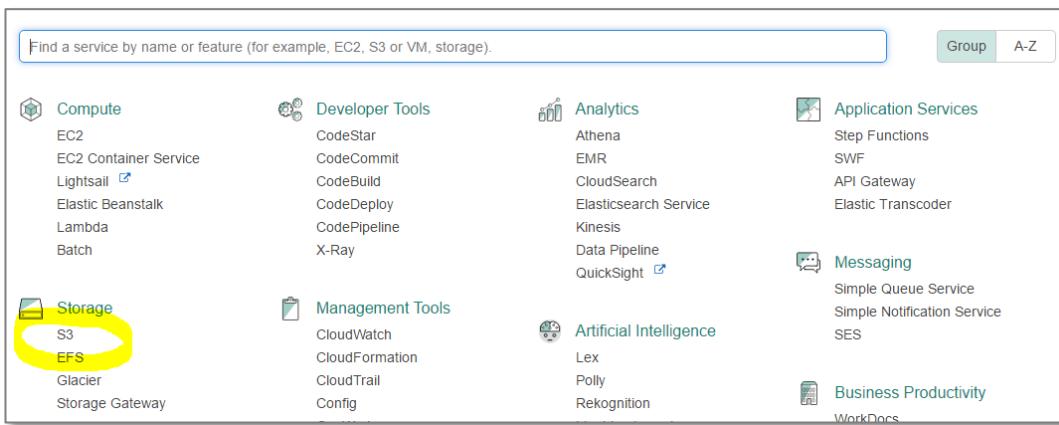
**Top Free Tier Services by Usage**

Service	Month-to-date usage/Free Tier limit	Forecasted month-end usage/Free Tier limit
S3 - Puts	62.00% (1,240,000/2,000 Requests)	120.13% (2,402,500/2,000 Requests)
EBS - Snapshots	47.99% (0.48/1 GB-Mo)	92.99% (0.93/1 GB-Mo)

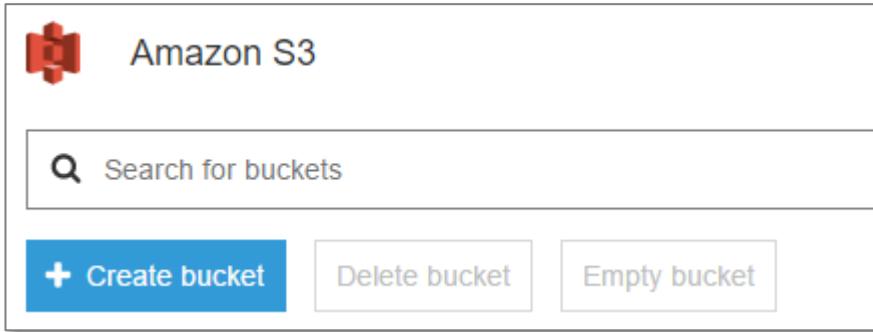
# ▶ Creating an S3 Bucket and Uploading Files

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

Click on the services menu and select S3.



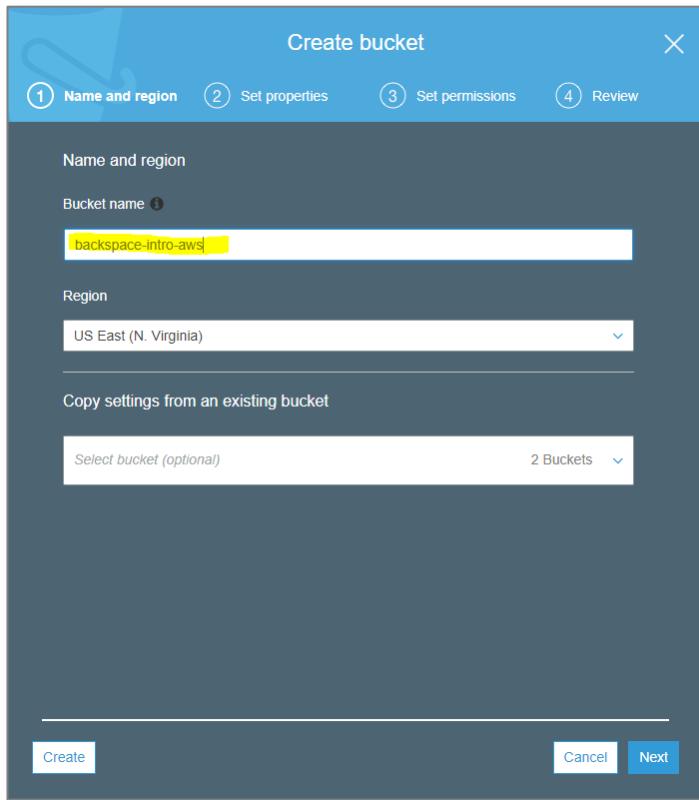
Click on Create Bucket



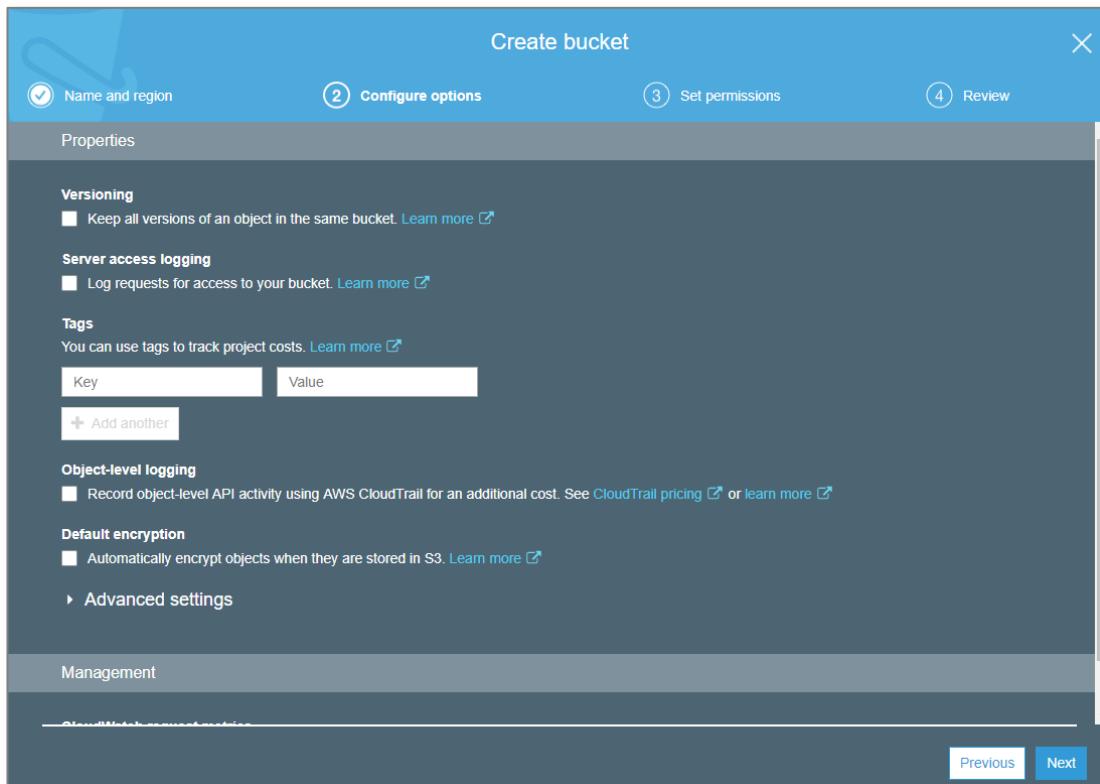
The create bucket dialog box will appear.

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

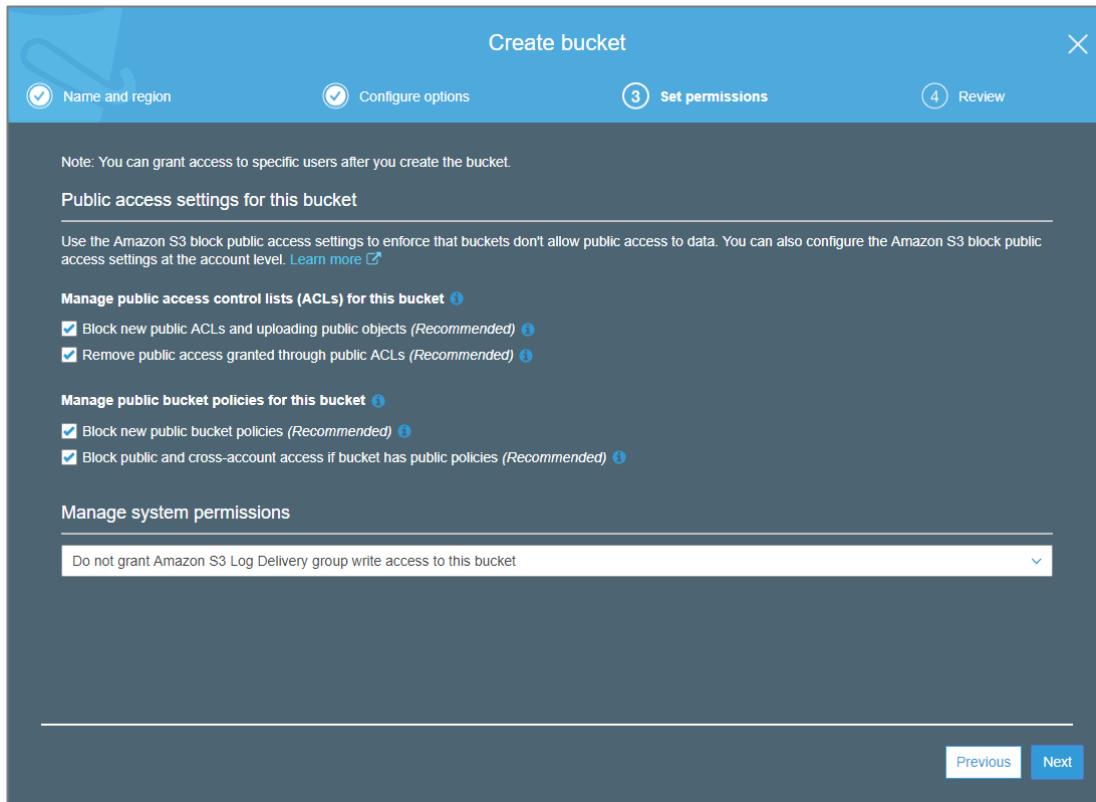
Click 'Next'



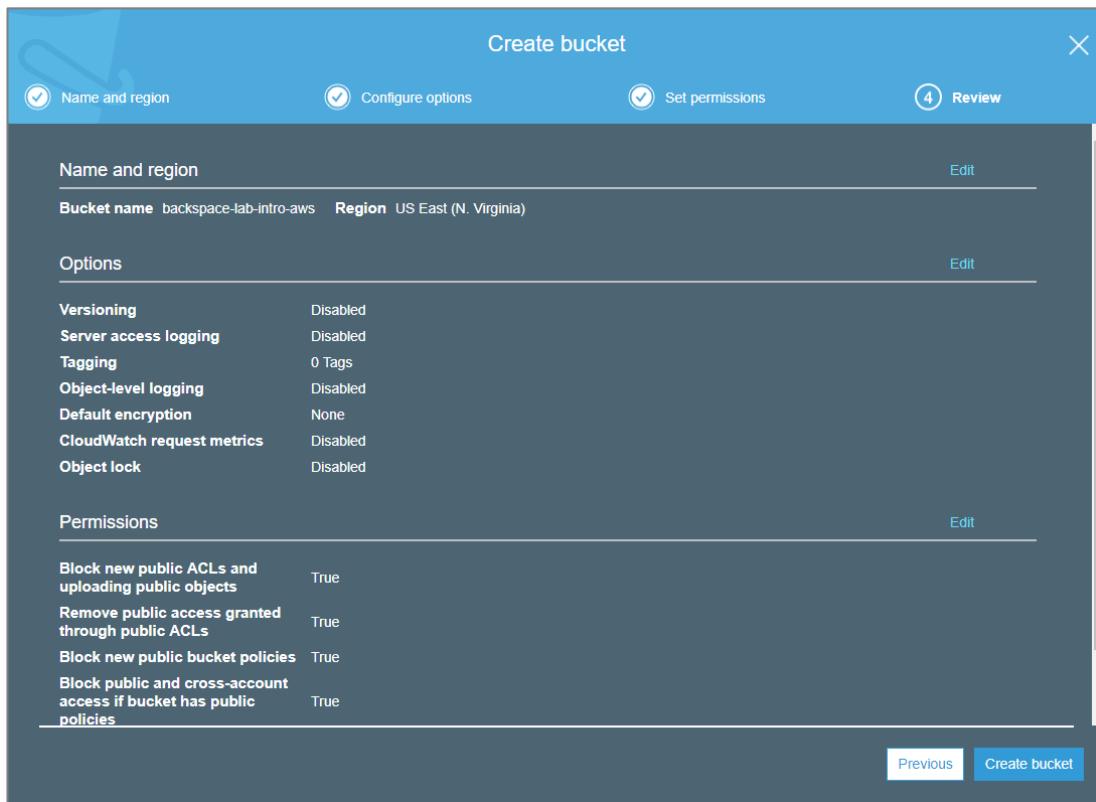
Leave as is and click 'Next"



Leave as is and click 'Next"

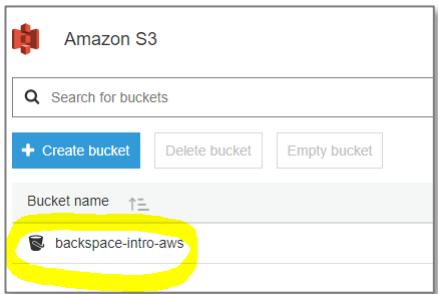


Click 'Create Bucket"

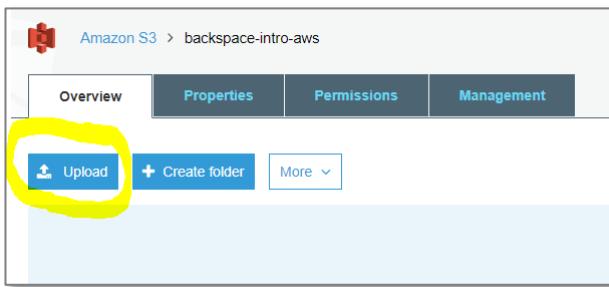


## Uploading Files to your Bucket

Click on the link to the bucket

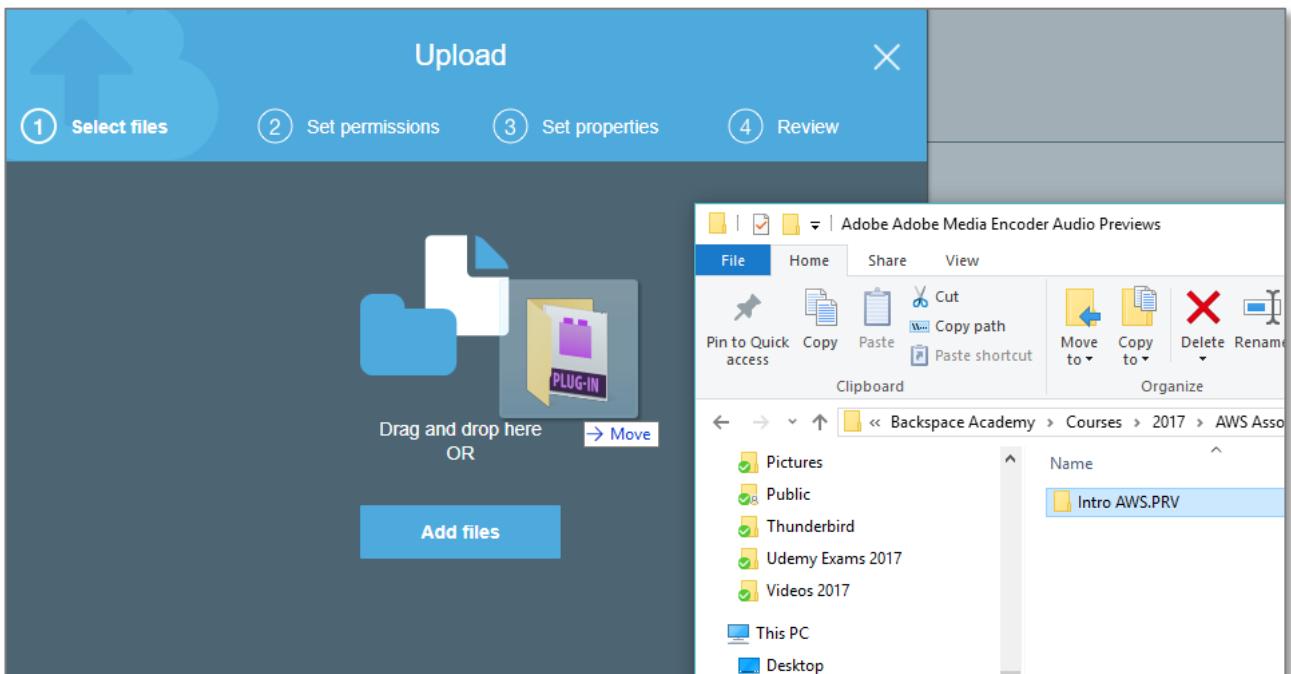


Select 'Upload'

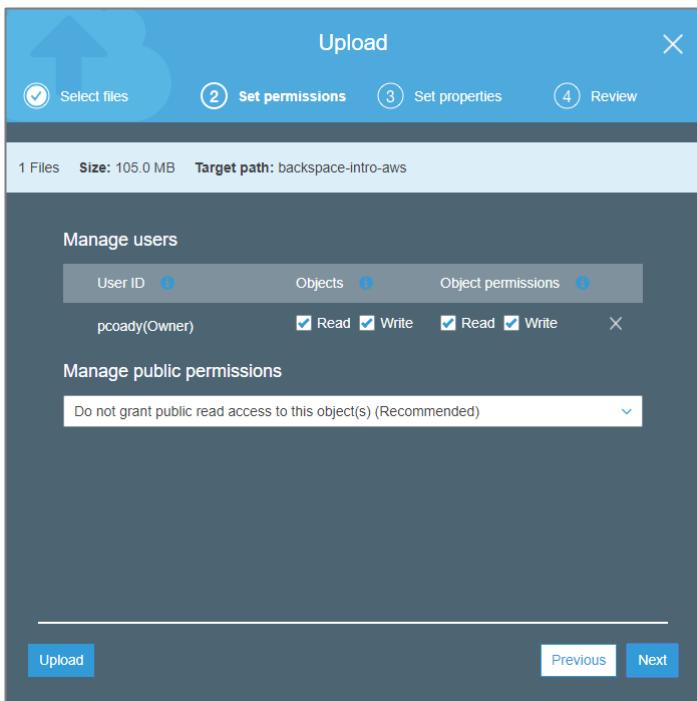


Drag a folder with files onto the form.

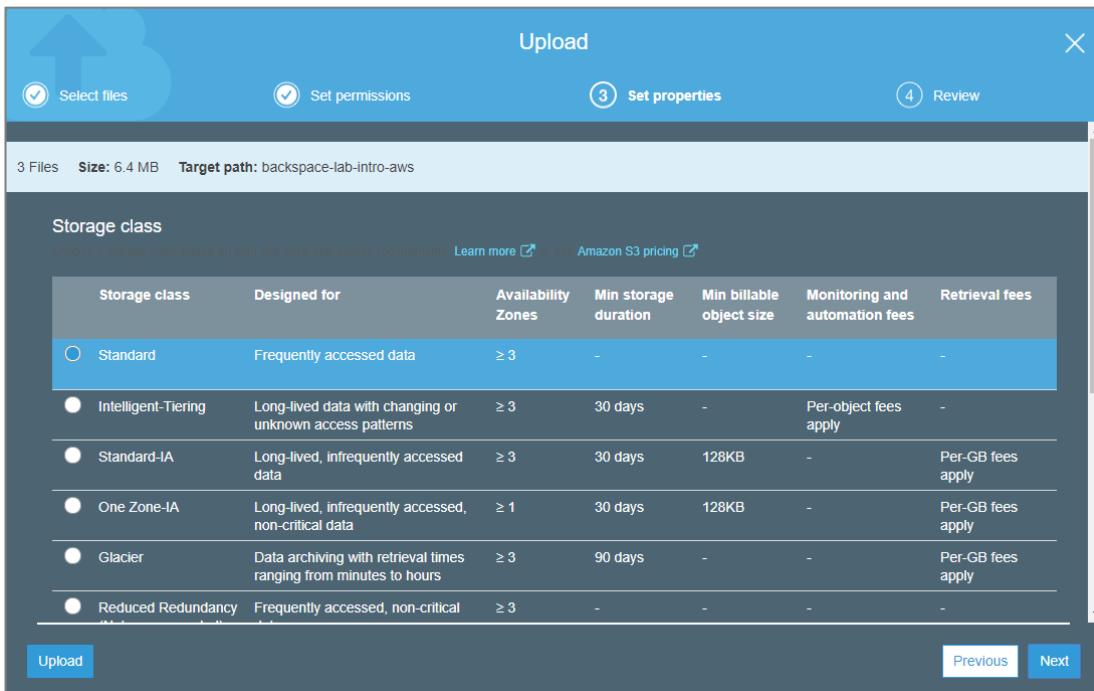
Click Next



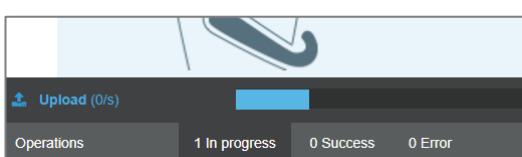
Leave as is and click 'Next"



Leave as is and click 'Next"



Click 'Upload"



Your upload will eventually complete.

Amazon S3 > backspace-intro-aws

Overview Properties Permissions Management

Type a prefix and press Enter to search. Press ESC to clear.

Upload + Create folder More

US East (N. Virginia)

Name Last modified Size Storage class

img

Viewing 1 to 1

## Downloading files from your bucket

Click the link for your folder

Upload + Create folder More

Name

img

Select a file

Upload + Create folder More

Name Last modified

BackSpace.png Aug 14, 2017

BackSpace.psd Aug 14, 2017

Select "More", "Download As"

Upload + Create folder More

Get size

Download as

Rename

Delete

Undo delete

Name

BackSpace.png

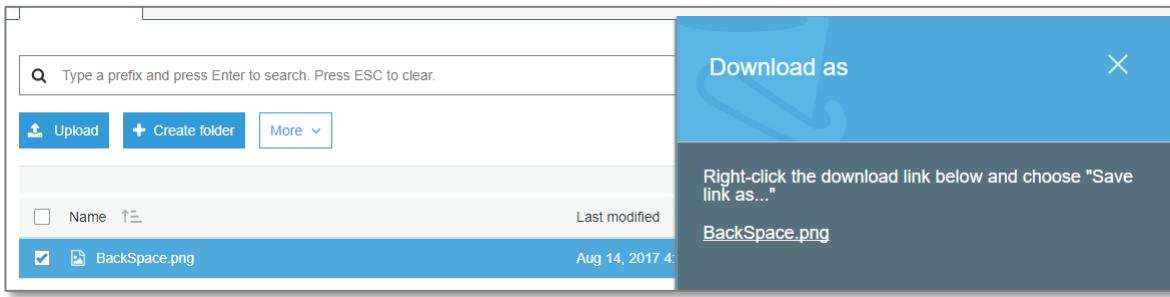
BackSpace.psd

Last modified

Aug 14, 2017

Aug 14, 2017

Click the download link to download the file.



## Troubleshooting

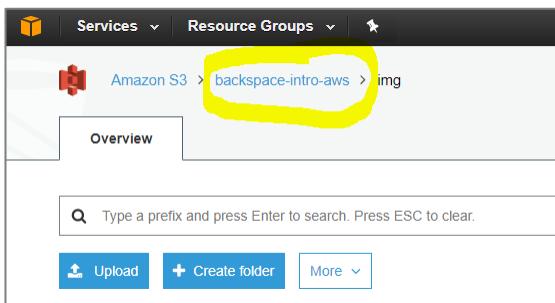
If you get the following screen it means you have clicked on the Object URL and not the download link as detailed above. You cannot access files directly from a URL as they have private access.



## Clean Up

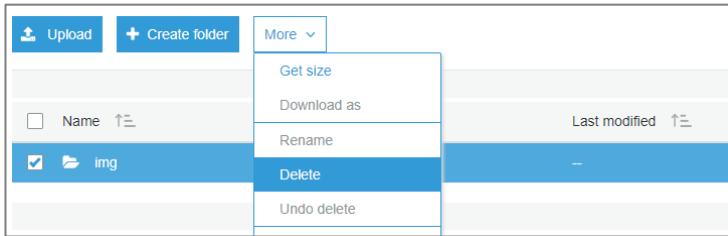
We will now delete the files and bucket so that you will not be billed by AWS.

Go back to your bucket by clicking its link.

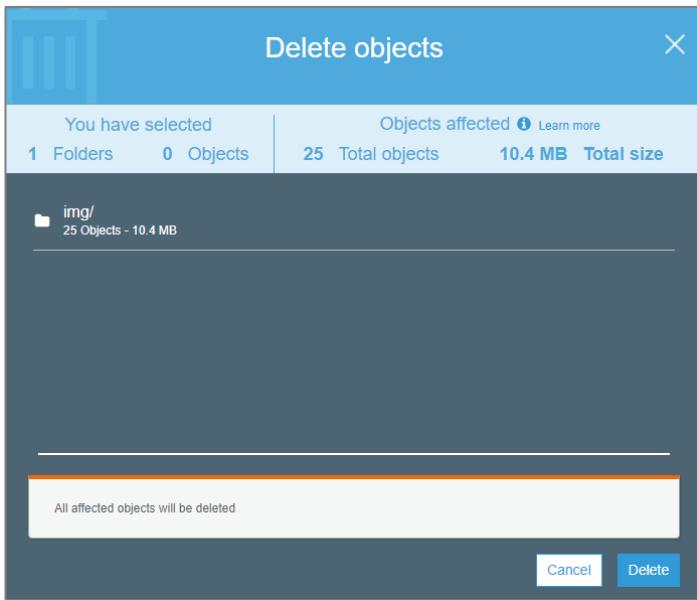


Select the folder

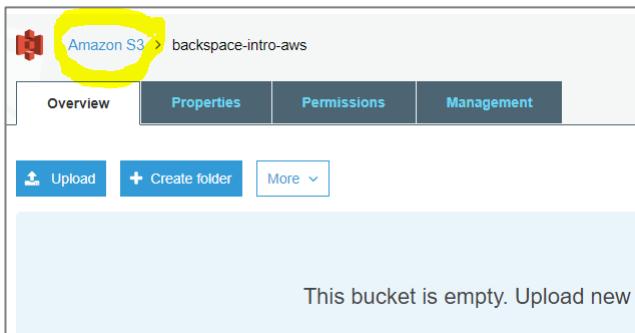
Select “More”, “Delete”



Click "Delete"



Go back to the S3 dashboard by clicking the link

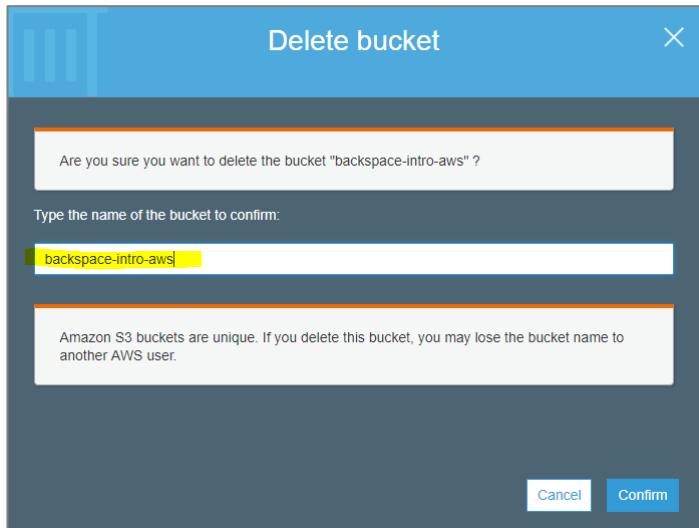


Click on the bucket line but not on the bucket link to select the bucket.

Click "Delete Bucket"

The screenshot shows the Amazon S3 console interface. At the top, there are links to 'Switch to the old console', 'Discover the new console', and 'Quick tips'. Below that is a search bar labeled 'Search for buckets'. There are three buttons: '+ Create bucket', 'Delete bucket' (which is highlighted with a yellow box), and 'Empty bucket'. To the right, it shows '3 Buckets' and '1 Regions'. A table lists one bucket: 'backspace-intro-aws' located in 'US East (N. Virginia)' with a creation date of 'Aug 14, 2017 4:19:05 PM'. A large yellow circle highlights the entire row for the bucket.

Confirm the name of the bucket to delete



# Creating a SQL Database with RDS

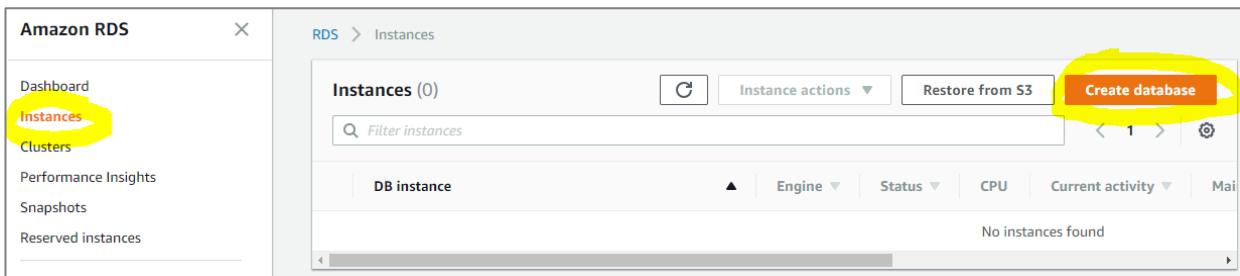
In this section, we will use the Relational Database Service to create a database. We will also connect in to the database.

From the AWS console select 'RDS' from the Database services.



Select 'instances'

Select 'Create database'



Select 'MySQL'

Select 'Only enable options eligible for RDS free tier usage'

Click 'Next'

Step 1  
**Select engine**

Step 2  
Specify DB details

Step 3  
Configure advanced settings

RDS > Instances > Launch DB instance

## Select engine

**Engine options**

Amazon Aurora

MySQL

MariaDB

PostgreSQL

Oracle

Microsoft SQL Server

**MySQL**  
MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 16 TB.
- Instances offer up to 32 vCPUs and 244 GiB Memory.
- Supports automated backup and point-in-time recovery.
- Supports cross-region read replicas.

Only enable options eligible for RDS Free Usage Tier [info](#)

[Cancel](#) **Next**

### Select db.t2.micro instance class

DB instance class [info](#)

**db.t2.micro — 1 vCPU, 1 GiB RAM**

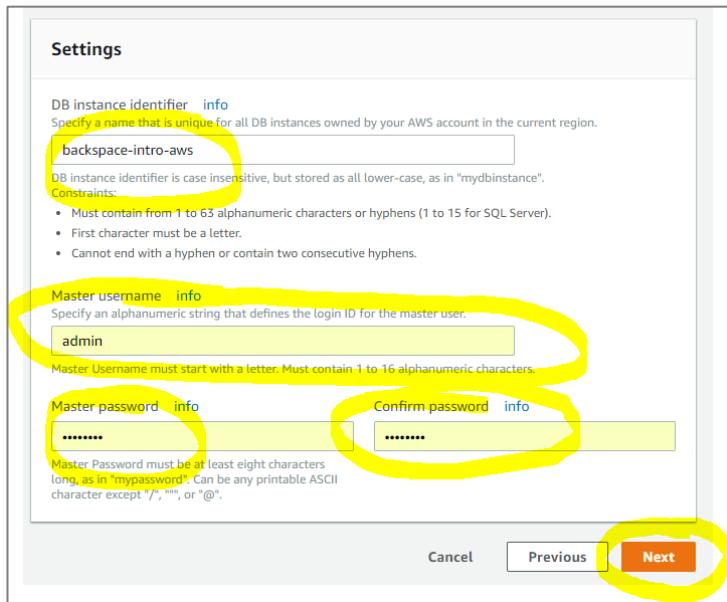
Multi-AZ deployment [info](#)

Create replica in different zone  
Creates a replica in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.

Give your instance a name/identifier.

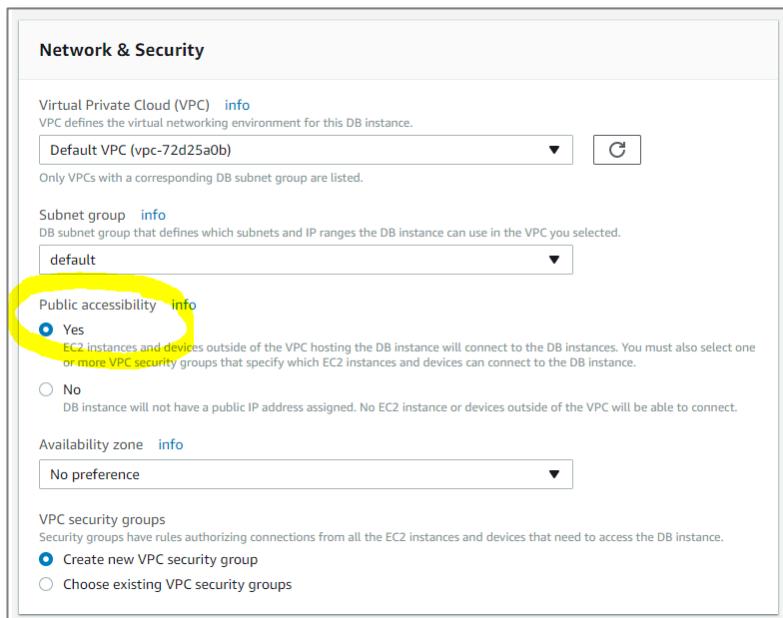
Fill in a master username and password

Click 'Next"



Leave settings for Network and Security as below.

Make sure it is publicly accessible (we will look at security later on in the course)



Enter a database name.

Leave all other options default as below.

**Database options**

Database name   

Note: if no database name is specified then no initial MySQL database will be created on the DB Instance.

Database port   

DB parameter group [info](#)  ▼

Option group [info](#)  ▼

Copy tags to snapshots

IAM DB authentication [info](#)

Enable IAM DB authentication  
Manage your database user credentials through AWS IAM users and roles.

Disable

Change 'Backup Retention Period' to disable automated backups.

**Backup**

⚠ Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to detail [here](#).

Backup retention period [info](#)   

Select the number of days that Amazon RDS should retain automatic backups of this DB instance.

A backup retention period of zero days will disable automated backups for this DB Instance.

Backup window [info](#)

Select window

No preference

Scroll down and click 'Create database'

ℹ Ensure that General, Slow Query, and Audit Logs are turned on. Error logs are enabled by default. [Learn more](#)

**Maintenance**

Auto minor version upgrade [Info](#)

Enable auto minor version upgrade  
Enables automatic upgrades to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the DB instance.

Disable auto minor version upgrade

Maintenance window [Info](#)

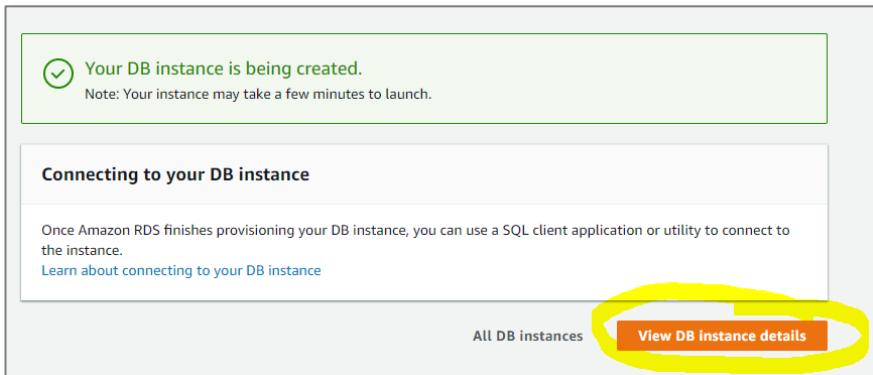
Select the period in which you want pending modifications or patches applied to the DB instance by Amazon RDS.

Select window

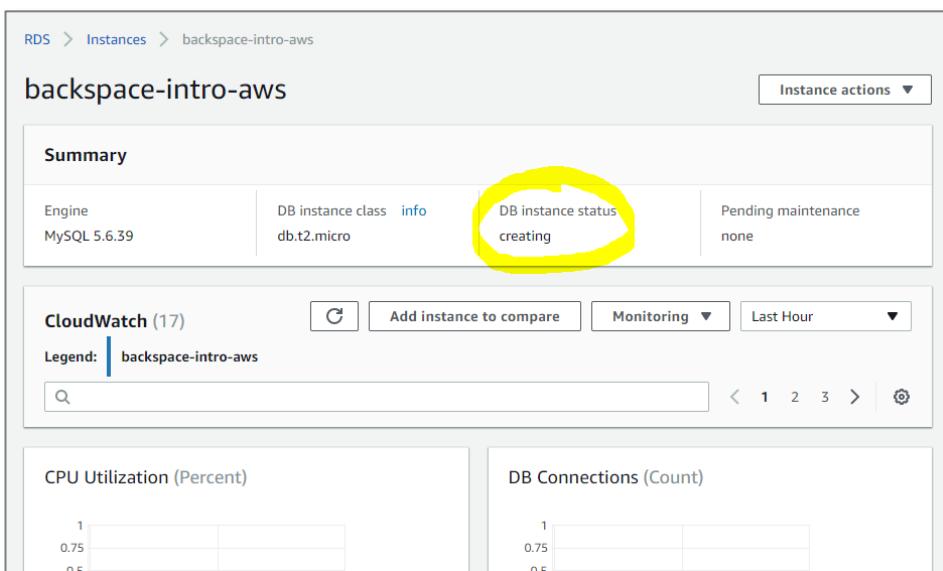
No preference

Cancel Previous Create database

Click 'View your DB Instances



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 from

<https://dev.mysql.com/downloads/workbench/>

Wait for your instance status to be 'available'

Summary

Engine MySQL 5.6.39	DB instance class info db.t2.micro	DB instance status available	Pending maintenance none
------------------------	--	---------------------------------	-----------------------------

CloudWatch (17)

CPU Utilization (Percent) DB Connections (Count)

Scroll down and copy the database server endpoint

Connect

Endpoint backspace-intro-aws.cv9gzwmqrrv.us-east-1.rds.amazonaws.com	Port 3306	Publicly accessible Yes
---	--------------	----------------------------

Security group rules (2)

Security group	Type	Rule
rds-launch-wizard-5 (sg-2e8b8e58)	CIDR/IP - Inbound	103.75.207.55/32
rds-launch-wizard-5 (sg-2e8b8e58)	CIDR/IP - Outbound	0.0.0.0/0

Open the MySQL Workbench application click to add a new connection

Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

Browse Documentation > Read the Blog > Discuss on the Forums >

MySQL Connections +

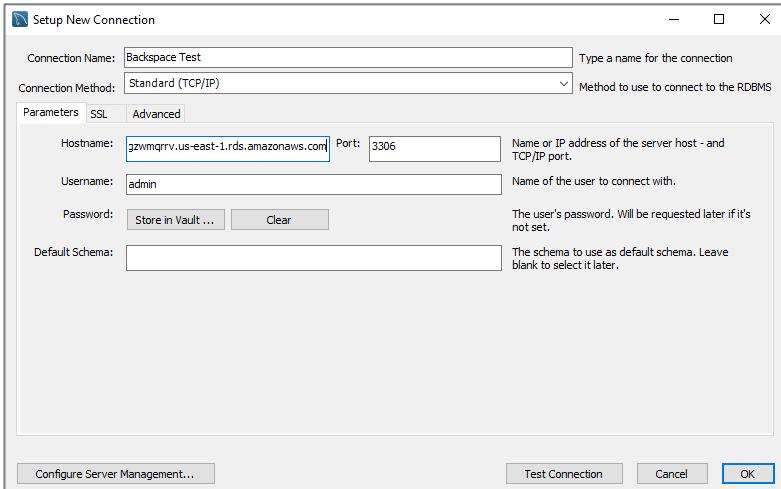
Give the connection a name.

The Hostname will be the RDS server endpoint with the ':3306" removed from the end.

The port will be 3306.

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

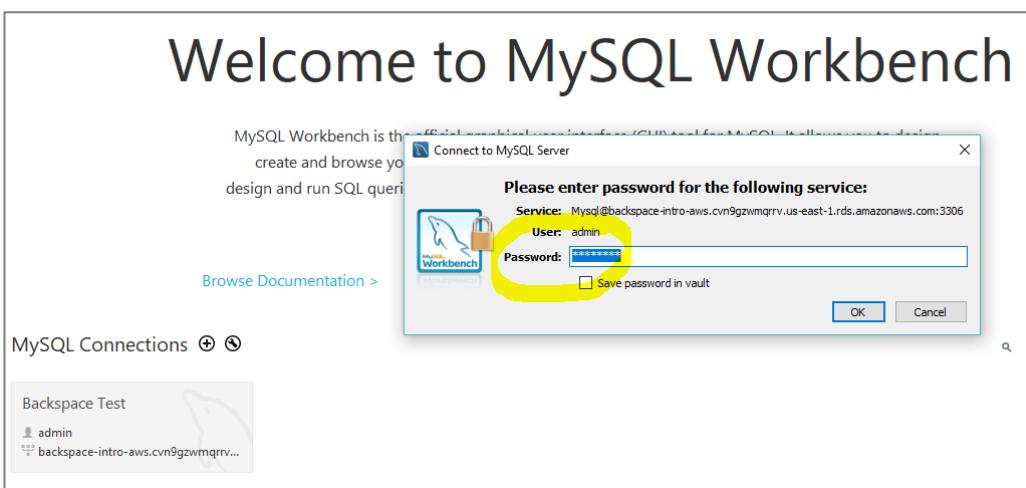
Click OK



Click on the Connection

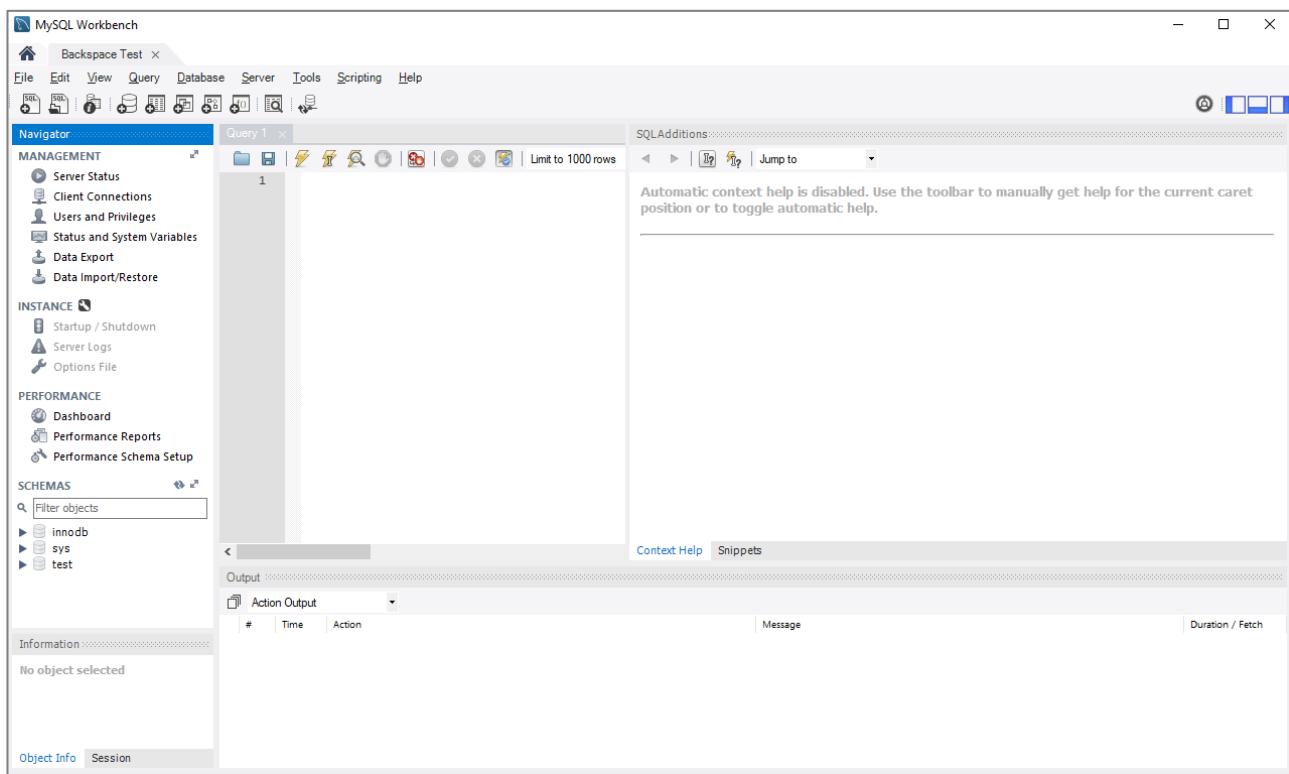


Enter the password you created in RDS for your master username

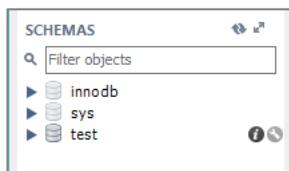


You will soon be connected to your database server

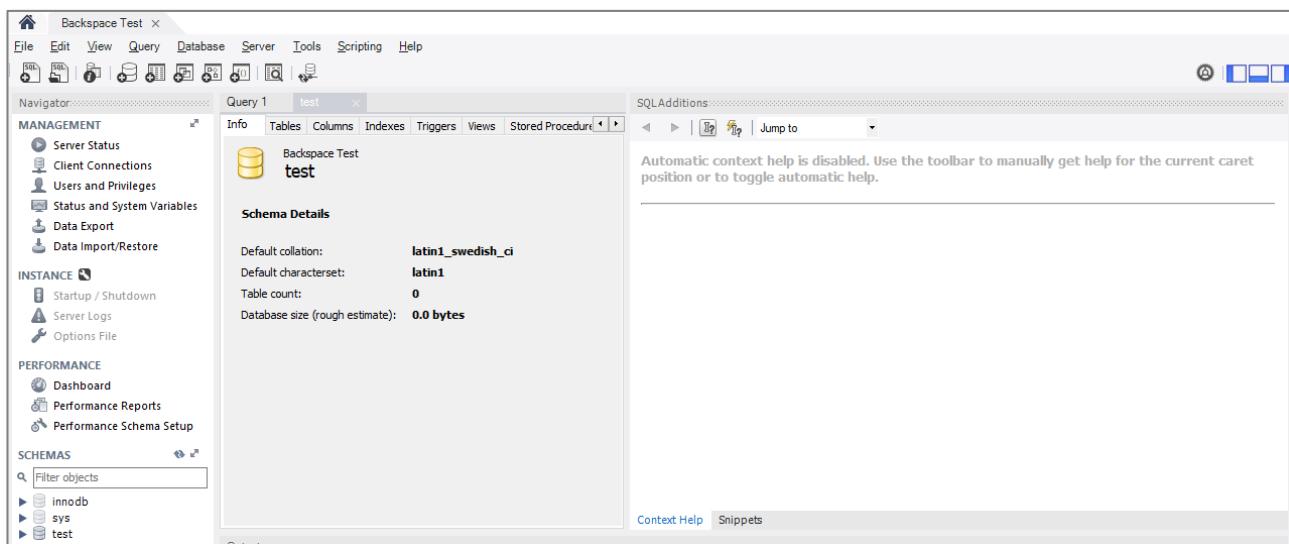
If you cannot connect then please see the 'Troubleshooting Connection Issues' below.



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.



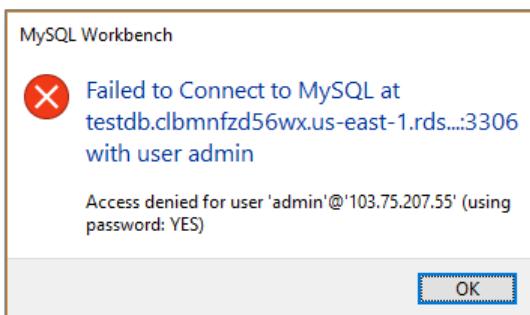
You then get an information screen for the database.



## Troubleshooting Connection Issues

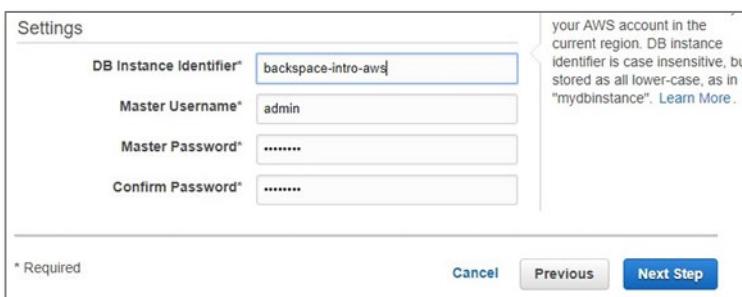
If you are getting connection errors then check the following:

### Wrong Username / Password

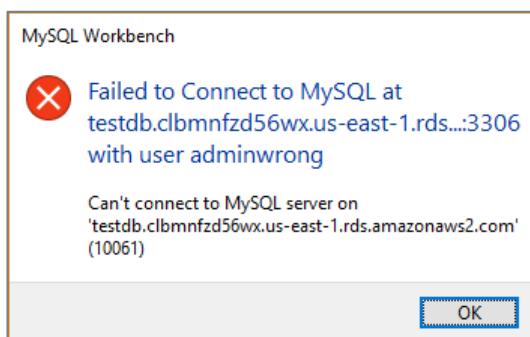


Make sure you use the correct username and password.

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



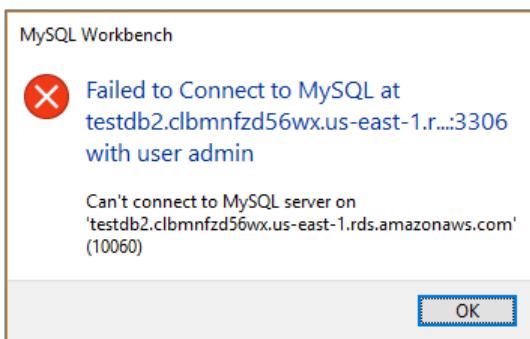
### Bad Connection String



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. 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

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.

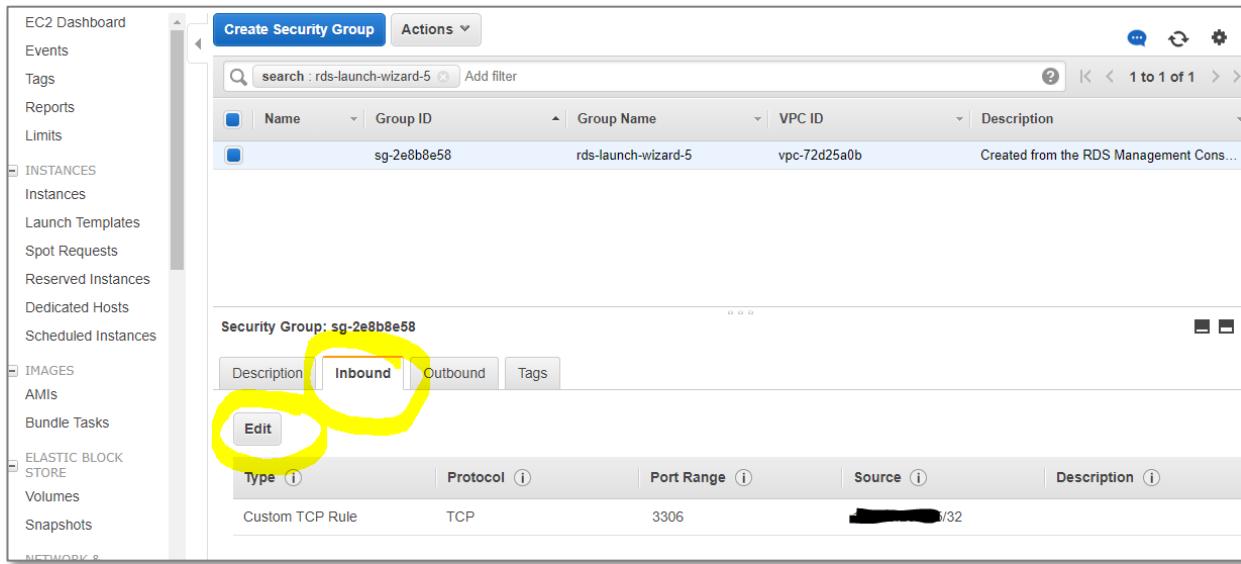
Click the security group

Connect											
Endpoint	Port	Publicly accessible									
backspace-intro-aws.cvn9gzwmqrw.us-east-1.rds.amazonaws.com	3306	Yes									
<b>Security group rules (2)</b> <table border="1"> <thead> <tr> <th>Security group</th> <th>Type</th> <th>Rule</th> </tr> </thead> <tbody> <tr> <td>rds-launch-wizard-5 (sg-2e8b8e58)</td> <td>CIDR/IP - Inbound</td> <td>[REDACTED] /32</td> </tr> <tr> <td>rds-launch-wizard-5 (sg-2e8b8e58)</td> <td>CIDR/IP - Outbound</td> <td>0.0.0.0/0</td> </tr> </tbody> </table>			Security group	Type	Rule	rds-launch-wizard-5 (sg-2e8b8e58)	CIDR/IP - Inbound	[REDACTED] /32	rds-launch-wizard-5 (sg-2e8b8e58)	CIDR/IP - Outbound	0.0.0.0/0
Security group	Type	Rule									
rds-launch-wizard-5 (sg-2e8b8e58)	CIDR/IP - Inbound	[REDACTED] /32									
rds-launch-wizard-5 (sg-2e8b8e58)	CIDR/IP - Outbound	0.0.0.0/0									

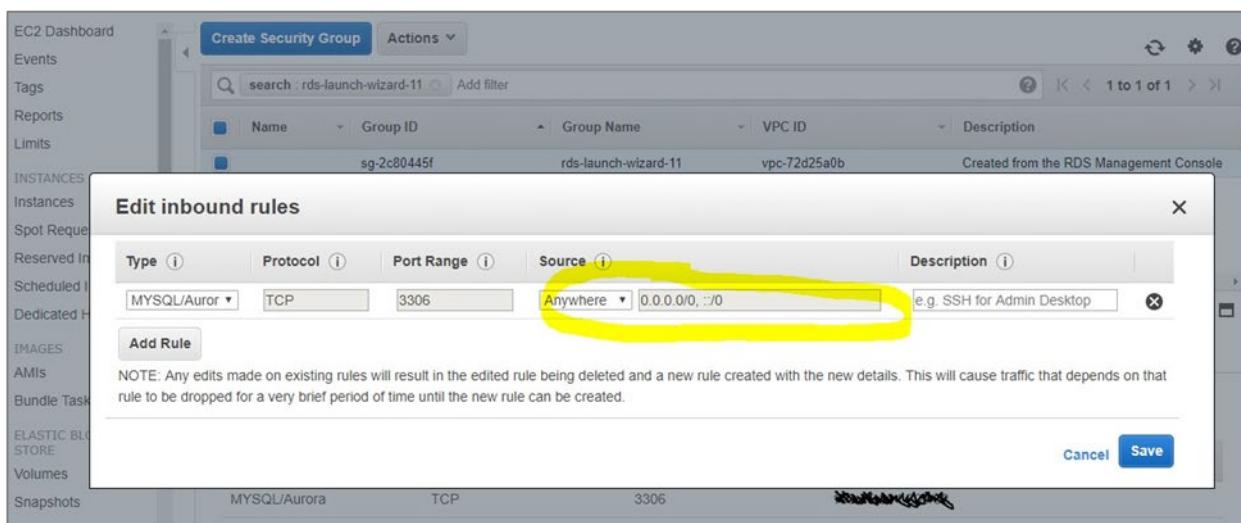
You will be taken to the EC2 console

Select the “Inbound” tab

Click “Edit”



Change inbound rule to “Anywhere” 0.0.0.0/0, ::/0



### *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.

## Connecting to your RDS Instance using the Command Line

### Mac Users:

Download the TAR archive (not the DMG file!) from <https://dev.mysql.com/downloads/shell/>

Extract the archive and run mysqlsh in the bin folder. This will open the shell.

## Windows Users:

The latest version of MySQL Workbench has MySQL Shell pre-installed. You don't need to install it separately.

Open a Windows Command Prompt.

Navigate to the installation directory of MySQL Workbench (default for Windows is *C:\Program Files\MySQL\MySQL Workbench 8.0 CE*)

Connect your database using the following command from the command line (if using Windows Powershell use *./mysql*)

Please note the instructions in the AWS docs are incorrect (missing password tag), use the correct command below.

```
mysql -h <endpoint> -P 3306 -u <mymasteruser> -p
```

Enter your password when requested.

After a while you will be connected to your RDS instance.

This will open the MySQL Shell

```
F:\>Program Files\MySQL\MySQL Workbench 8.0 CE>mysql -h backspace-intro-aws.clbmnfd56wx.us-east-1.rds.amazonaws.com -u admin -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.7.22 Source distribution

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> -
```

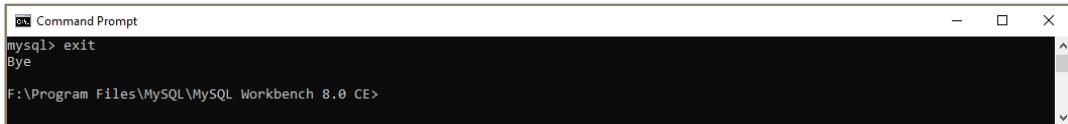
Enter the SQL command to list databases (**don't forget the ';' on the end**):

```
show databases;
```

```
SHOW DATABASES' at line 1
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| innodb |
| mysql |
| performance_schema |
| sys |
| test |
+-----+
6 rows in set (0.23 sec)

mysql> -
```

Type *exit* to leave the MySQL Shell



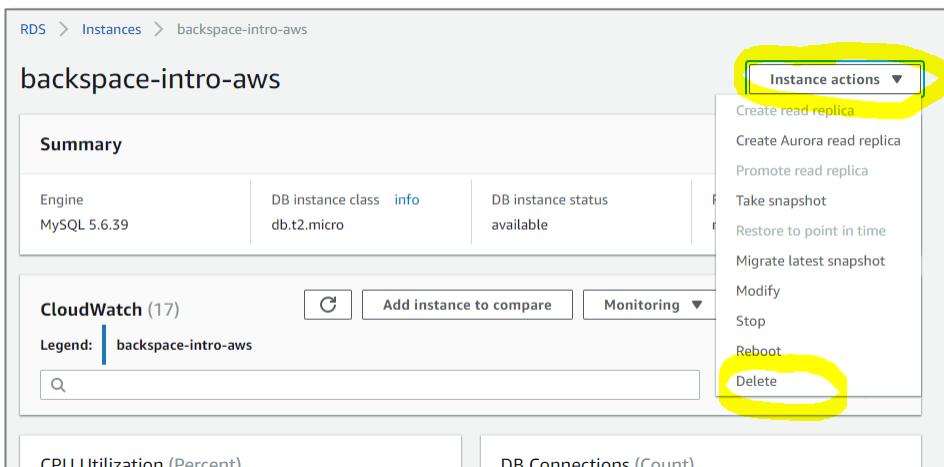
```
Command Prompt
mysql> exit
Bye
F:\Program Files\MySQL\MySQL Workbench 8.0 CE>
```

## Clean Up

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

Go back to the RDS console.

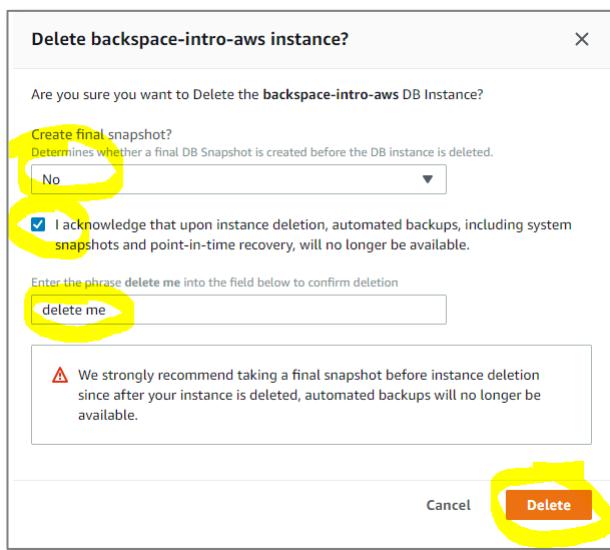
Click 'Instance Actions", "Delete" to terminate the instance



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 'Instances' to see its status as 'deleting'

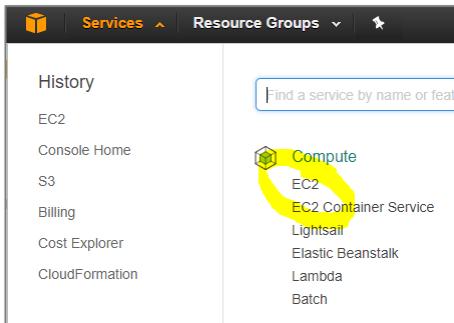
The Amazon RDS Instances page shows one instance named "backspace-intro-aws" in the "Status" column, which is circled in yellow and labeled "deleting".

DB instance	Engine	Status	CPU	Current activity
backspace-intro-aws	MySQL	deleting	1.17%	0 Conn

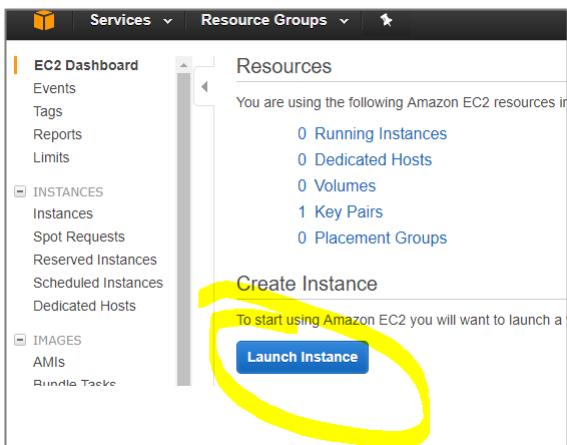
# Creating a Web Server with EC2

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

From the AWS console select 'EC2' from the Compute services.



Select 'Launch Instance'



Select the 'AWS Marketplace" and search for WordPress

Step 1: Choose an Amazon Machine Image (AMI)

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

Quick Start

My AMIs

**AWS Marketplace**

Community AMIs

Categories

Search: WordPress

aws marketplace

Find and buy software that runs in the AWS Cloud, software from true Marketplace products you are currently subscribed to by visiting [Your Subscriptions](#).

Select the Bitnami AMI

Step 1: Choose an Amazon Machine Image (AMI)

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

Quick Start

My AMIs

**AWS Marketplace**

Community AMIs

Categories

All Categories

Search: WordPress

**WordPress powered by Bitnami**

★★★★★ (18) | 4.8-0 on Ubuntu 14.04 | [Previous versions](#) | Sold by [Bitnami](#)

\$0.00/hr for software + AWS usage fees

Linux/Unix, Ubuntu 14.04 | 64-bit Amazon Machine Image (AMI) | Updated: 7/26/17

Bitnami WordPress is a pre-configured, ready to run image for running WordPress on Amazon EC2. WordPress is one of the world's most popular web publishing platforms for building ...

[More info](#)

1 to 25 of 128 Products

Select

Scroll to the bottom of the page and click 'Continue"

R4 Extra Large	\$0.00	\$0.266	\$0.266/hr
R4 Double Extra Large	\$0.00	\$0.532	\$0.532/hr
R4 Quadruple Extra Large	\$0.00	\$1.064	\$1.064/hr
R4 Eight Extra Large	\$0.00	\$2.128	\$2.128/hr
R4 16 Extra Large	\$0.00	\$4.256	\$4.256/hr
High I/O Extra Large	\$0.00	\$0.312	\$0.312/hr
High I/O Double Extra Large	\$0.00	\$0.624	\$0.624/hr
High I/O Quadruple Extra Large	\$0.00	\$1.248	\$1.248/hr
High I/O Eight Extra Large	\$0.00	\$2.496	\$2.496/hr
High I/O Sixteen Extra Large	\$0.00	\$4.992	\$4.992/hr
High I/O Large	\$0.00	\$0.156	\$0.156/hr

EBS General Purpose (SSD) volumes  
\$0.10 per GB-month of provisioned storage

You will not be charged until you launch this instance.

Cancel Continue

Choose the t2 Micro instance.  
Click 'Next: Configure Instance Details"

## Step 2: Choose an Instance Type

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

Note: The vendor recommends using a m3.medium instance (or larger) for the best experience with this product.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Instance Details](#)

Select enable for 'Auto-assign Public IP'

Click 'Review and Launch'

## Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	<input type="text" value="1"/>	<a href="#">Launch into Auto Scaling Group</a>
Purchasing option	<input type="checkbox"/> Request Spot instances	
Network	<input type="text" value="vpc-72d25a0b (default)"/>	<a href="#">Create new VPC</a>
Subnet	<input type="text" value="No preference (default subnet in any Availability Zone)"/>	<a href="#">Create new subnet</a>
Auto-assign Public IP	<input checked="" type="checkbox"/> Enable	
IAM role	<input type="text" value="None"/>	
Shutdown behavior	<input type="text" value="Stop"/>	
Enable termination protection	<input type="checkbox"/> Protect against accidental termination	
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring	

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

Click 'Launch'

### Step 7: Review Instance Launch

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

**⚠ Improve your instances' security.** Your security group, WordPress powered by Bitnami-4-8-0 on Ubuntu 14-04-AutogenByAWSMP-, is open to the world.  
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.  
You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers.  
[Edit security groups](#)

**AMI Details** [Edit AMI](#)

 **WordPress powered by Bitnami**  
<https://bitnami.com>

Free tier eligible | Root Device Type: ebs | Virtualization type: hvm

[Cancel](#) [Previous](#) **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"

**Select an existing key pair or create a new key pair**

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

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

Proceed without a key pair  
 I acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI.

[Cancel](#) **Launch Instances**

Wait for launch to initiate

**Launch Status**



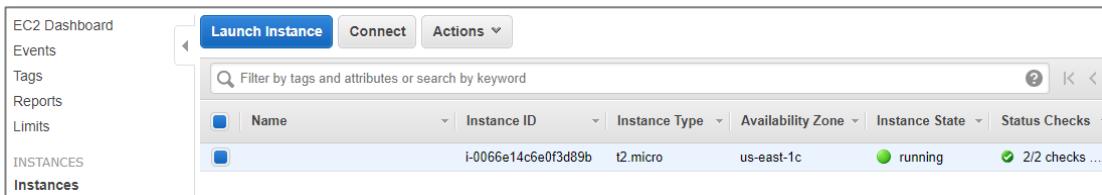
Initiating Instance Launches  
Please do not close your browser while this is loading

Creating security groups... Successful  
Authorizing inbound rules... Successful  
Subscribing to Product...

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

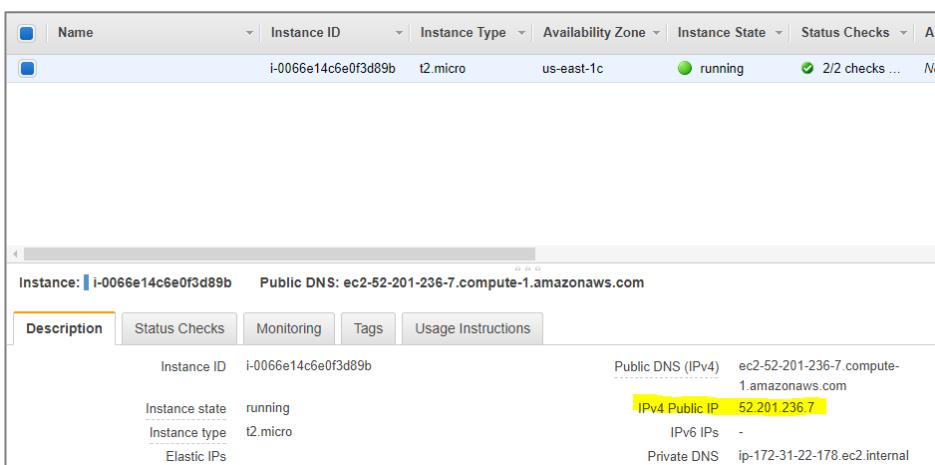


After a few minutes, the status of the instance will change to running.

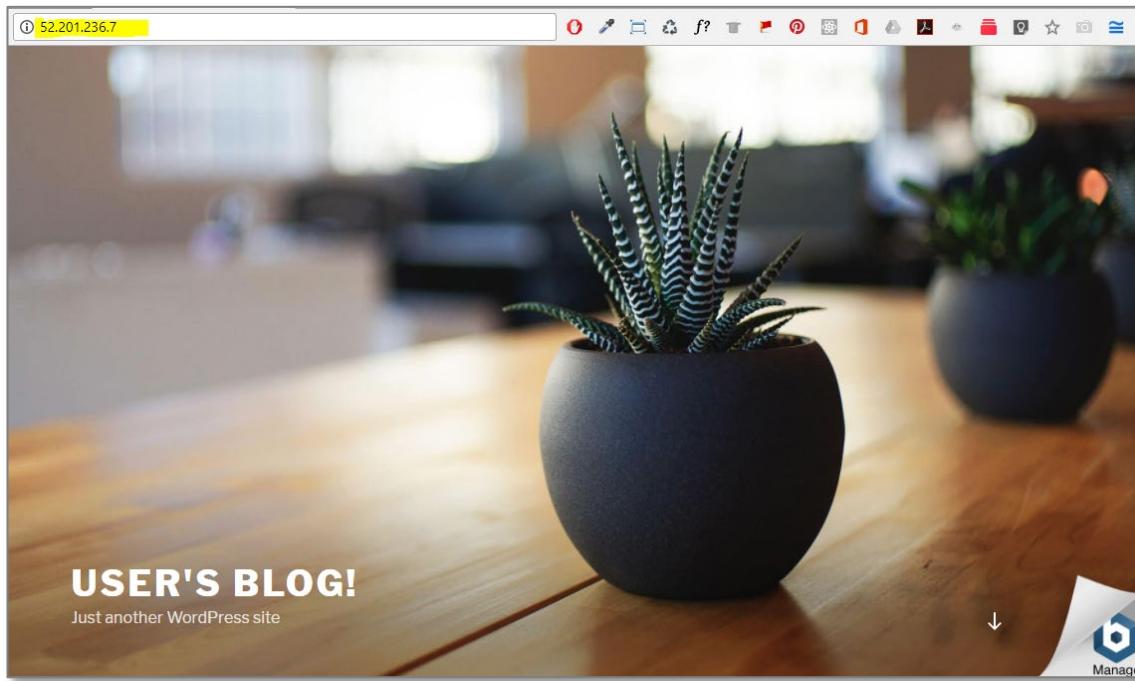


## Viewing your web server

Copy the public IP address of your web server.



Navigate to the IP address in your browser.



## Troubleshooting viewing your WordPress application

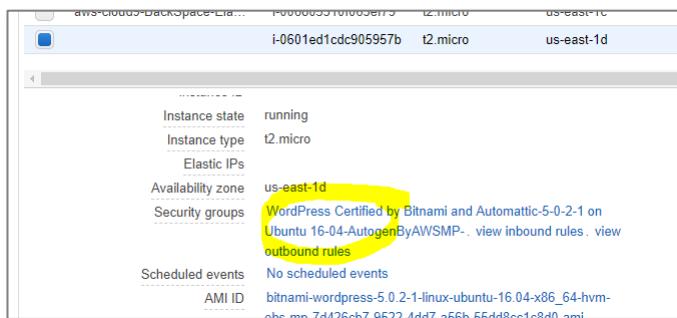
If you cannot view your website it probably hasn't finished the launch process completely.

If 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

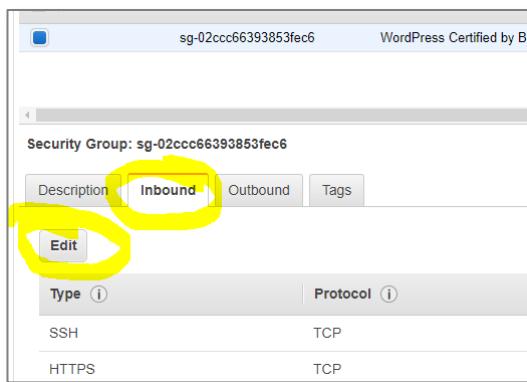
Security Groups associated with i-0601ed1cdc905957b			
Ports	Protocol	Source	WordPress Certified by Bitnami and Automatic-5-0-2-1 on Ubuntu 16-04-AutogenByAWSMP...
80	tcp	0.0.0.0/0	✓
22	tcp	0.0.0.0/0	✓
443	tcp	0.0.0.0/0	✓

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* tab

Click on *Edit*



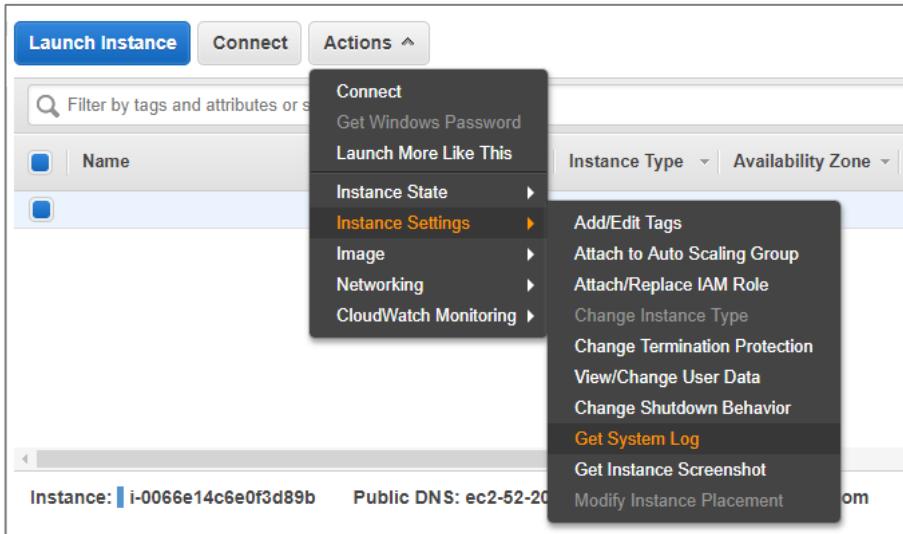
Add a rule for *HTTP* and *Anywhere*

Click *Save*



## Finding the Username and Password for your WordPress application

Go back to the EC2 console and select 'Instance Settings', 'Get System Log'. **Do not click on connect.**



Scroll down until you find the log entry for the application password and copy it.

System Log: i-0066e14c6e0f3d89b

```

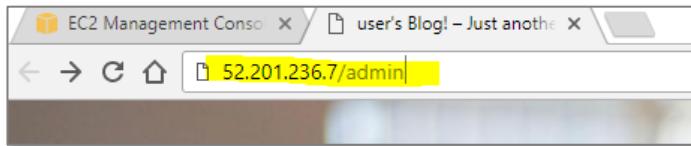
Setting up swapspace version 1, size = 649996 KiB
no label, UUID=847335f8-9b27-49bf-85e2-4642e4See7a0
micro
#####
#      Setting Bitnami application password to maIdmWlEy0QPS#
#####
[Mon Aug 14 05:29:19 UTC 2017] Regenerating keys for wordpress
[Mon Aug 14 05:29:24 UTC 2017] Regenerating keys for wordpress finished
[Mon Aug 14 05:29:24 UTC 2017] Finished regenerating keys
[Mon Aug 14 05:29:24 UTC 2017] Setting up password for mysql service
[Mon Aug 14 05:29:44 UTC 2017] Setting up password for mysql service finished
[Mon Aug 14 05:29:44 UTC 2017] Setting up password for wordpress application
[Mon Aug 14 05:29:58 UTC 2017] Setting up password for wordpress application finished
[Mon Aug 14 05:29:58 UTC 2017] Finished setting password
/opt/bitnami/mysql/scripts/ctl.sh : mysql (pid 2981) already running
/opt/bitnami/php/scripts/ctl.sh : php-fpm started
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd started at port 80

Enter new UNIX password: Retype new UNIX password: passwd: password updated successfully
Starting gonit daemon
* Stopping System V runlevel compatibilityD[74G[ OK ]
Cloud-init v. 0.7.5 running 'modules:final' at Mon, 14 Aug 2017 05:30:16 +0000. Up 80.11 s
ci-info: no authorized ssh keys fingerprints found for user bitnami.
ci-info: no authorized ssh keys fingerprints found for user bitnami.
ec2:
ec2: #####
ec2: -----BEGIN SSH HOST KEY FINGERPRINTS-----

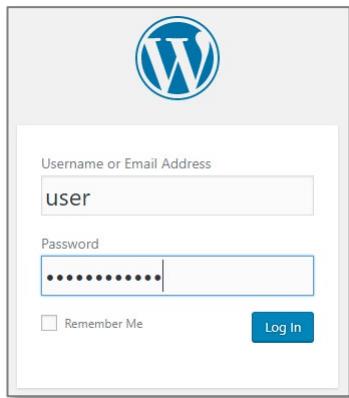
```

**Close**

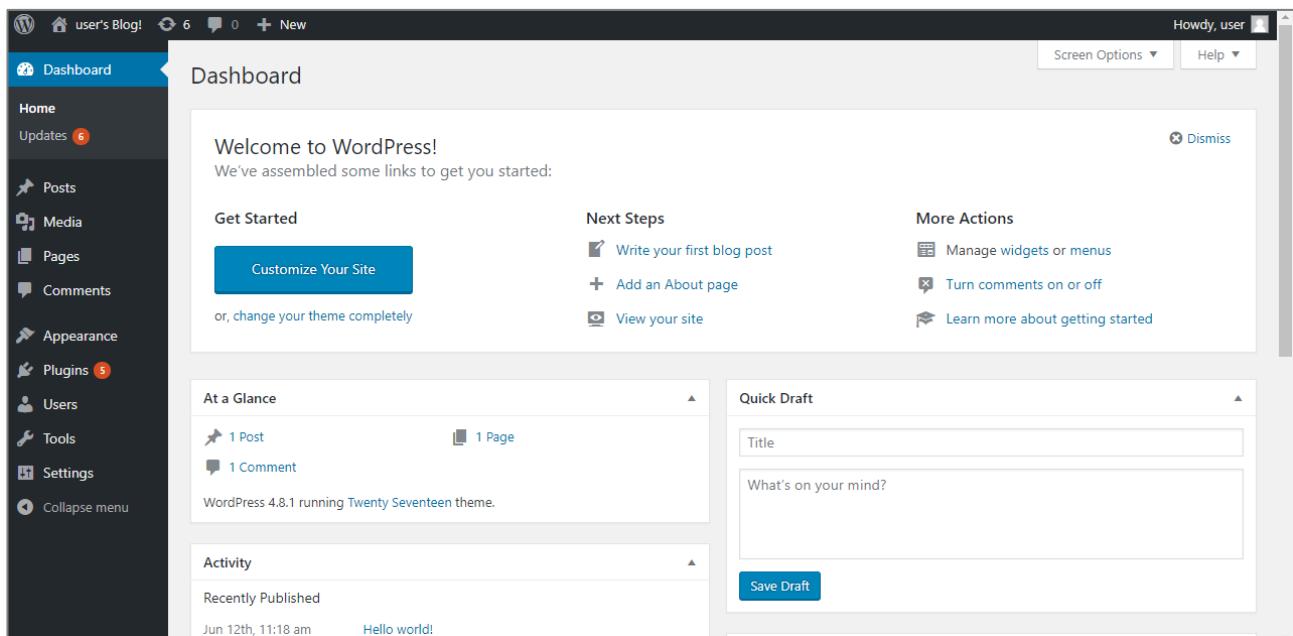
Go to the admin subdirectory of your website in your browser



Enter Username 'user' and paste in the password

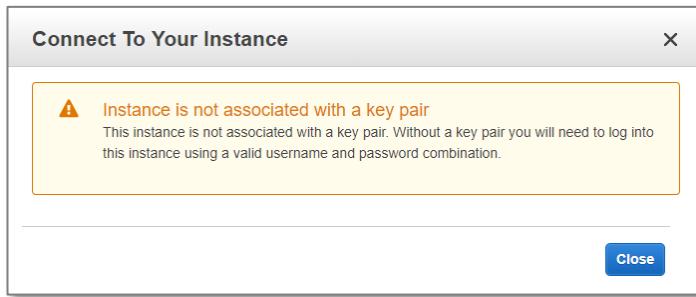


You will now be in the admin section of your WordPress application



## Troubleshooting logging in to the WordPress application

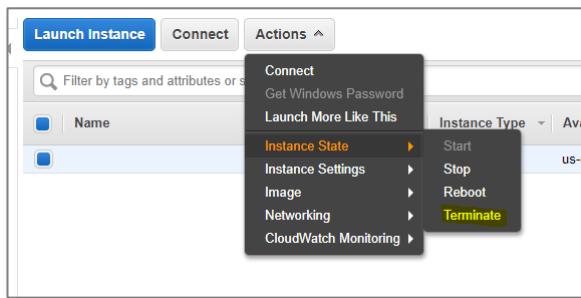
If you get the following message:



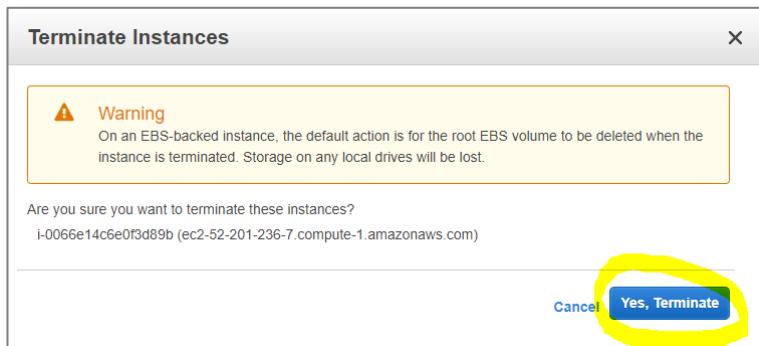
You have tried to connect to the Linux operating system by clicking on 'Connect'. Do not click on connect, select 'Actions – 'Instance settings" - 'Get System Log" as detailed previously.

## Clean up

Select 'Actions", 'Instance State", 'Terminate".



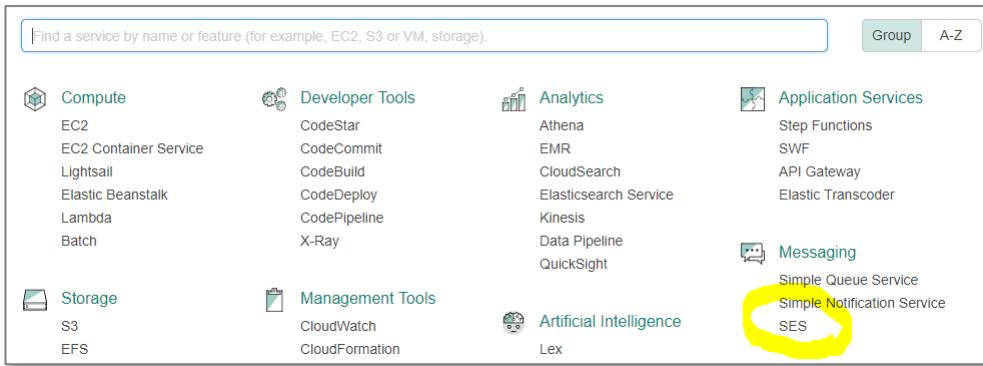
Make sure you terminate the instance so that you are not billed for it any more.



# ▶ Sending emails with Amazon SES

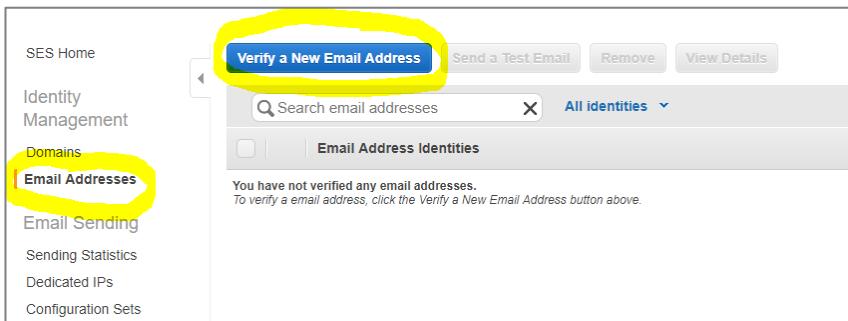
In this section, we will use the Simple Email Service to send an email.

From the AWS console select 'SES' from the Messaging services.

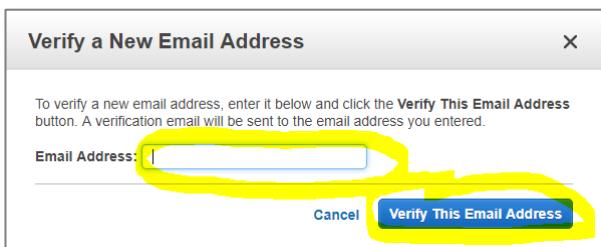


Click on 'Email addresses'

Click on 'Verify a New Email Address'



Enter your email address and click 'Verify this Email Address'



When you receive your verification email click on the supplied link.

You will then receive a success page

## Congratulations!

You have successfully verified an email address. You can now start sending email from this address.

**For new Amazon SES users**—If you have not yet applied for a sending limit increase, then you are still in the [sandbox environment](#), and you can only send email to addresses that have been verified. To verify a new email address or domain, see the [Identity Management](#) section of the [Amazon SES console](#).

**For new Amazon Pinpoint users**—If you have not yet applied for a sending limit increase, then you are still in the [sandbox environment](#), and you can only send email to addresses that have been verified. To verify a new email address or domain, see the [Settings > Channels](#) page on the [Amazon Pinpoint console](#).

If you have already been approved for a sending limit increase, then you can start sending email to non-verified addresses.

Thank you for using Amazon Web Services!

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

The screenshot shows the 'Verify a New Email Address' page. At the top, there are buttons for 'Verify a New Email Address', 'Send a Test Email' (which is highlighted with a yellow circle), 'Remove', and 'View Details'. Below this is a search bar and a dropdown menu set to 'All identities'. A table lists one email identity, which is highlighted with a yellow circle. The 'Status' column shows 'verified' in green. The 'Actions' column contains a refresh icon and a question mark icon, also highlighted with a yellow circle.

Click on the email address and select 'Send a test email'

This screenshot is similar to the previous one, but the 'Send a Test Email' button is explicitly highlighted with a large yellow circle. The rest of the interface, including the table and other buttons, remains the same.

Enter the same email address for from and to.

Fill out the email information and click 'Send test email'

The screenshot shows the 'Send Test Email' dialog box. It includes fields for 'From\*', 'To\*', 'Subject\*', and 'Body'. The 'From\*' field contains a redacted email address, and the 'Subject\*' field contains 'This is an SES test'. The 'Body' field contains 'This is an SES test'. At the bottom, there are 'Cancel' and 'Send Test Email' buttons, with the latter being highlighted with a yellow circle.

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 'Sending Statistic"

Click on 'Request a Sending Limit Increase"

The screenshot shows the 'Your Amazon SES Sending Limits' page. On the left, there's a sidebar with links: SES Home, Identity Management, Domains, Email Addresses, Email Sending, **Sending Statistics**, Dedicated IPs, and Configuration Sets. The main content area has a message box: "Your Amazon SES account has 'sandbox' access in region US East (N. Virginia). With sandbox access you can only send email to the Amazon SES mailbox simulator and to email addresses or domains that you have verified. To be moved out of the sandbox, please request a sending limit increase. [Learn more](#)". Below this, another message says: "Can't find your existing account settings? Your account may be set up in a different AWS region. Try switching regions in the upper right corner of the console." At the bottom of this section is a blue button with white text: "Request a Sending Limit Increase". This button is highlighted with a large yellow circle. Below the message box, the title "Your Amazon SES Sending Limits" is displayed with a dropdown arrow, and a note: "Below are the latest statistics and metrics related to your Amazon SES Usage." A refresh icon is also present.

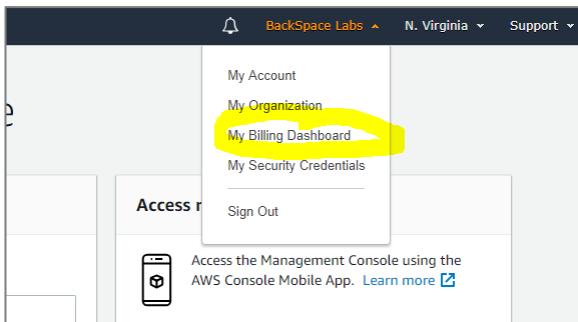


# Creating a Billing Alert with CloudWatch and SNS

In this section, we will create a CloudWatch billing alert that will send an email through the Simple Notification Service whenever our estimated monthly bill exceeds a certain level.

## Enabling Billing Alerts

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



Select *Preferences*

Check *Receive Free Tier Usage Alerts*

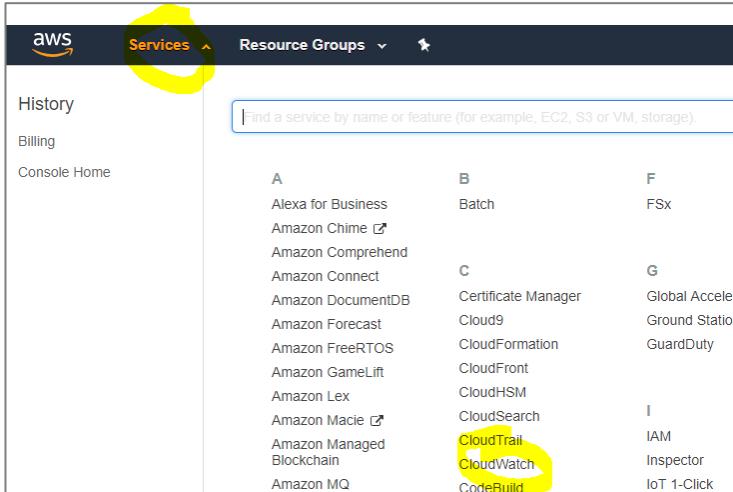
Check *Receive Billing Alerts*

Dashboard	Preferences
Bills	<input type="checkbox"/> <b>Receive PDF Invoice By Email</b> Turn on this feature to receive a PDF version of your invoice by email. Invoices are generally available within the first three days of the month.
Cost Explorer	<input type="checkbox"/> <b>Disable credit sharing</b> When credit sharing is disabled, credits will only be applied to the credit owner's account, and will not be shared across accounts in the same billing family. Download credit sharing preference history.
Budgets	<input type="checkbox"/> <b>RI discount sharing</b>
Reports	<input type="checkbox"/> <b>Cost Management Preferences</b>
Cost Allocation Tags	<input checked="" type="checkbox"/> <b>Receive Free Tier Usage Alerts</b> Turn on this feature to receive email alerts when your AWS service usage is approaching, or has exceeded, the AWS Free Tier usage limits. If you wish to receive these alerts at an email address that is not the primary email address associated with this account, please specify the email address below.
Payment Methods	Email Address: [redacted]
Payment History	<input checked="" type="checkbox"/> <b>Receive Billing Alerts</b> Turn on this feature to monitor your AWS usage charges and recurring fees automatically, making it easier to track and manage your spending on AWS. You can set up billing alerts to receive email notifications when your charges reach a specified threshold. Once enabled, this preference cannot be disabled. <a href="#">Manage Billing Alerts</a> or try the new budgets feature!
Consolidated Billing	
<b>Preferences</b>	
Credits	
Tax Settings	

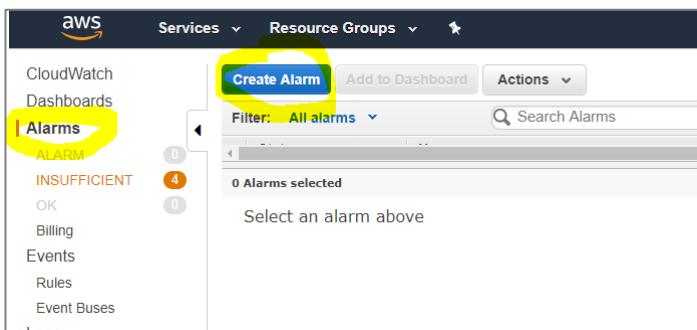
Click *Save preferences*

## Creating a CloudWatch Alarm

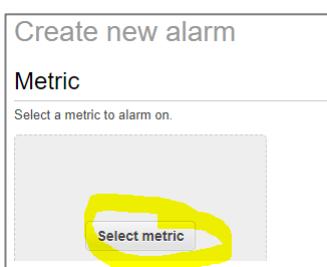
Click the Services menu and select 'CloudWatch' from 'Management Tools'



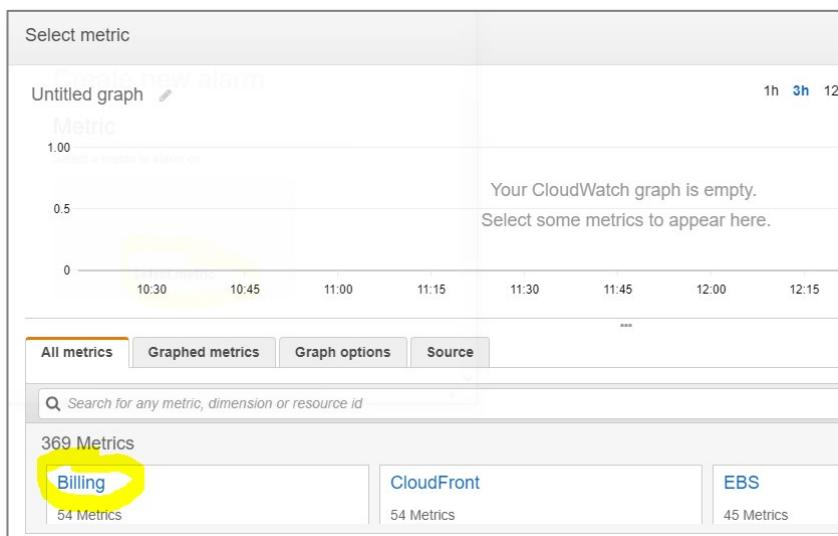
Click on 'Alarms', 'Create Alarm'



Click *Select metric*

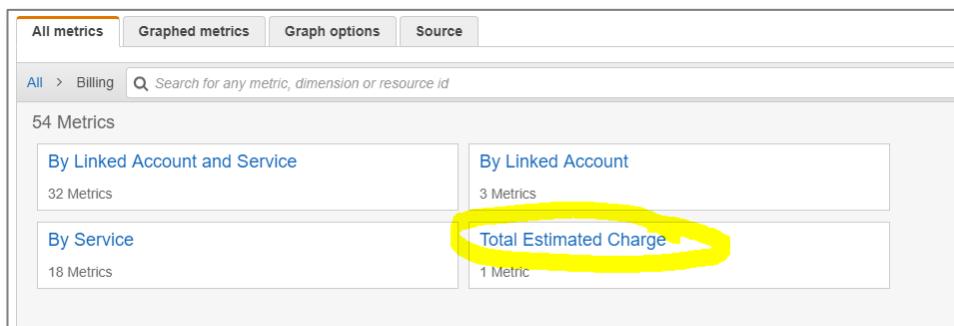


Click *Billing*



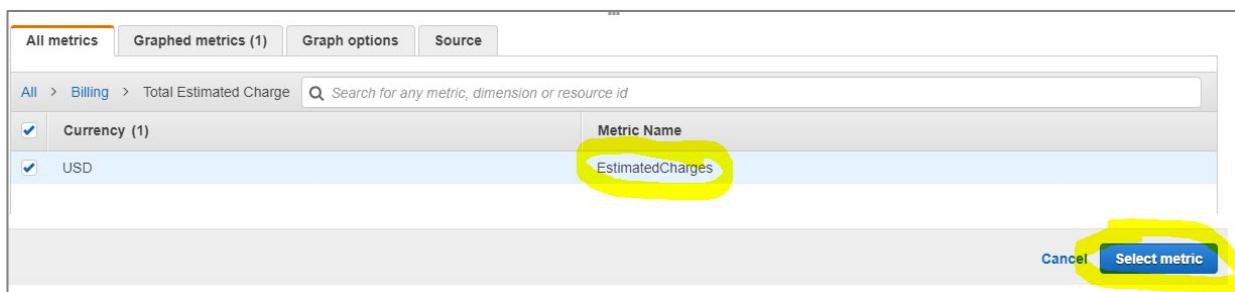
If you do not see any metrics it is most probably you are not operating in the US-East1 (N. Virginia) region. Please ensure you always operate from the US-East region during the course.

Select *Total Estimated Charge* from the billing metrics.



Select *EstimatedCharges* metric

Click *Select metric*



Set the alarm threshold to not exceed \$10

Select *anonymous-mail* for notification

Click Create Alarm.

### Billing alarm

You can create a billing alarm to receive e-mail alerts when your AWS charges exceed a threshold you choose. Simply:

1. Enter a spending threshold
2. Provide an email address
3. Check your inbox for a confirmation email and click the link provided

**When my total AWS charges for the month**

exceed:  USD

send a notification to:  New list

Reminder: for each address you add, you will receive a confirmation email. Click the link provided in the message to confirm that AWS may deliver notifications to that address.

**Additional settings**

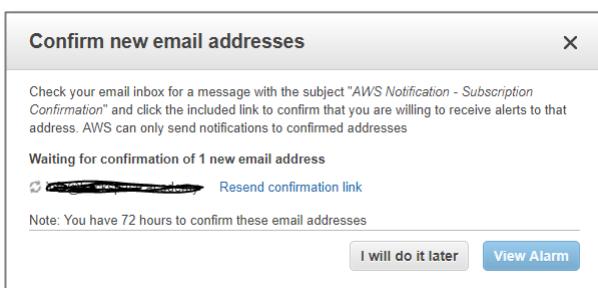
Provide additional configuration for your alarm.

Treat missing data as:  ?

showing simple options | [show advanced](#)

[Cancel](#) [Create Alarm](#)

If you haven't already confirmed your email a confirmation email will be sent to you.



Go back to the CloudWatch console and refresh the screen.

The Alarm State will be INSUFFICIENT\_DATA until enough data has been collected by CloudWatch

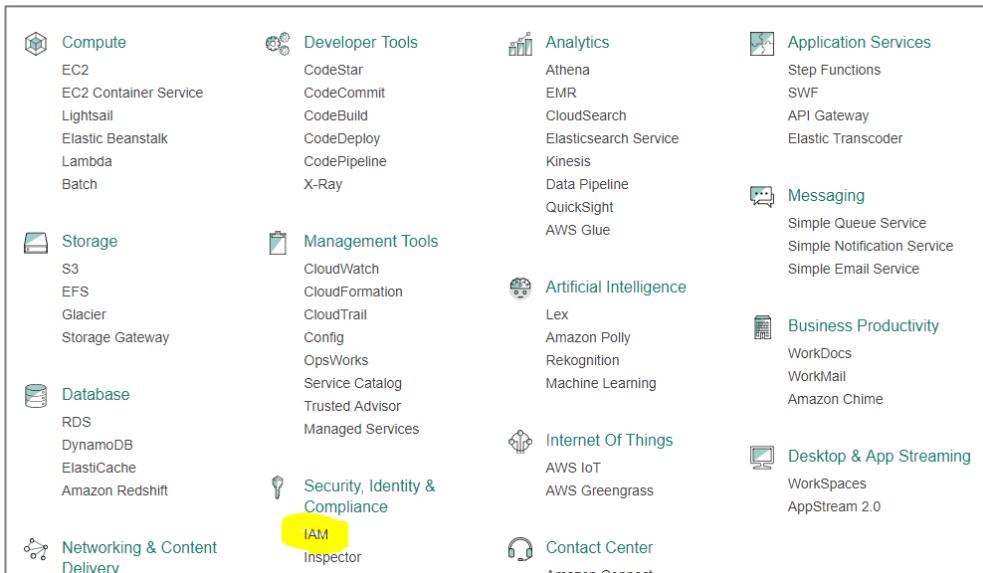
The screenshot shows the CloudWatch Alarms dashboard. The left sidebar lists "CloudWatch", "Dashboards", "Alarms", "ALARM" (0), and "INSUFFICIENT" (5). The main area shows a table of alarms. One alarm is listed: "INSUFFICIENT\_DATA" with the name "BillingAlarm". The "State" column shows "INSUFFICIENT\_DATA" and the "Name" column shows "BillingAlarm". A tooltip for the alarm indicates "EstimatedCharges > 10 for 1 datapoints within 6 hours". The top navigation bar includes "Create Alarm", "Add to Dashboard", and "Actions". The search bar shows "Search Alarms" and a filter set to "All alarms".

# 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.

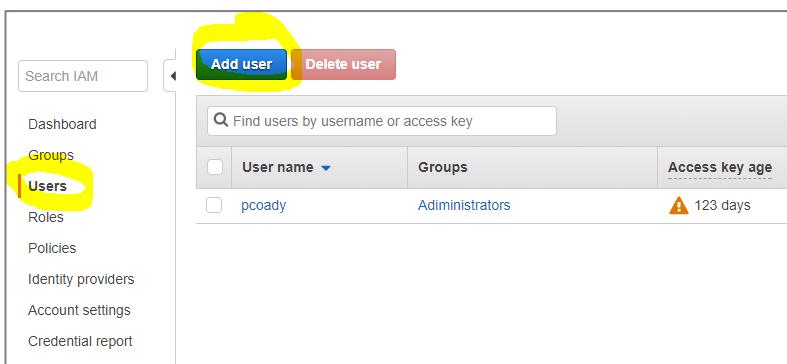
From the AWS console click 'Services'

Select 'IAM' from the Security, Identity & Compliance services.



Select 'Users'

Click 'Add user'



Give the user a name

Add user

1 Details    2 Permissions    3 Review    4 Complete

### Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*   

[+ Add another user](#)

Check 'Programmatic access'

Check 'AWS Management Console access'

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type\*  **Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

**AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password\*  Autogenerated password  
 Custom password

Require password reset   
User must create a new password at next sign-in  
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

\* Required Cancel Next: Permissions

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

Click 'Next Review"

Add user

1    2  3    4

### Set permissions for test-user

Add user to group Copy permissions from existing user Attach existing policies directly

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Add user to group

[Create group](#) Refresh



Click 'Create user'

Add user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	test-user
AWS access type	Programmatic access and AWS Management Console access
Console password type	Autogenerated
Require password reset	Yes

Permissions summary

The user shown above will be added to the following groups.

Type	Name
Managed policy	<a href="#">IAMUserChangePassword</a>

Cancel Previous **Create user**

Download the csv file containing the user credentials (access key and secret access key) to a safe location.

You will need this for access using the Command Line Interface (CLI) later in the course.

Add user

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://backspace-academy.signin.aws.amazon.com/console>

**Download .csv**

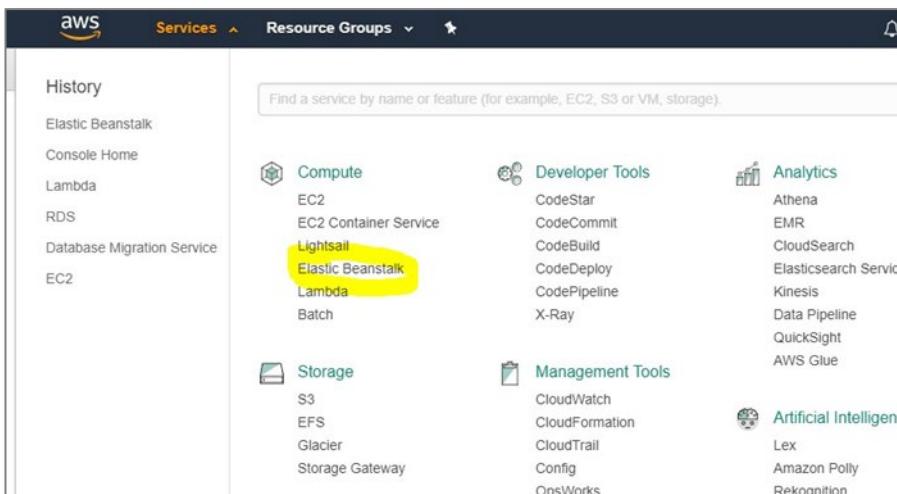
User	Access key ID	Secret access key	Password	Email login instructions
test-user	AKIAJZGZ6UMOZT3U5V6Q	***** Show	***** Show	<a href="#">Send email</a>

Close

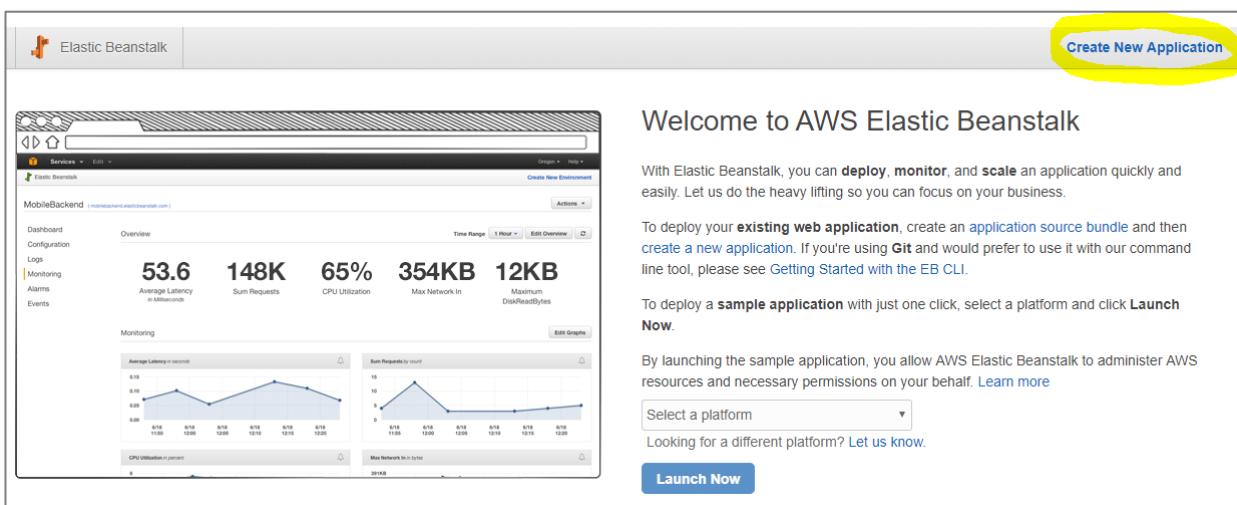
# Creating a Highly Available Architecture with Elastic Beanstalk

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

Click on the services menu and select *Elastic Beanstalk*

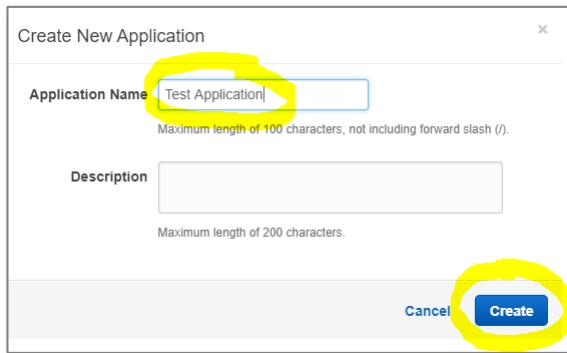


Click 'Create New Application'



Give your application a name *Test Application*.

Click "Create"



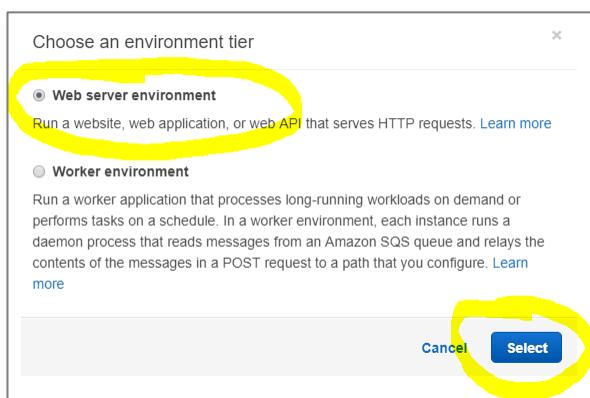
Your application will now be created.

Select “Actions” - “Create Environment”



Select “Web server environment”

Click “Select”



Leave Environment information with default values

 **Create a new environment**

Launch an environment with a sample application or your own code. By creating an environment, you allow AWS Elastic Beanstalk to manage AWS resources and permissions on your behalf. [Learn more](#)

**Environment information**

Choose the name, subdomain, and description for your environment. These cannot be changed later.

**Application name** Test Application

**Environment name**

**Domain** Leave blank for autogenerated value  [Check availability](#)

**Description**

Select **Node.Js** as the platform

Select **Sample Application** for Application Code

Click **Configure More Options**

Base configuration

**Tier** Web Server ([Choose tier](#))

**Platform**  Preconfigured platform  
Platforms published and maintained by AWS Elastic Beanstalk.  
 (highlighted)

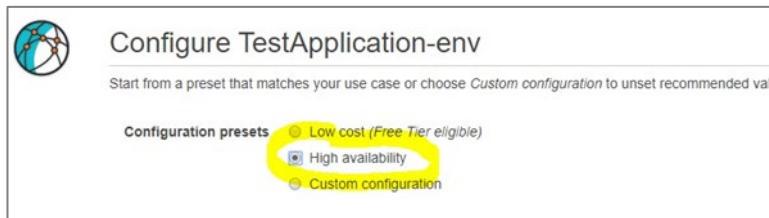
Custom platform NEW  
Platforms created and owned by you. [Learn more](#)

**Application code**  Sample application  
Get started right away with sample code.  
 Existing version  
Application versions that you have uploaded for Test Application.

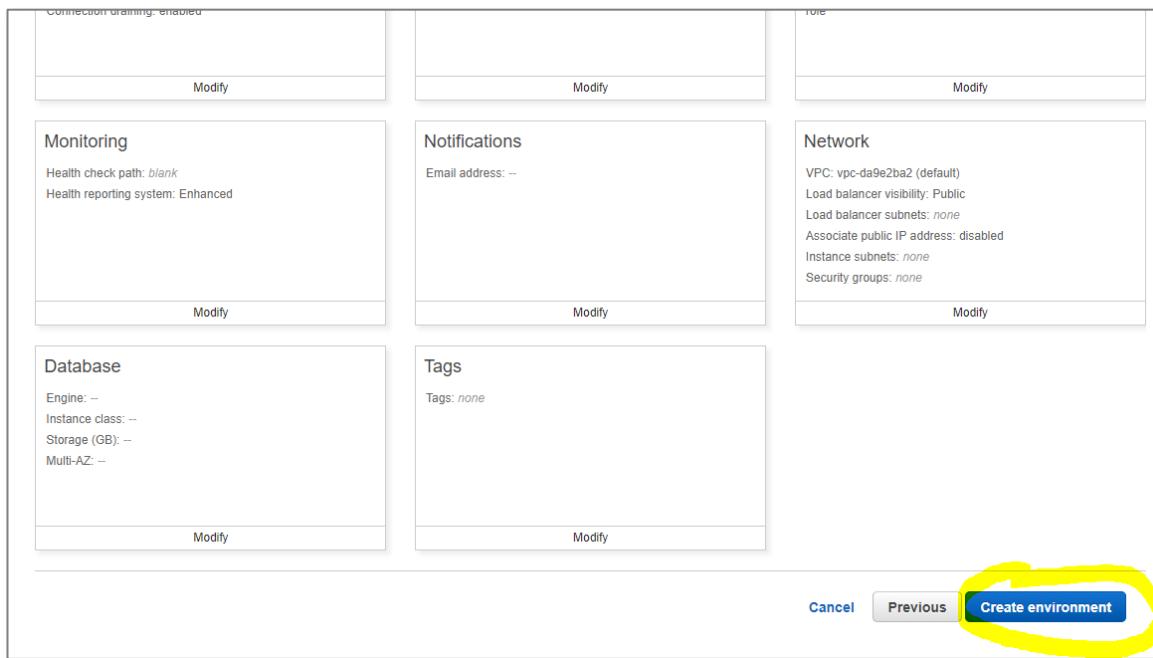
Upload your code  
Upload a source bundle from your computer or copy one from Amazon S3.

[Cancel](#) [Configure more options](#) (highlighted) [Create environment](#)

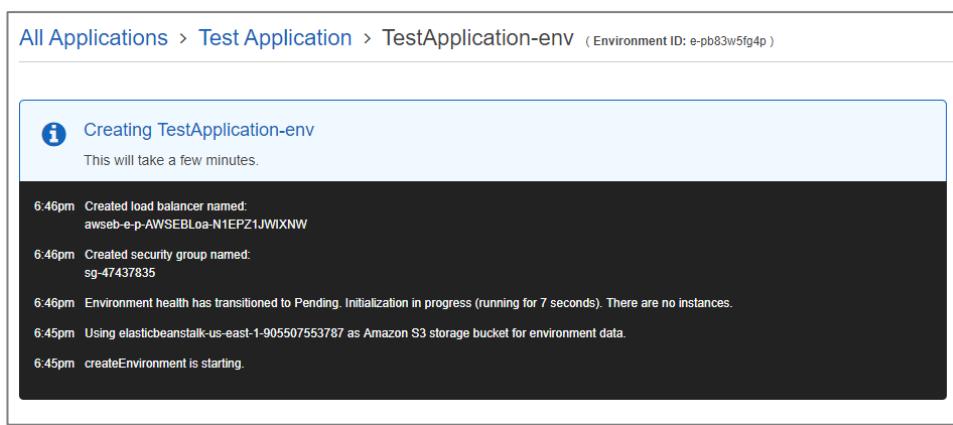
Select **High availability**



Scroll down and click *Create environment*



Your environment will start being created



After some time, your environment will be created.

Click on the website url

All Applications > Test Application > TestApplication-env (Environment ID: e-pb83w5fg4p, URL: [TestApplication-env.mxafx3y3j9.us-east-1.elasticbeanstalk.com](https://TestApplication-env.mxafx3y3j9.us-east-1.elasticbeanstalk.com)) Actions ▾

Dashboard Overview Refresh

Configuration

Logs Health **Ok** Causes

Monitoring

Alarms

Managed Updates

Events

Tags

Running Version Sample Application Upload and Deploy

node Configuration  
64bit Amazon Linux 2017.03  
v4.3.0 running Node.js Change

You will now see the Sample Application

Congratulations

Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud

What's Next?

- [AWS Elastic Beanstalk overview](#)
- [AWS Elastic Beanstalk concepts](#)
- [Deploy an Express Application to AWS Elastic Beanstalk](#)
- [Deploy an Express Application with Amazon ElastiCache to AWS Elastic Beanstalk](#)
- [Deploy a Geddy Application with Amazon ElastiCache to AWS Elastic Beanstalk](#)
- [Customizing and Configuring a Node.js Container](#)
- [Working with Logs](#)

## Clean Up

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

Navigate back to the Test Application

All Applications > **Test Application** > TestApplication-env (Environment ID: e-pb83w5fg4p, URL: [TestApplication-env.mxafrx3y3j9.us-east-1.elasticbeanstalk.com](http://TestApplication-env.mxafrx3y3j9.us-east-1.elasticbeanstalk.com)) Actions ▾

**Overview**

Health: **OK** Causes: [Upload and Deploy](#)

Running Version: Sample Application

Configuration: node v4.3.0 running Node.js

Recent Events: [Show All](#)

- Dashboard
- Logs
- Health
- Monitoring
- Alarms
- Managed Updates
- Events
- Tags

Click Actions

Select Delete Application

All Applications > Test Application

**Actions** ▾

- Create environment
- Restore terminated environment
- Swap environment URLs
- Delete application**

Environments: TestApplication-env

Application versions: Environment tier: Web Server  
Platform: 64bit Amazon Linux 2017.03 v4.3.0 running Node.js  
Running versions: Sample Application  
Last modified: 2017-11-05 18:50:40 UTC+1100  
URL: [TestApplication-env.mxafrx3y3j9.us-east-1.elasticbeanstalk.com](http://TestApplication-env.mxafrx3y3j9.us-east-1.elasticbeanstalk.com)

Application versions: Environment tier: Web Server  
Platform: 64bit Amazon Linux 2017.03 v4.3.0 running Node.js  
Running versions: Sample Application  
Last modified: 2017-11-05 18:50:40 UTC+1100  
URL: [TestApplication-env.mxafrx3y3j9.us-east-1.elasticbeanstalk.com](http://TestApplication-env.mxafrx3y3j9.us-east-1.elasticbeanstalk.com)

Saved configurations:

Click "Delete"

Delete Application

Are you sure you want to delete the application: **Test Application**?

[Cancel](#) **Delete**

Click on the environment

All Applications

Test Application

TestApplication-env

Environment tier: Web Server  
Platform: 64bit Amazon Linux 2017.03 v4.3.0 running Node.js  
Running versions: Sample Application  
Last modified: 2017-11-05 18:55:47 UTC+1100  
URL: [TestApplication-env.mxafrx3y3j9.us-east-1.elasticbeanstalk.com](http://TestApplication-env.mxafrx3y3j9.us-east-1.elasticbeanstalk.com)

You will now see your environment is being terminated.

All Applications > [Test Application](#) > TestApplication-env ( Environment ID: e-pb83w5fg4p, URL: [TestApplication-env.mxafx3y3j9.us-east-1.elasticbeanstalk.com](#) ) Actions ▾

Dashboard Configuration Logs Health Monitoring Alarms Managed Updates Events Tags

Overview Refresh

Elastic Beanstalk is terminating your environment. View Events

Health: Ok Causes

Running Version: Sample Application Upload and Deploy

node Configuration: 64bit Amazon Linux 2017.03 v4.3.0 running Node.js Change

The screenshot shows the AWS Elastic Beanstalk console for the 'Test Application' environment. A yellow oval highlights a message at the top stating 'Elastic Beanstalk is terminating your environment.' with a 'View Events' link. Below this, the 'Health' status is shown as 'Ok' with a 'Causes' button. The 'Running Version' is listed as 'Sample Application' with a 'Upload and Deploy' button. On the right, there's a 'node' logo and configuration details: 'Configuration' section showing '64bit Amazon Linux 2017.03 v4.3.0 running Node.js' and a 'Change' button.