



lab title

**Introduction to AWS  
V1.18**



Course title

**BackSpace Academy  
AWS Certified Associate**



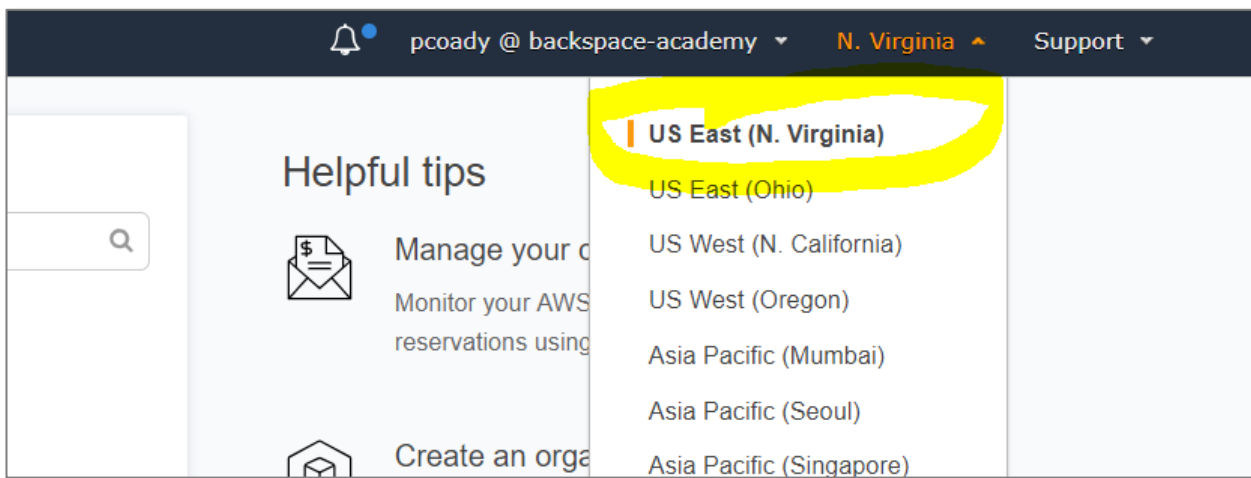
# Table of Contents

## Contents

<b>Table of Contents</b> .....	1
<b>About the Lab</b> .....	2
<b>Checking your AWS Usage and Monthly Bill</b> .....	3
<b>Creating an S3 Bucket and Uploading Files</b> .....	4
Uploading Files to your Bucket .....	7
Downloading files from your bucket .....	9
Clean Up .....	10
<b>Creating a SQL Database with RDS</b> .....	13
Connecting to your RDS Instance .....	18
Troubleshooting Connection Issues .....	21
Connecting to your RDS Instance using the Command Line .....	24
Clean Up .....	25
<b>Creating a Web Server with EC2</b> .....	27
Viewing your web server .....	31
Finding the Username and Password for your WordPress application .....	31
Clean up .....	34
<b>Sending emails with Amazon SES</b> .....	35
Requesting full access to SES .....	37
<b>Creating a Billing Alert with CloudWatch and SNS</b> .....	38
Enabling Billing Alerts .....	38
Creating a CloudWatch Alarm .....	38
<b>Creating an IAM User</b> .....	43
<b>Creating a Highly Available Architecture with Elastic Beanstalk</b> .....	46
Clean Up .....	50

## 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.**



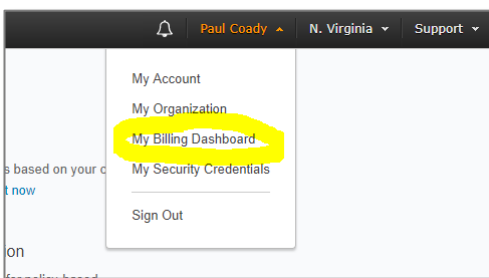
These lab notes are to support the hands on instructional videos of the Introduction to AWS section of the AWS Certified Associate Course.

**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 latest version with any updates or corrections.**

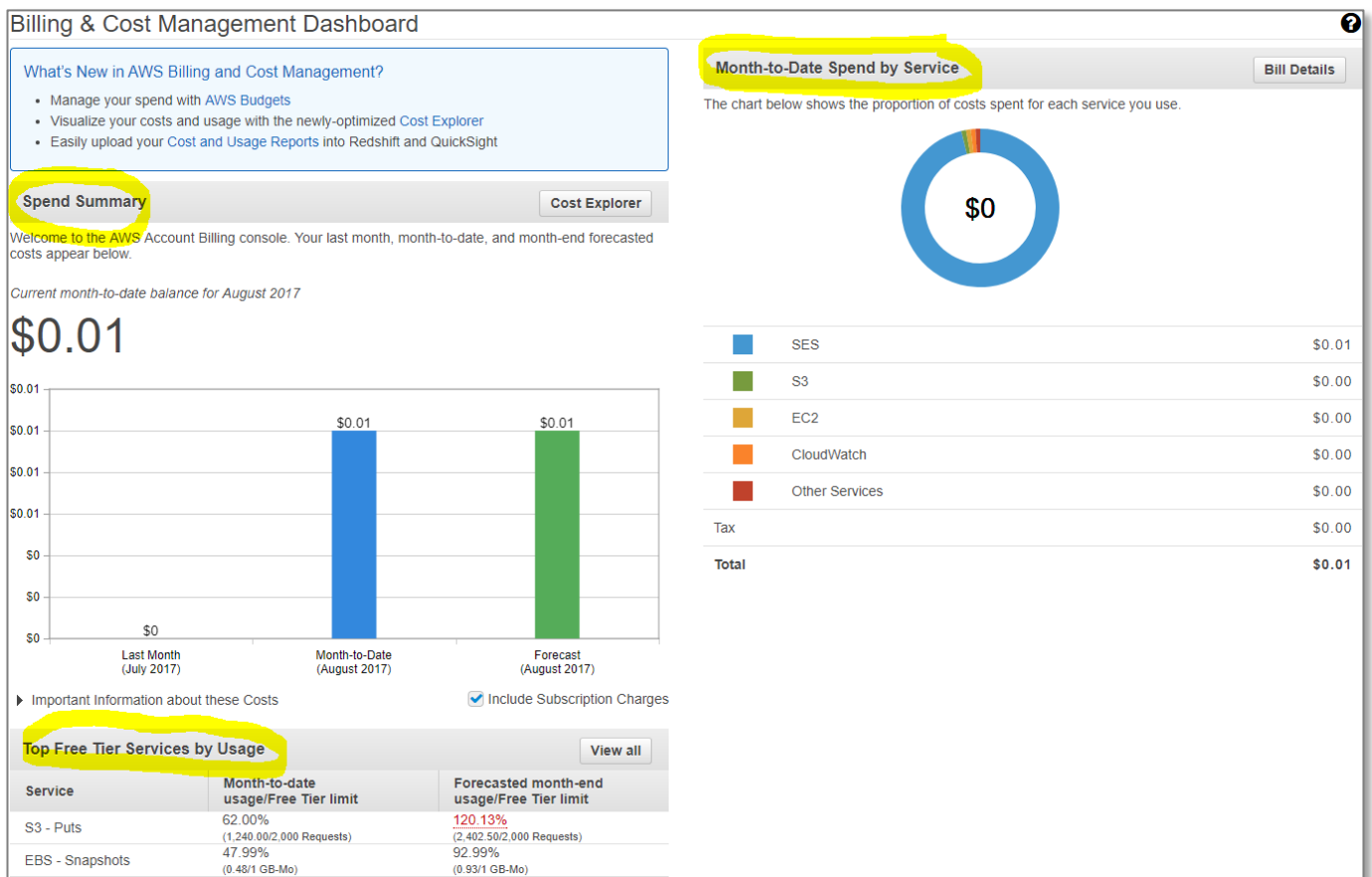
# Checking your AWS Usage and Monthly Bill

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.



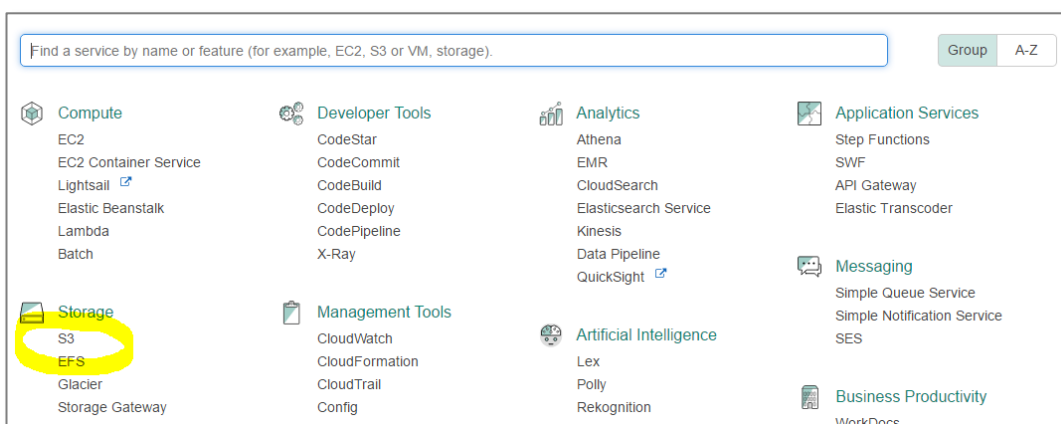
You will now see your total spend summary, spend by service and forecast spend.



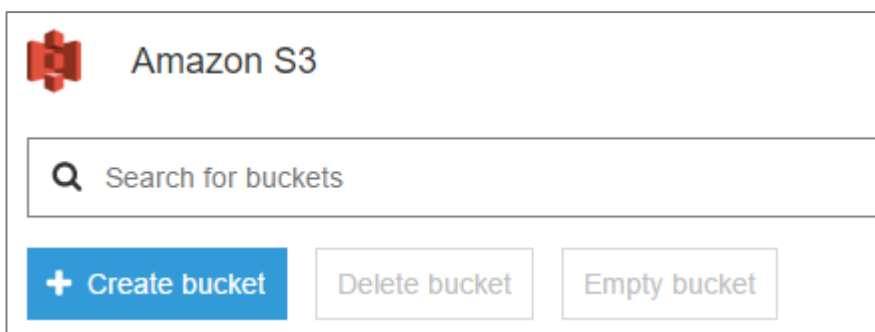
# Creating an S3 Bucket and Uploading Files

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"

The screenshot shows the 'Create bucket' wizard in the AWS Management Console. The title bar is blue with a close button (X) on the right. Below the title bar is a progress bar with four steps: 1. Name and region (active), 2. Set properties, 3. Set permissions, and 4. Review. The main content area is dark blue. Under the heading 'Name and region', there is a 'Bucket name' field with a help icon (i) and a text input containing 'backspace-intro-aws'. Below this is a 'Region' dropdown menu showing 'US East (N. Virginia)'. Further down is a section 'Copy settings from an existing bucket' with a 'Select bucket (optional)' dropdown showing '2 Buckets'. At the bottom, there are three buttons: 'Create' (disabled), 'Cancel', and 'Next' (active).

Leave as is and click “Next”

The screenshot shows the 'Create bucket' wizard in the AWS Management Console, Step 2: Set properties. The progress bar now shows Step 1 as completed (with a checkmark) and Step 2 as active. The main content area is dark blue and contains three white cards. The 'Versioning' card has the title 'Versioning', a description 'Keep multiple versions of an object in the same bucket.', a 'Learn more' link, and a 'Disabled' toggle. The 'Logging' card has the title 'Logging', a description 'Set up access log records that provide details about access requests.', a 'Learn more' link, and a 'Disabled' toggle. The 'Tags' card has the title 'Tags', a description 'Use tags to track your cost against projects or other criteria.', a 'Learn more' link, and a '0 Tags' toggle. At the bottom, there are two buttons: 'Previous' (disabled) and 'Next' (active).

Leave as is and click “Next”

The screenshot shows the 'Create bucket' wizard in the AWS Management Console, specifically Step 3: Set permissions. The progress bar at the top indicates four steps: 1. Name and region, 2. Set properties, 3. Set permissions (current), and 4. Review. The main content area is divided into three sections: 'Manage users', 'Manage public permissions', and 'Manage system permissions'. The 'Manage users' section contains a table with columns for 'User ID', 'Objects', and 'Object permissions'. The 'Manage public permissions' section has a dropdown menu set to 'Do not grant public read access to this bucket (Recommended)'. The 'Manage system permissions' section has a dropdown menu set to 'Do not grant Amazon S3 Log Delivery group write access to this bucket'. At the bottom right, there are 'Previous' and 'Next' buttons.

User ID	Objects	Object permissions
pcoady(Owner)	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write

Click "Create Bucket"

The screenshot shows the 'Create bucket' wizard in the AWS Management Console, specifically Step 4: Review. The progress bar at the top indicates four steps: 1. Name and region, 2. Set properties, 3. Set permissions, and 4. Review (current). The main content area is divided into three sections: 'Name and region', 'Properties', and 'Permissions'. The 'Name and region' section shows 'Bucket name' as 'backspace-intro-aws' and 'Region' as 'US East (N. Virginia)'. The 'Properties' section shows 'Versioning' as 'Disabled', 'Logging' as 'Disabled', and 'Tagging' as '0 Tags'. The 'Permissions' section shows 'Users' as '1', 'Public permissions' as 'Disabled', and 'System permissions' as 'Disabled'. At the bottom right, there are 'Previous' and 'Create bucket' buttons.

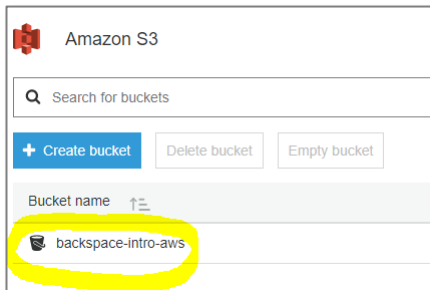
Name and region	
<b>Bucket name</b>	backspace-intro-aws
<b>Region</b>	US East (N. Virginia)

Properties	
<b>Versioning</b>	Disabled
<b>Logging</b>	Disabled
<b>Tagging</b>	0 Tags

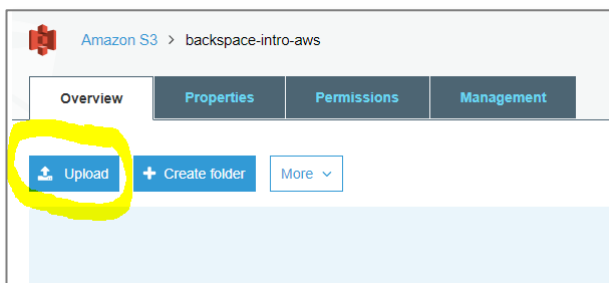
Permissions	
<b>Users</b>	1
<b>Public permissions</b>	Disabled
<b>System permissions</b>	Disabled

## Uploading Files to your Bucket

Click on the link to the bucket

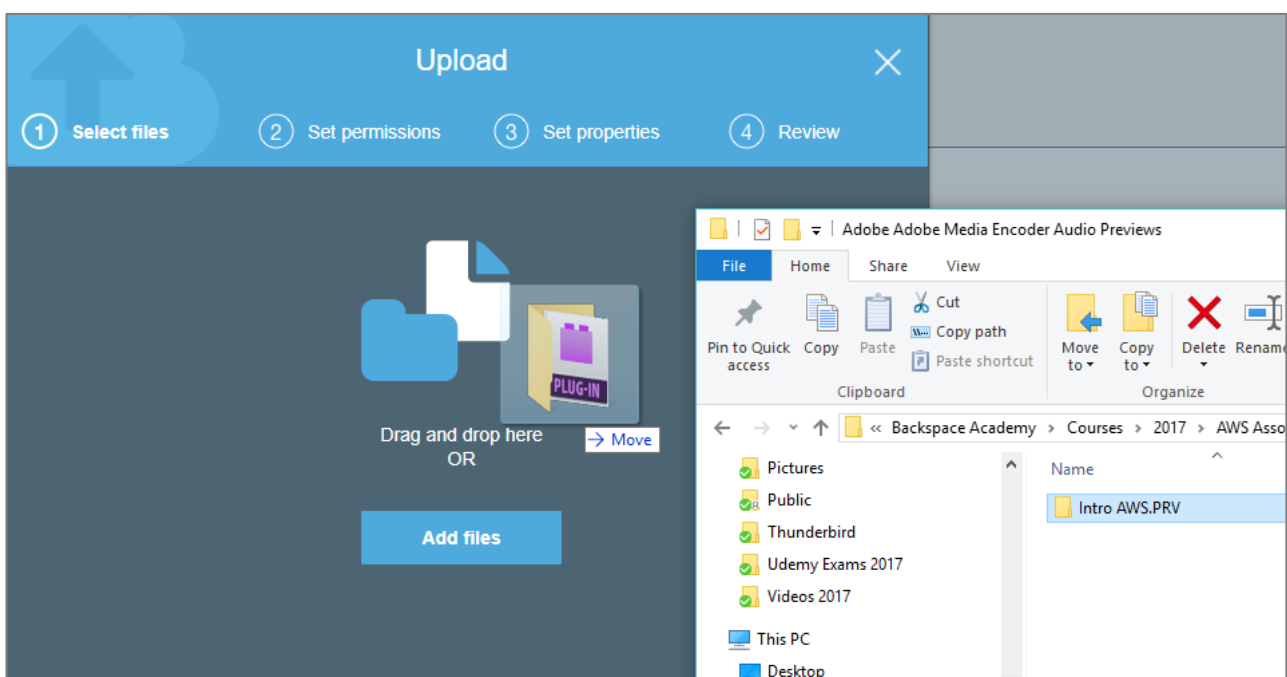


Select "Upload"



Drag a folder with files onto the form.

Click Next



Leave as is and click "Next"



**Upload**

1 Select files 2 **Set permissions** 3 Set properties 4 Review

1 Files Size: 105.0 MB Target path: backspace-intro-aws

**Manage users**

User ID	Objects	Object permissions
pcoady(Owner)	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write

**Manage public permissions**

Do not grant public read access to this object(s) (Recommended)

Upload Previous Next

Leave as is and click “Next”

**Upload**

1 Select files 2 Set permissions 3 **Set properties** 4 Review

1 Files Size: 105.0 MB Target path: backspace-intro-aws

**Storage class**  
Choose one depending on your use case scenario and performance access requirements.

☒ Standard ☐ Standard-IA ☐ Reduced redundancy

**Encryption**  
Protect data at rest by using Amazon S3 master-key or by using AWS KMS master-key.

☒ None ☐ Amazon S3 master-key ☐ AWS KMS master-key

**Metadata**  
Metadata is a set of name-value pairs. You cannot modify object metadata after it is uploaded.

Header	Value
--------	-------

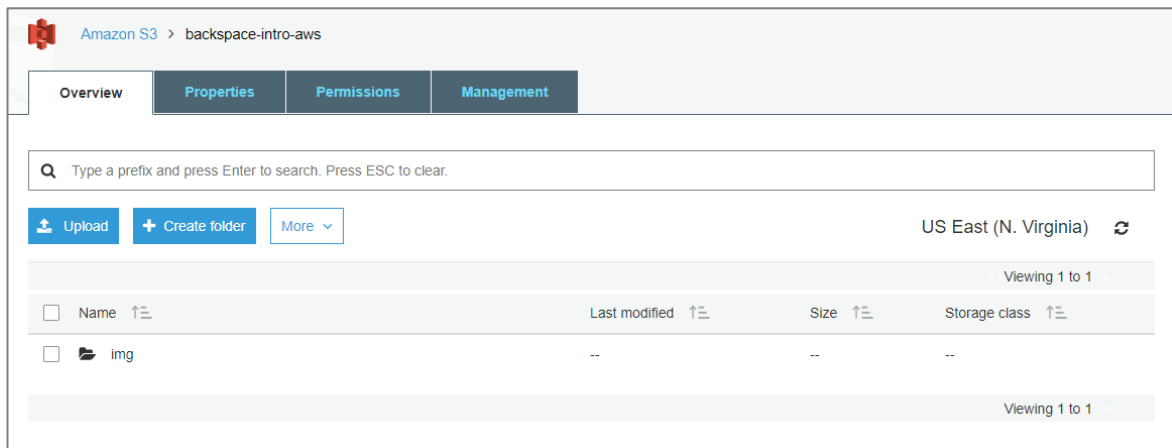
Upload Previous Next

Click “Upload”

**Upload** (0/s)

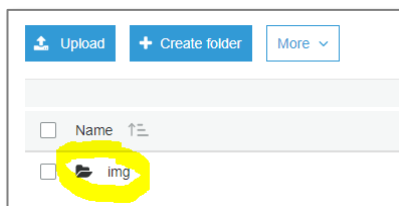
Operations 1 In progress 0 Success 0 Error

Your upload will eventually complete.

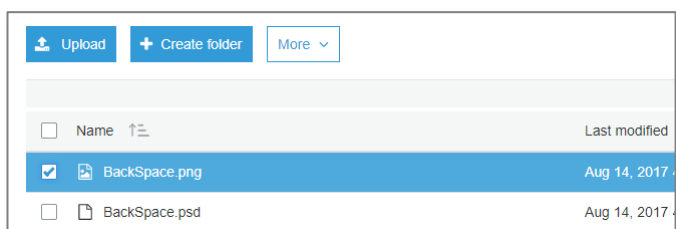


## Downloading files from your bucket

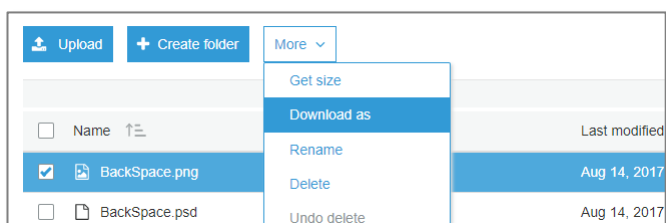
Click the link for your folder



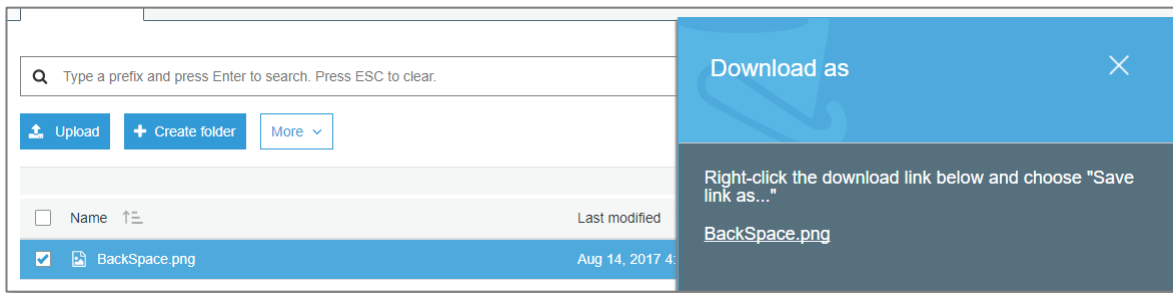
Select a file



Select "More", "Download As"



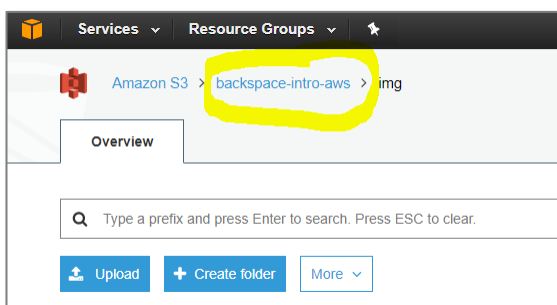
Click the download link to download the file.



## Clean Up

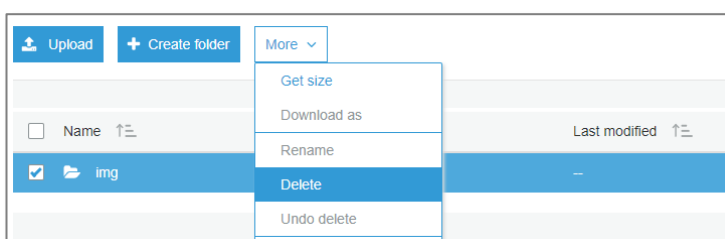
We will now delete the files and bucket so that you will not be billed by AWS.

Go back to your bucket by clicking its link.

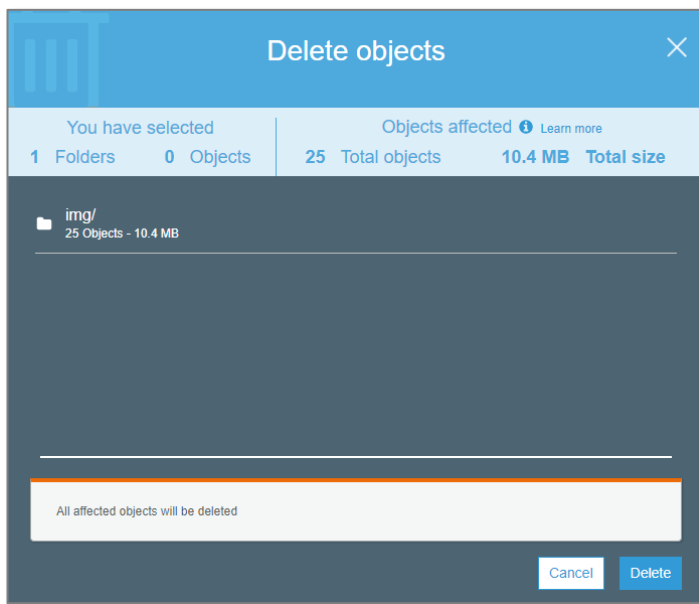


Select the folder

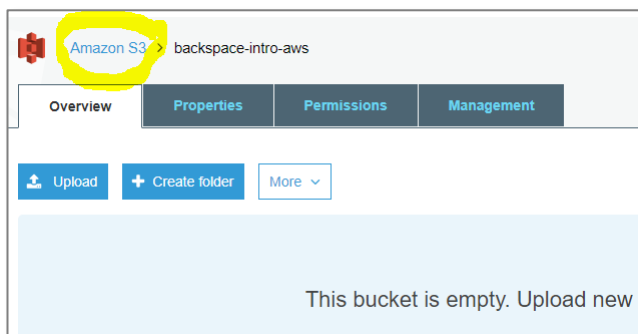
Select "More", "Delete"



Click "Delete"

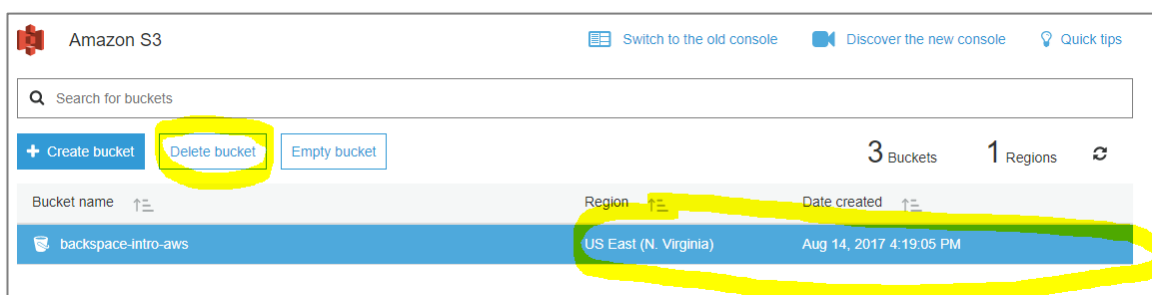


Go back to the S3 dashboard by clicking the link



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

Click "Delete Bucket"



Confirm the name of the bucket to delete

Delete bucket

Are you sure you want to delete the bucket "backspace-intro-aws" ?

Type the name of the bucket to confirm:  
backspace-intro-aws

Amazon S3 buckets are unique. If you delete this bucket, you may lose the bucket name to another AWS user.

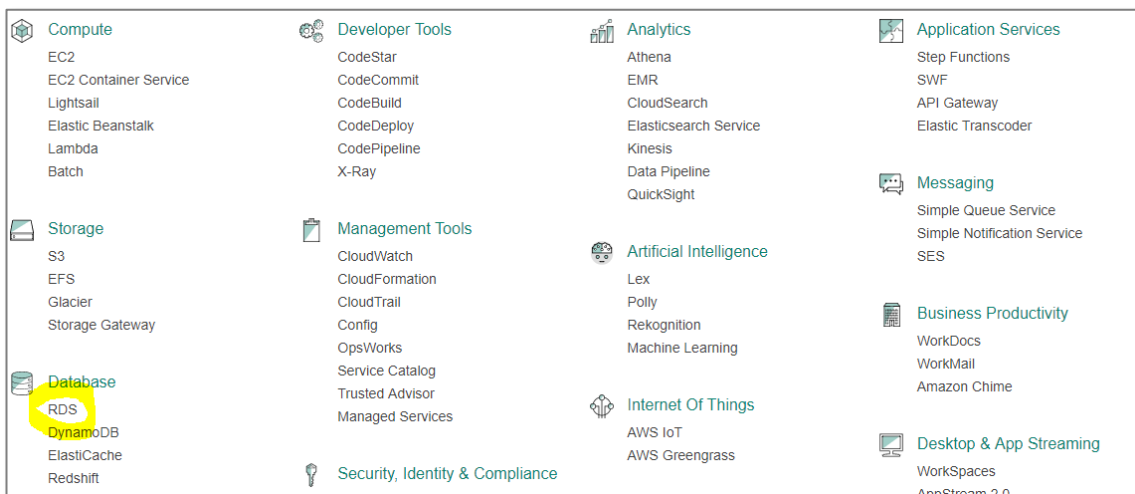
Cancel

Confirm

# Creating a SQL Database with RDS

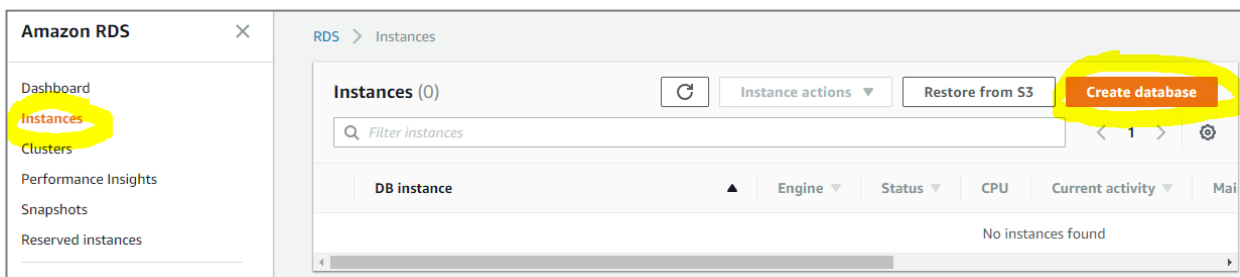
In this section, we will use the Relational Database Service to create a database. We will also connect in to the database.

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"

**Settings**

**DB instance identifier** [info](#)  
Specify a name that is unique for all DB instances owned by your AWS account in the current region.  
backspace-intro-aws  
DB instance identifier is case insensitive, but stored as all lower-case, as in "mydbinstance".  
Constraints:

- Must contain from 1 to 63 alphanumeric characters or hyphens (1 to 15 for SQL Server).
- First character must be a letter.
- Cannot end with a hyphen or contain two consecutive hyphens.

**Master username** [info](#)  
Specify an alphanumeric string that defines the login ID for the master user.  
admin  
Master Username must start with a letter. Must contain 1 to 16 alphanumeric characters.

**Master password** [info \*\*Confirm password\*\* \[info\]\(#\)  
Master Password must be at least eight characters long, as in "mypassword". Can be any printable ASCII character except /, ", "", or @.](#)

Cancel Previous **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)

**Network & Security**

**Virtual Private Cloud (VPC)** [info](#)  
VPC defines the virtual networking environment for this DB instance.  
Default VPC (vpc-72d25a0b) [refresh](#)  
Only VPCs with a corresponding DB subnet group are listed.

**Subnet group** [info](#)  
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.  
default

**Public accessibility** [info](#)  
☒ Yes  
EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.  
☐ No  
DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

**Availability zone** [info](#)  
No preference

**VPC security groups**  
Security groups have rules authorizing connections from all the EC2 instances and devices that need to access the DB instance.  
☒ Create new VPC security group  
☐ Choose existing VPC security groups

Enter a database name.

Leave all other options default as below.



**Database options**

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

Database port  
TCP/IP port the DB instance will use for application connections.  
3306

DB parameter group [info](#)  
default:mysql5.6


Option group [info](#)  
default:mysql-5-6

☐ 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.  
0 days

 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”

ⓘ 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 DB instance is being created.**  
Note: Your instance may take a few minutes to launch.

### Connecting to your DB instance

Once Amazon RDS finishes provisioning your DB instance, you can use a SQL client application or utility to connect to the instance.  
[Learn about connecting to your DB instance](#)

All DB instances **View DB instance details**

Your instance will show status "creating".

RDS > Instances > backspace-intro-aws

## backspace-intro-aws

Instance actions ▼

### Summary

Engine MySQL 5.6.39	DB instance class <a href="#">Info</a> db.t2.micro	<b>DB instance status</b> creating	Pending maintenance none
------------------------	---	---------------------------------------	-----------------------------

### CloudWatch (17)

Legend: backspace-intro-aws

Q

< 1 2 3 > ⚙

#### CPU Utilization (Percent)

1		
0.75		
0.5		

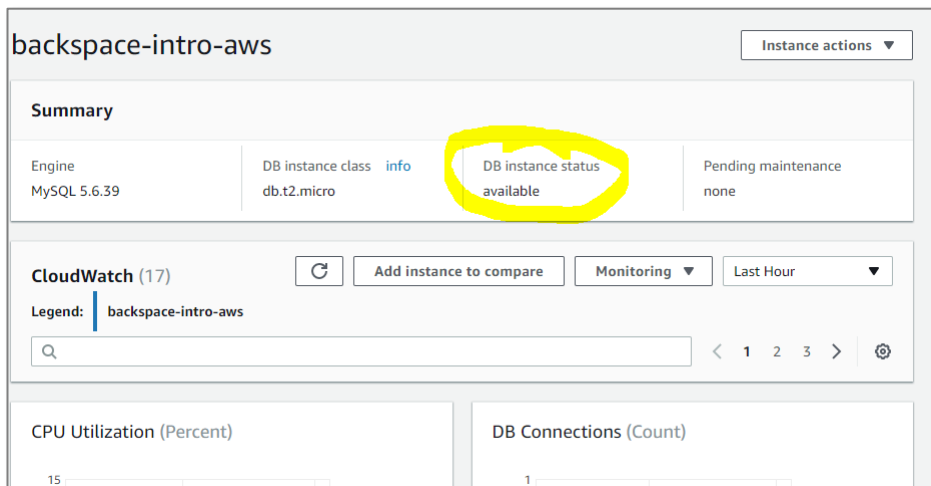
#### DB Connections (Count)

1		
0.75		
0.5		

## 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”

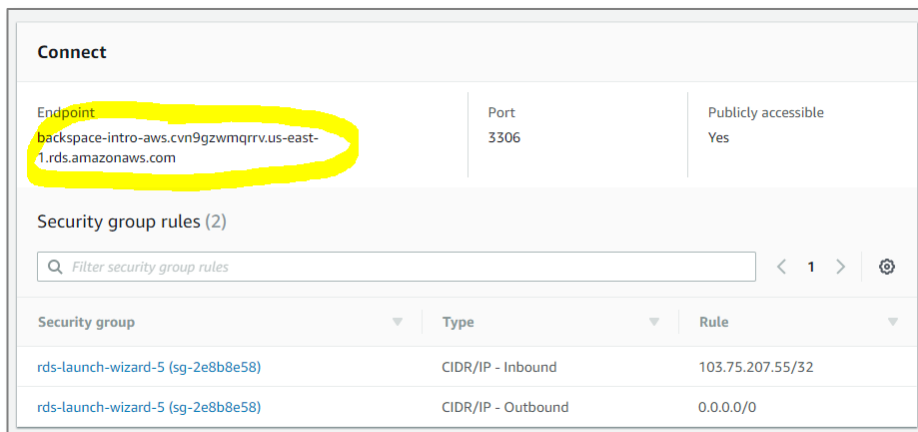


The screenshot shows the AWS Management Console for an RDS instance named 'backspace-intro-aws'. The 'Summary' section displays the following details:

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

The 'DB instance status' is highlighted in yellow and shows 'available'. Below the summary, there is a 'CloudWatch (17)' section with a legend for 'backspace-intro-aws'. At the bottom, there are two graphs: 'CPU Utilization (Percent)' and 'DB Connections (Count)'.

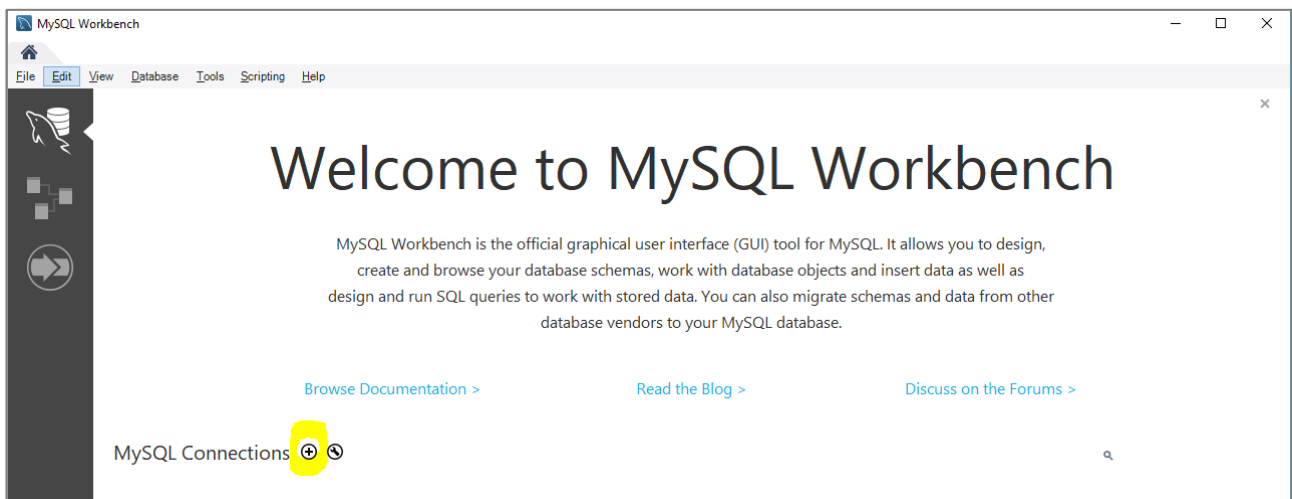
Scroll down and copy the database server endpoint



The screenshot shows the 'Connect' section of the AWS Management Console for the RDS instance. The 'Endpoint' is highlighted in yellow and shows 'backspace-intro-aws.cv9gzwmqrrv.us-east-1.rds.amazonaws.com'. The 'Port' is 3306 and 'Publicly accessible' is Yes. Below this, there is a 'Security group rules (2)' section with a table of rules:

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



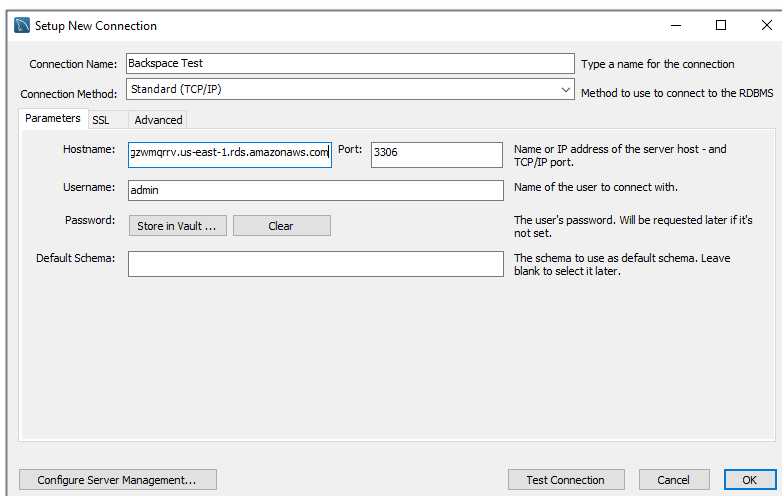
Give the connection a name.

The Hostname will be the RDS server endpoint with the “:3306” removed from the end.

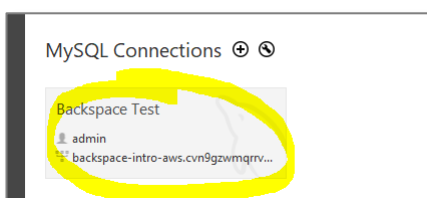
The port will be 3306.

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

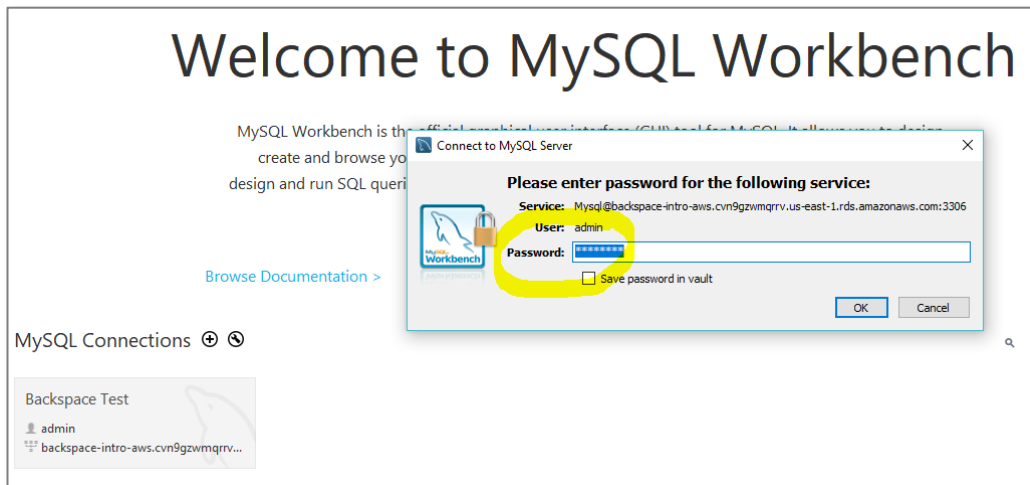
Click OK



Click on the Connection

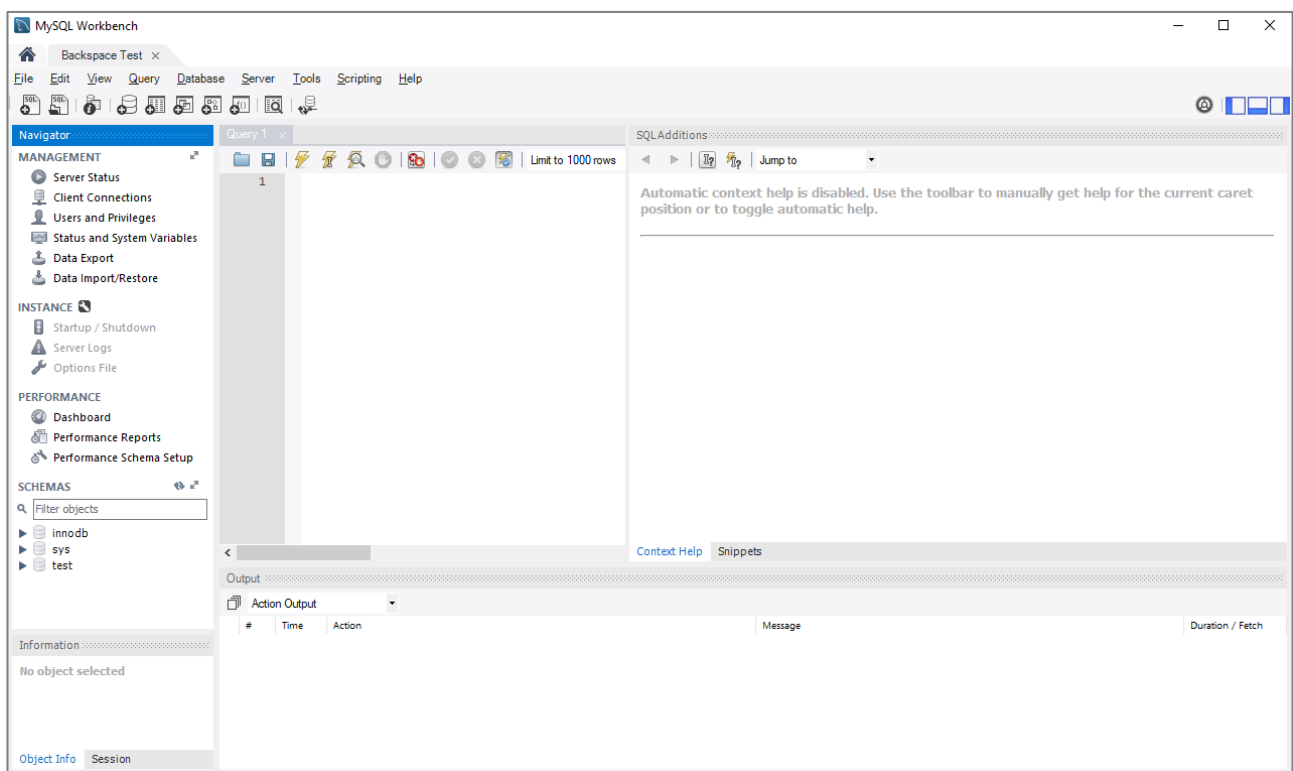


Enter the password you created in RDS for your master username

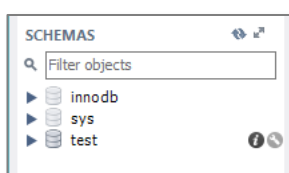


You will soon be connected to your database server

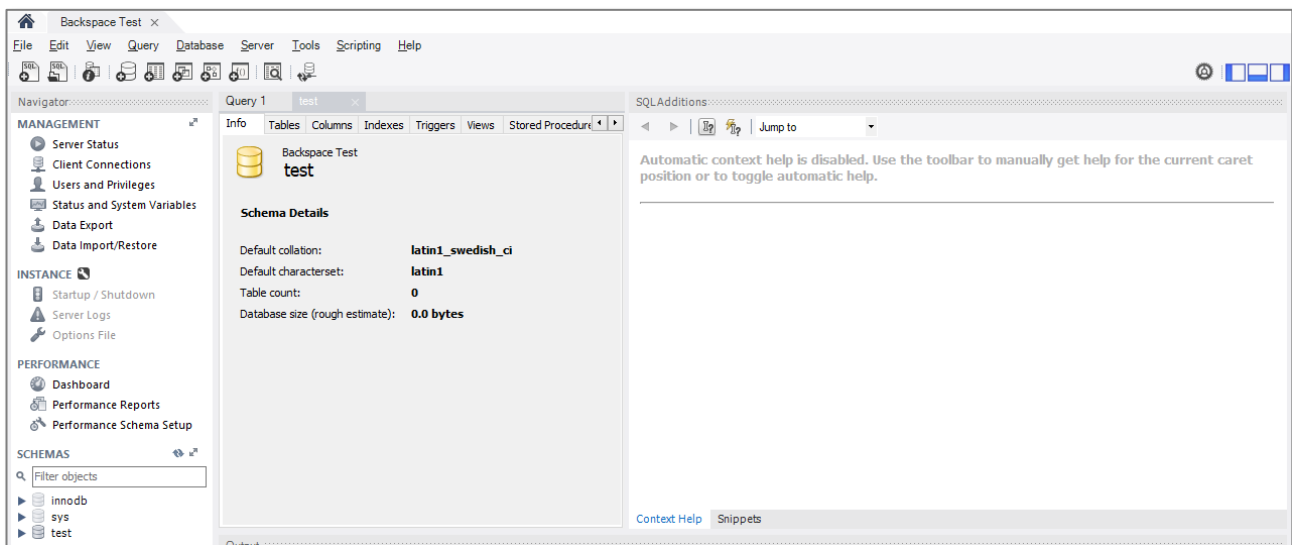
If you cannot connect then please see the “Troubleshooting Connection Issues” below.



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.



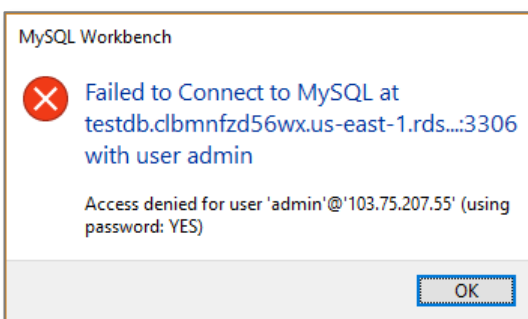
You then get an information screen for the database.



## Troubleshooting Connection Issues

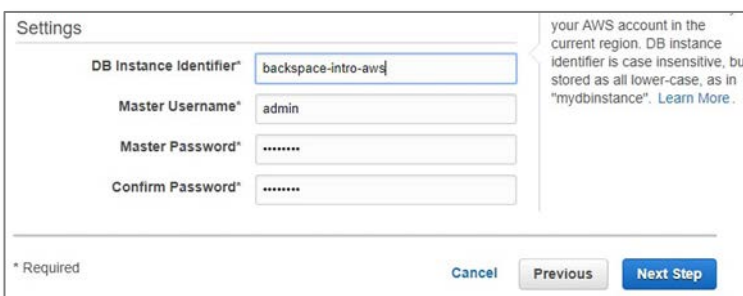
If you are getting connection errors then check the following:

### Wrong Username / Password

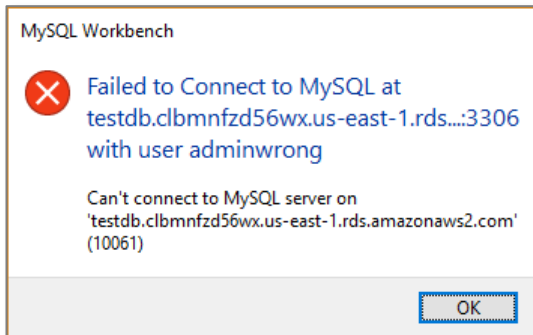


Make sure you use the correct username and password.

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



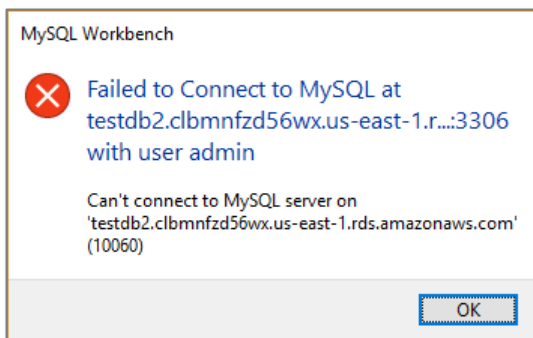
## Bad Connection String



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



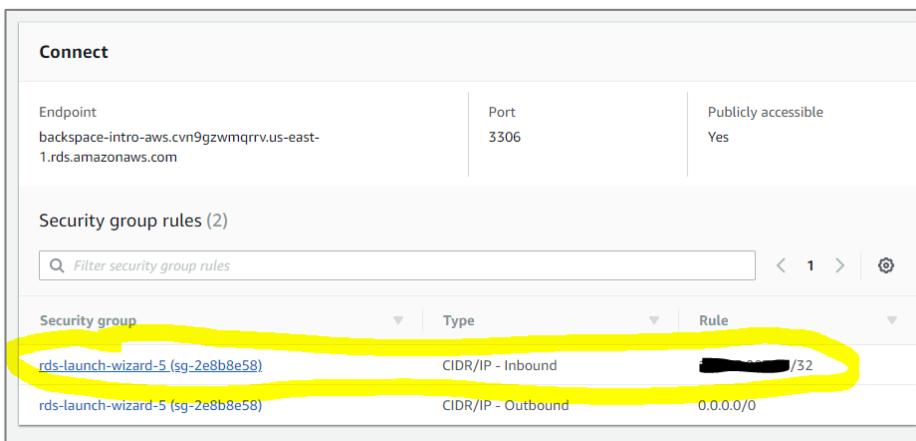
This error means your server exists but you are unable to connect to it. This can be caused by:

- 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.

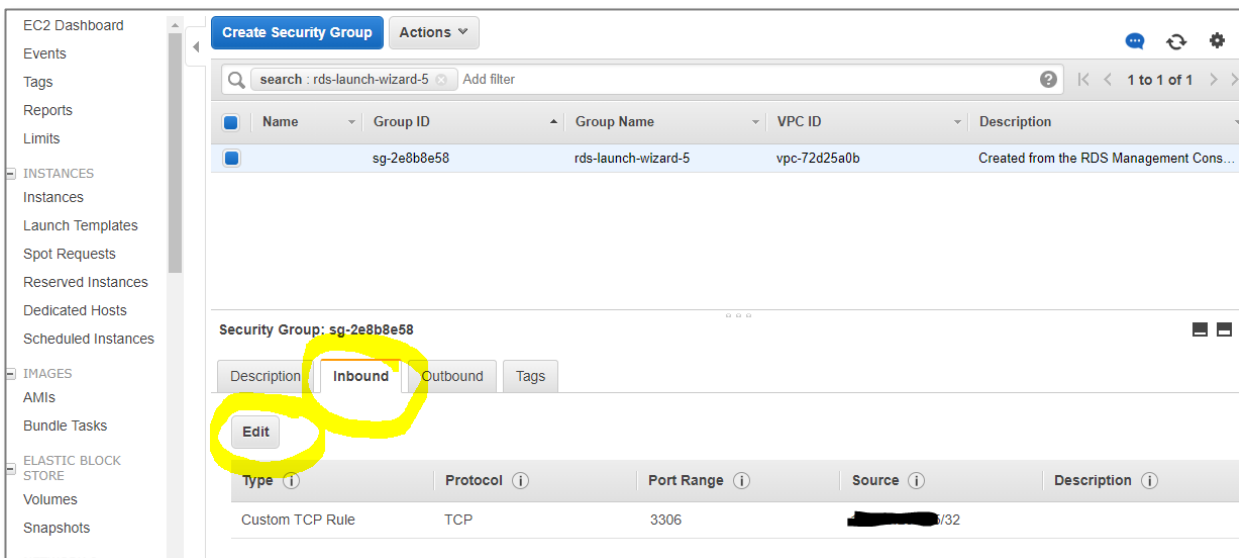
## Security Group Inbound Rules

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

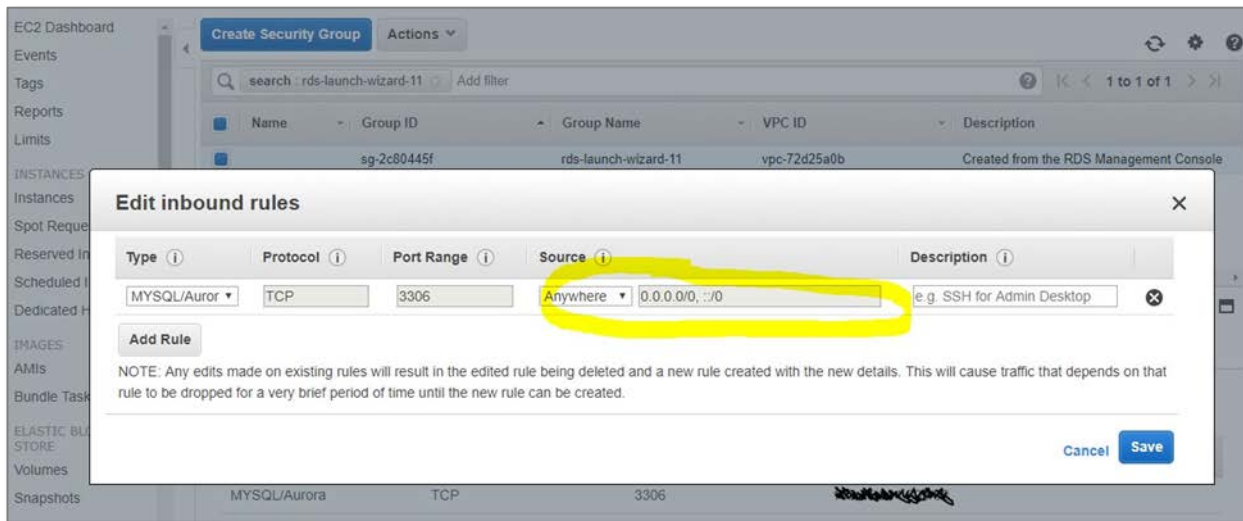


You will be taken to the EC2 console  
 Select the "Inbound" tab  
 Click "Edit"



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

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

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

```

Command Prompt - mysql -h backspace-intro-aws.clbmfnzd56wx.us-east-1.rds.amazonaws.com -u admin -p
F:\Program Files\MySQL\MySQL Workbench 8.0 CE>mysql -h backspace-intro-aws.clbmfnzd56wx.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;

```

Command Prompt - mysql -h backspace-intro-aws.clbmfnzd56wx.us-east-1.rds.amazonaws.com -u admin -p
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>

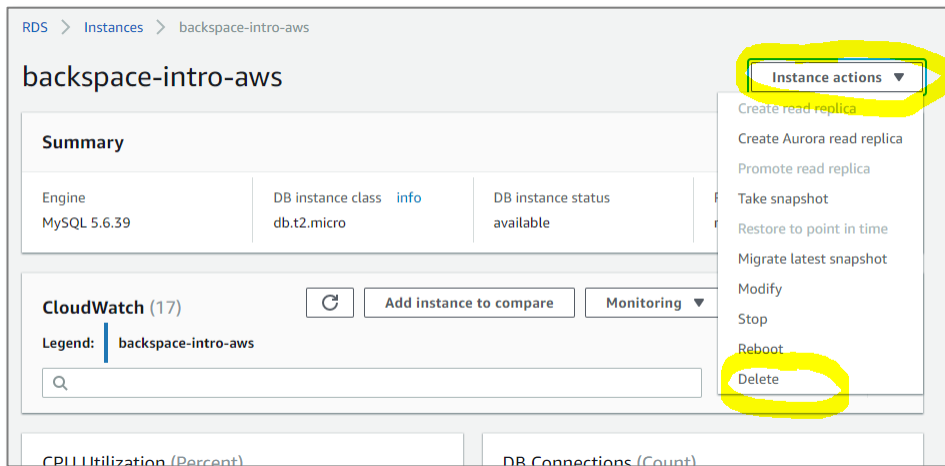
```

## Clean Up

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

Go back to the RDS console.

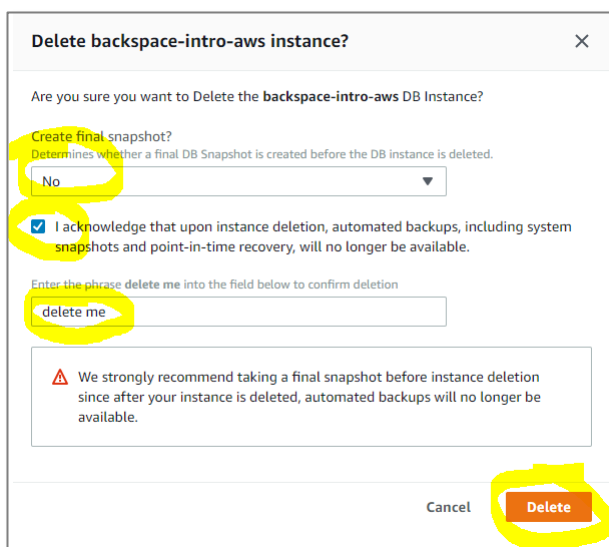
Click "Instance Actions", "Delete" to terminate the instance



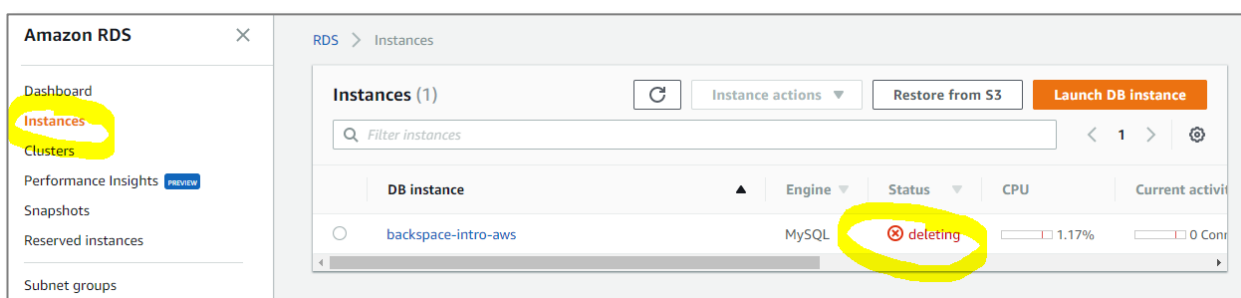
Select "No" for "Create final snapshot"

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

Click "Delete"



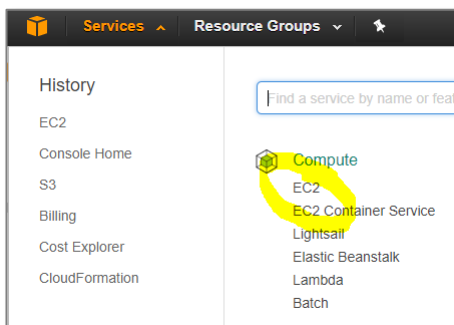
Click on "Instances" to see it status as "deleting"



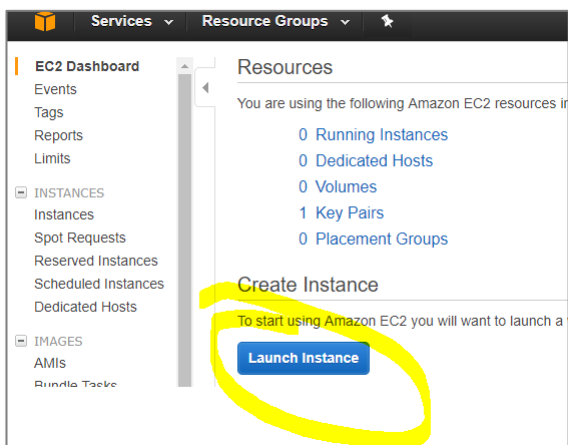
# Creating a Web Server with EC2

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

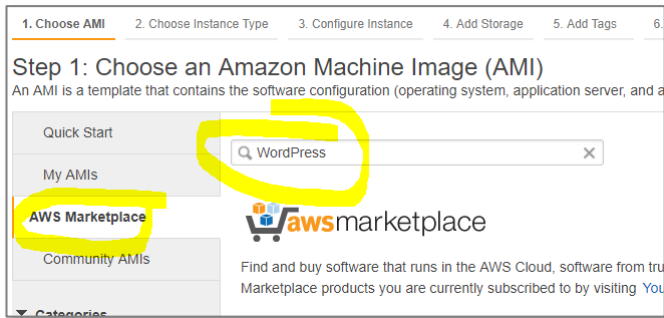
From the AWS console select “EC2” from the Compute services.



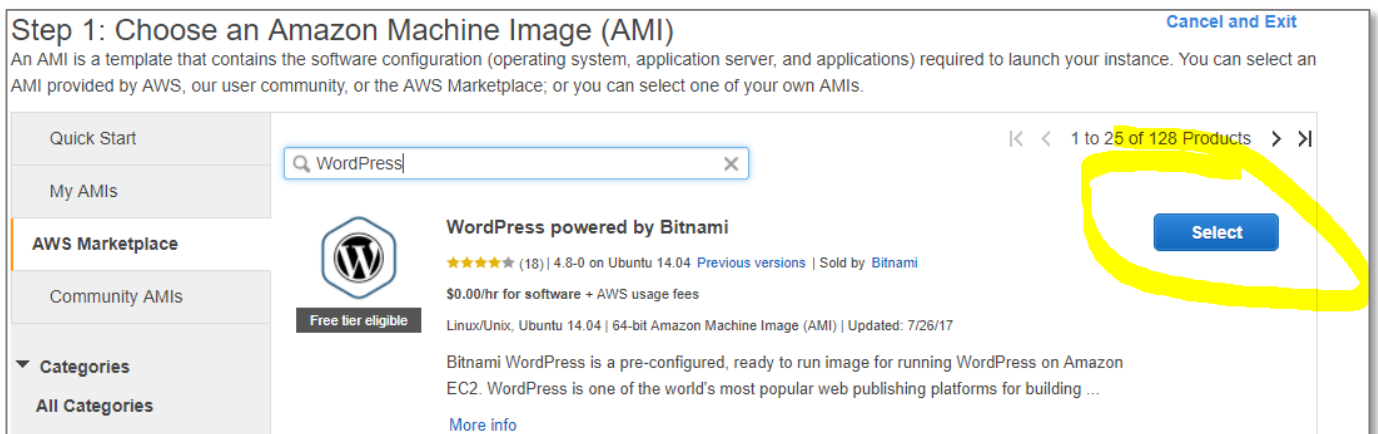
Select “Launch Instance”



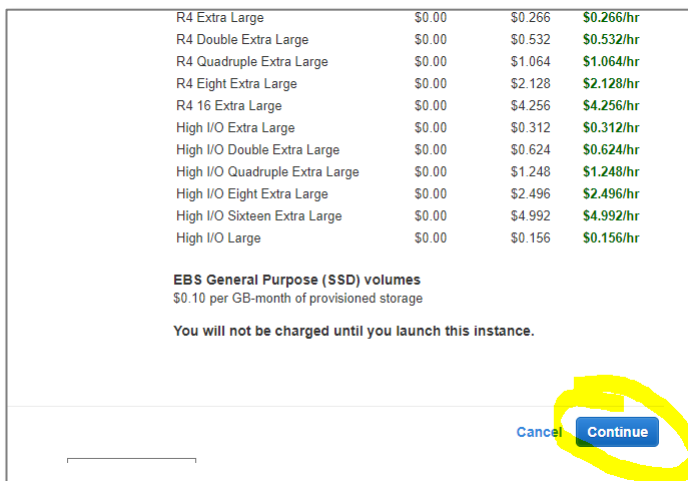
Select the “AWS Marketplace” and search for WordPress



Select the Bitnami AMI



Scroll to the bottom of the page and click “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**  [Launch into Auto Scaling Group](#)

**Purchasing option** ☐ Request Spot instances

**Network**  [Create new VPC](#)

**Subnet**  [Create new subnet](#)

**Auto-assign Public IP**

**IAM role**  [Create new IAM role](#)

**Shutdown behavior**

**Enable termination protection** ☐ Protect against accidental termination

**Monitoring** ☐ 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

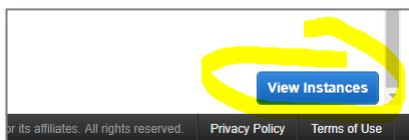
Please do not close your browser while this is loading

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.

EC2 Dashboard

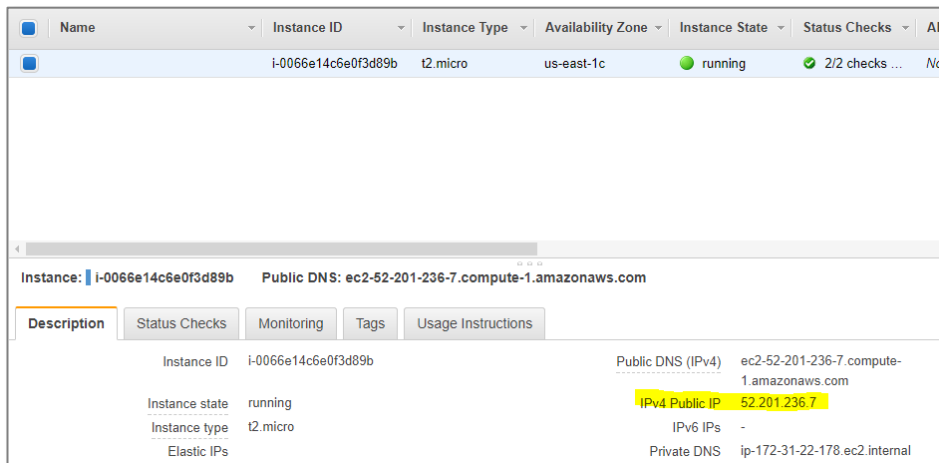
Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

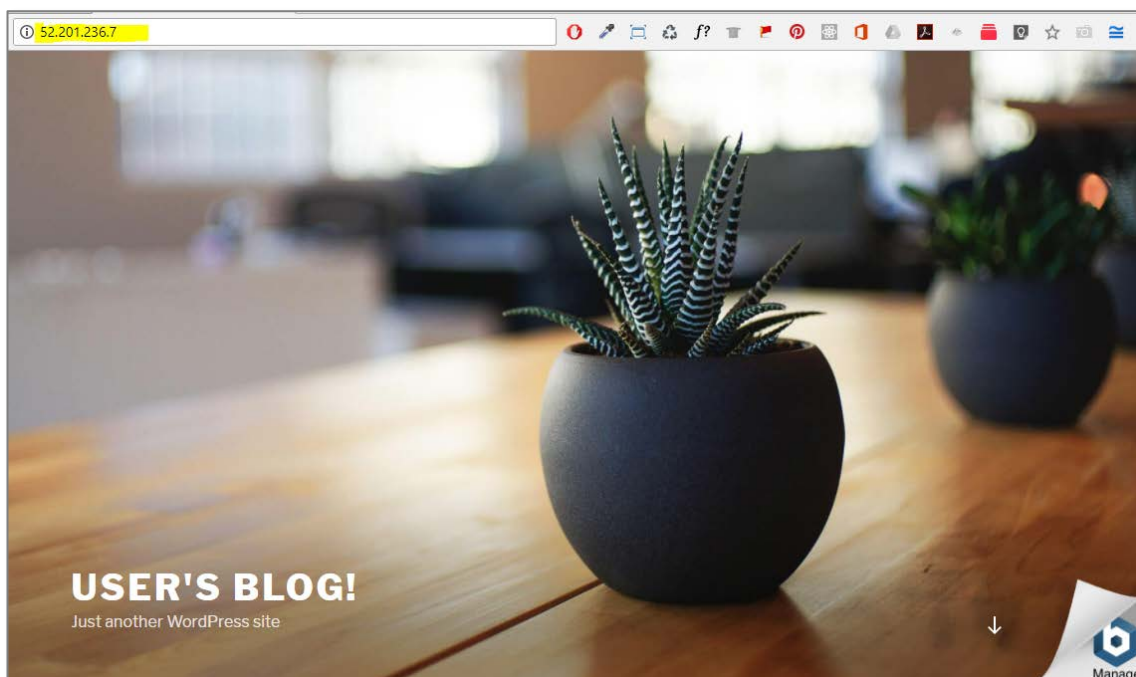
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
	i-0066e14c6e0f3d89b	t2.micro	us-east-1c	running	2/2 checks ...

## Viewing your web server

Copy the public IP address of your web server.



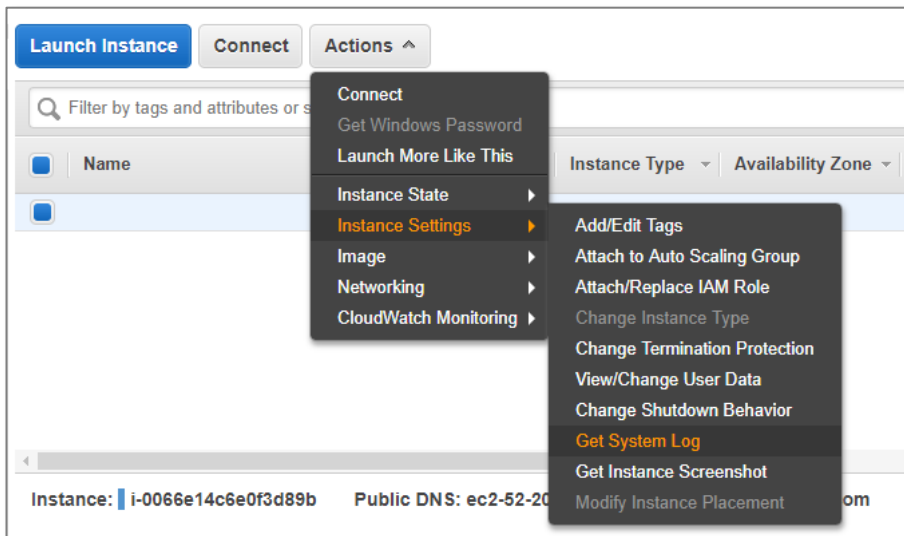
Navigate to the IP address in your browser.



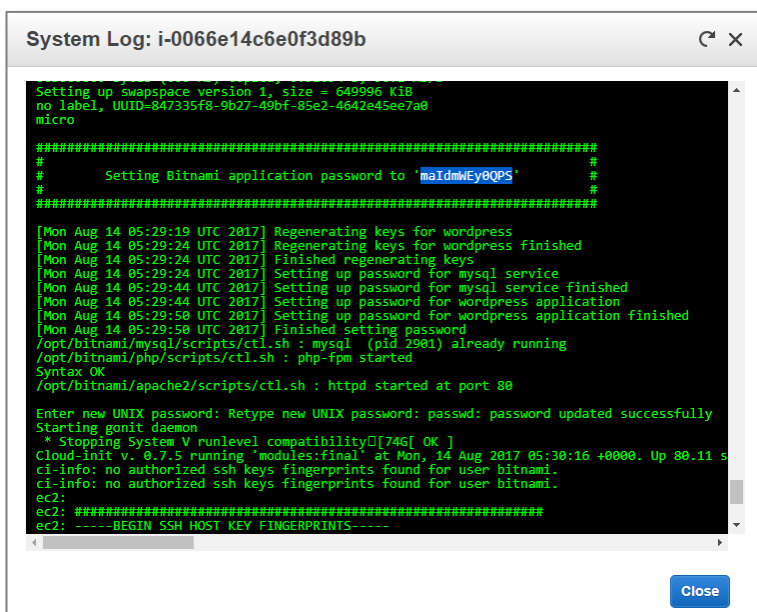
## Finding the Username and Password for your WordPress application

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

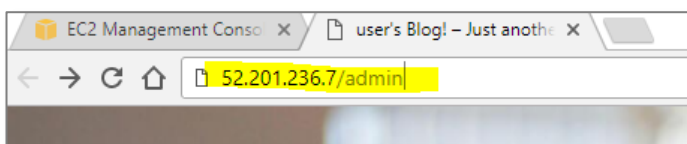




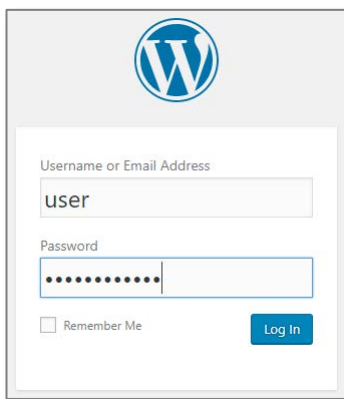
Scroll down until you find the log entry for the application password and copy it.



Go to the admin subdirectory of your website in your browser

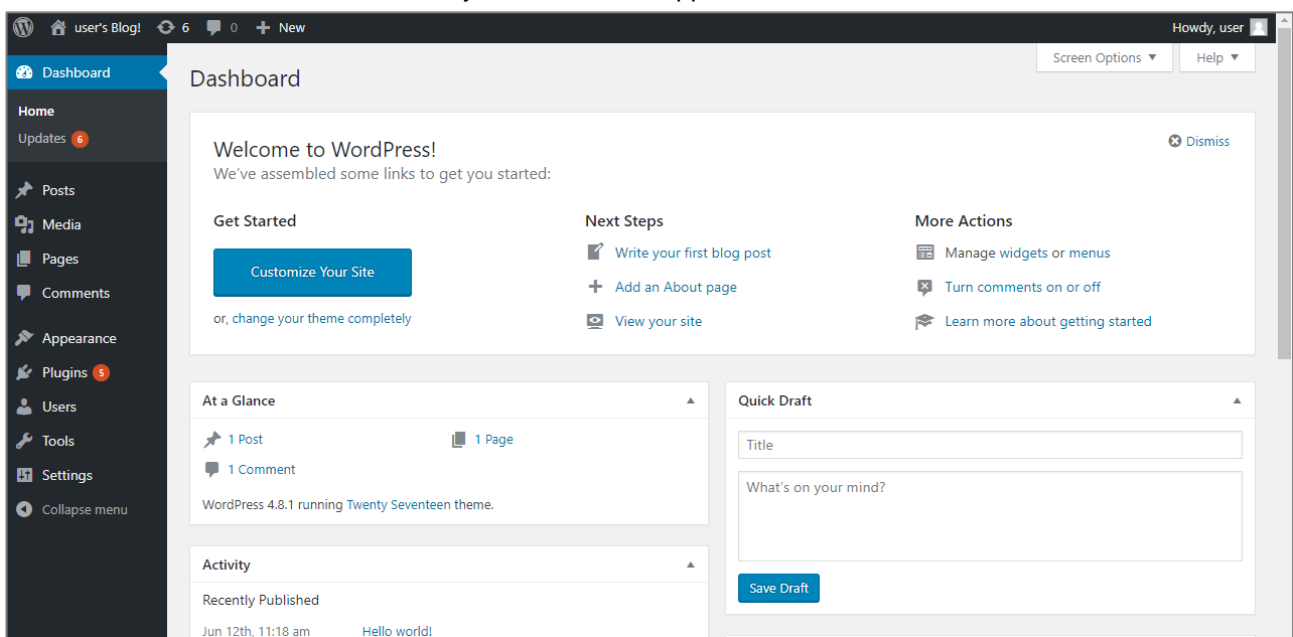


Enter Username "user" and paste in the password



WordPress login form showing the Username or Email Address field with 'user' entered, the Password field with masked characters, a 'Remember Me' checkbox, and a 'Log In' button.

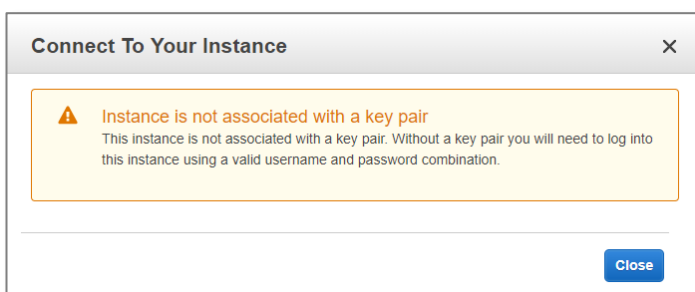
You will now be in the admin section of your WordPress application



WordPress Dashboard screenshot showing the 'Welcome to WordPress!' message, 'Get Started' button, 'Next Steps' (Write your first blog post, Add an About page, View your site), and 'More Actions' (Manage widgets or menus, Turn comments on or off, Learn more about getting started). The dashboard also displays 'At a Glance' (1 Post, 1 Page, 1 Comment) and 'Quick Draft' (Title, What's on your mind?, Save Draft).

## Troubleshooting logging in to the WordPress application

If you get the following message:

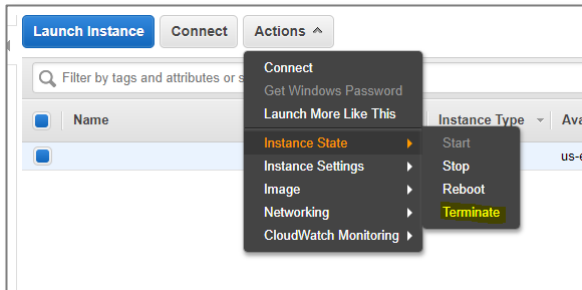


Connect To Your Instance error message: Instance is not associated with a key pair. This instance is not associated with a key pair. Without a key pair you will need to log into this instance using a valid username and password combination. Close

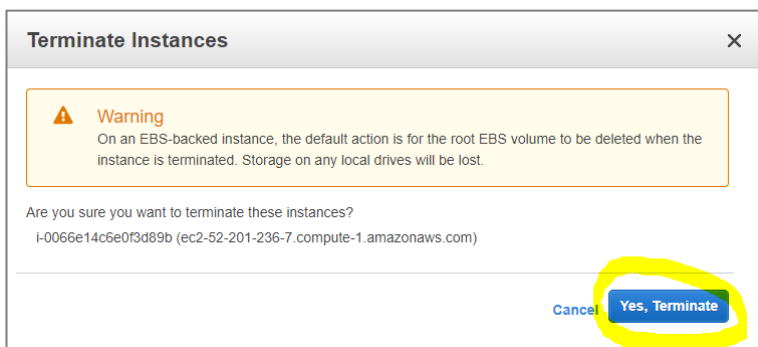
You have tried to connect to the Linux operating system by clicking on “Connect”. Do not click on connect, select “Actions – “Instance settings” - “Get System Log” as detailed previously.

## Clean up

Select “Actions”, “Instance State”, “Terminate”.



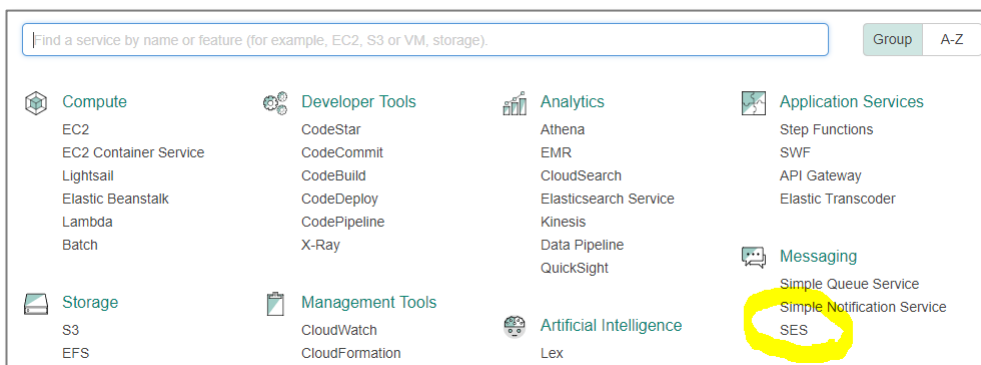
Make sure you terminate the instance so that you are not billed for it any more.



# Sending emails with Amazon SES

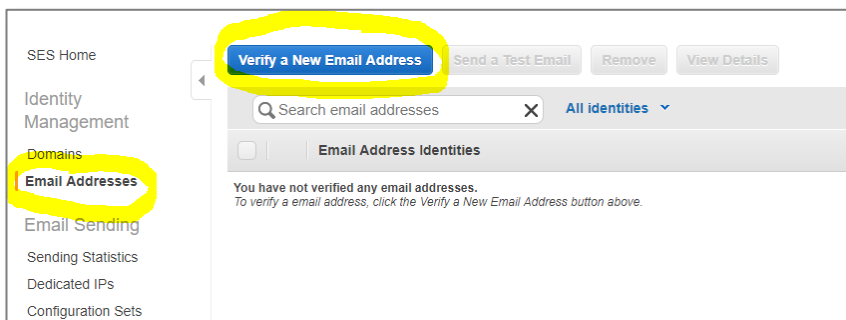
In this section, we will use the Simple Email Service to send an email.

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

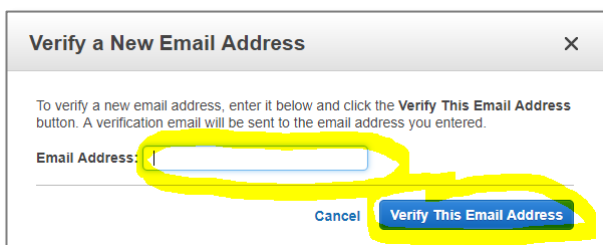


Click on "Email addresses"

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.

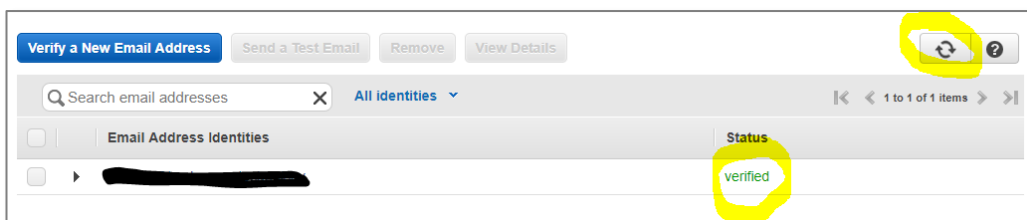
**For new Amazon SES users**—If you have not yet applied for a sending limit increase, then you are still in the [sandbox environment](#), and you can only send email to addresses that have been verified. To verify a new email address or domain, see the **Identity Management** section of the [Amazon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

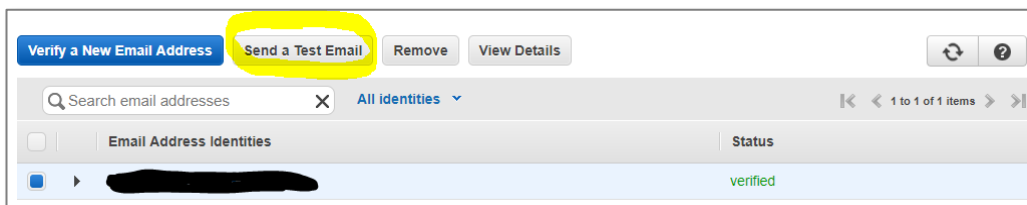
If you have already been approved for a sending limit increase, then you can start sending email to non-verified addresses.

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



Click on the email address and select “Send a test email”



Enter the same email address for from and to.

Fill out the email information and click “Send test email”

A screenshot of the 'Send Test Email' dialog box. It contains fields for 'From\*', 'To\*', 'Subject\*', and 'Body'. The 'From\*' and 'To\*' fields are redacted. 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 circled in yellow.

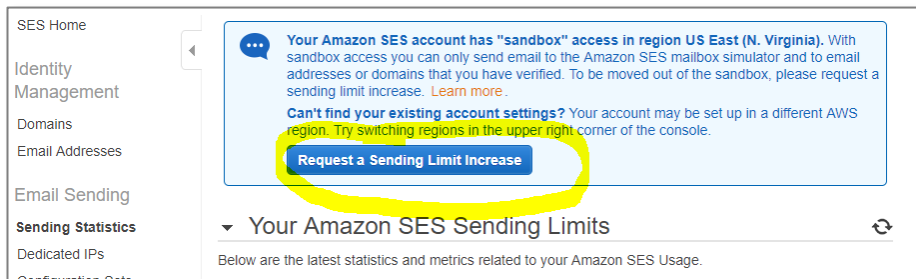
Check your email to see if it worked.

## Requesting full access to SES

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

Click on "Sending Statistics"

Click on "Request a Sending Limit Increase"



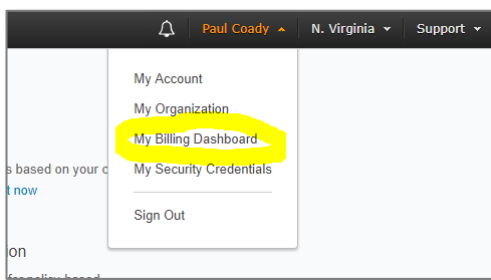


# Creating a Billing Alert with CloudWatch and SNS

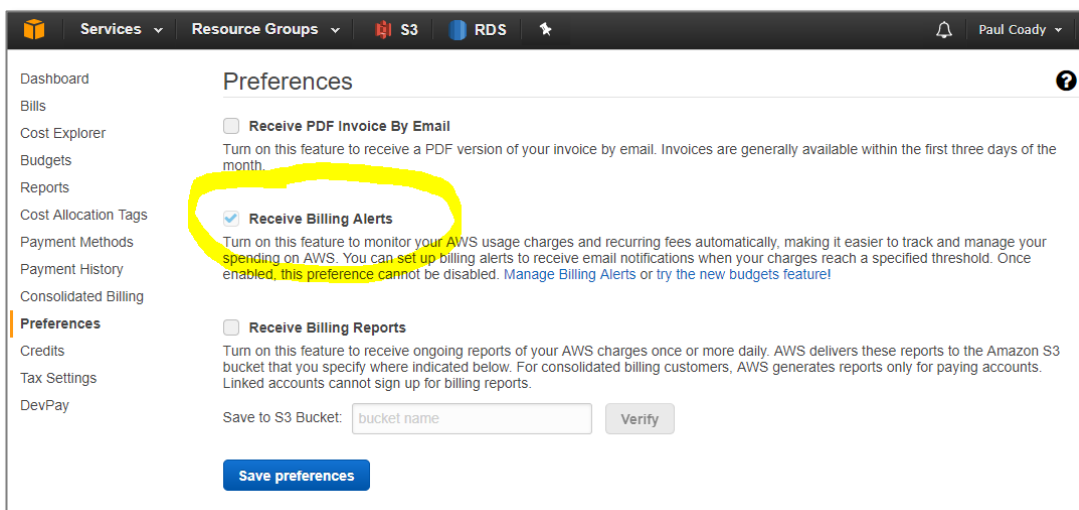
In this section, we will create a CloudWatch billing alert that will send an email through the Simple Notification Service whenever our estimated monthly bill exceeds a certain level.

## Enabling Billing Alerts

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

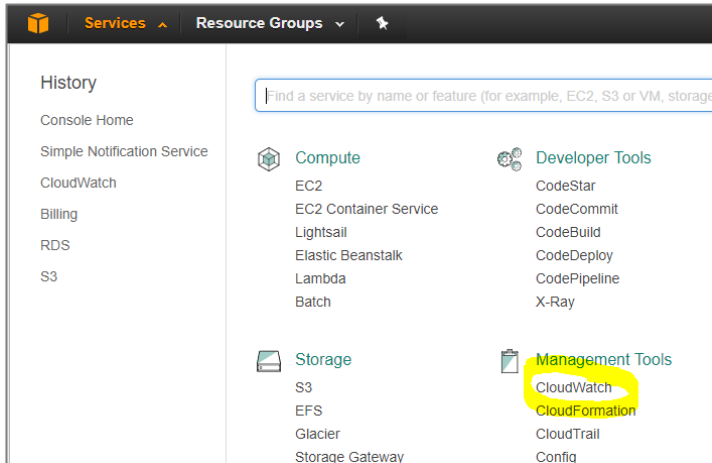


Select “Preferences” and check “Receive Billing Alerts”

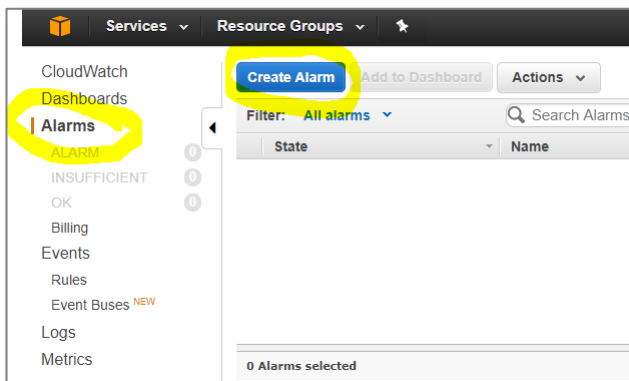


## Creating a CloudWatch Alarm

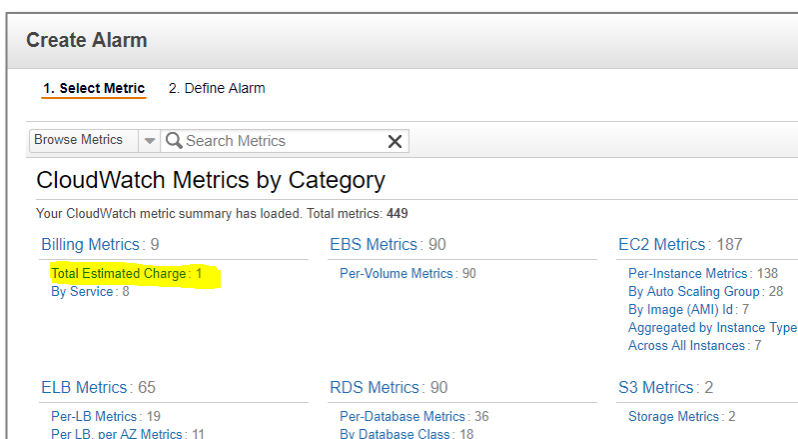
Click the Services menu and select “CloudWatch” from “Management Tools”



Click on “Alarms”, “Create Alarm”



Select “Total Estimated Charge” from the billing metrics.



Select EstimatedCharges metric (you may need to drag the divider down to see it)

Click “Next”



**Create Alarm**

1. Select Metric 2. Define Alarm

Billing Search Metrics X

Total Estimated Charge By Service

Billing > Total Estimated Charge

Currency Metric Name

☒ USD EstimatedCharges

Title: EstimatedCharges Maximum 6 Hours

Update Graph

Time Range

Relative Absolute UTC (GMT)

From: 12.03 hours ago

To: 0 hours ago

Zoom: 1h | 3h | 6h | 12h | 1d | 3d | 1w | 2w

Left Y-axis

Limits Min 0 Max

Cancel Previous **Next** Create Alarm

Give the alarm a name and description.

**Create Alarm**

1. Select Metric 2. Define Alarm

**Alarm Threshold**

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name: Monthly Billing Alarm

Description: Forecast costs have been exceeded

Whenever charges for: EstimatedCharges

is: >= USD \$ 0

**Additional settings**

Provide additional configuration for your alarm.

Treat missing data as: missing

**Alarm Preview**

This alarm will trigger when the blue line goes up to or above the red line

EstimatedCharges >= 0

0.013 0.011 0.008 0.006 0.003 0

8/11 00:00 8/13 00:00 8/15 00:00

Namespace: AWS/Billing

Currency: USD

Metric Name: EstimatedCharges

Set the alarm threshold to \$10

**Alarm Threshold**

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name: Monthly Billing Alert

Description: Forecast monthly costs exceeded

Whenever charges for: EstimatedCharges

is: >= USD \$ 10

Scroll down to the actions section. Click on "New List"

### Actions

Define what actions are taken when your alarm changes state.

Notification Delete

Whenever this alarm: State is ALARM

Send notification to: Select a notification list New list: Enter list

+ Notification
+ AutoScaling Action
+ EC2 Action

The topic a name "monthly-billing-alert" and put in your email address.

### Actions

Define what actions are taken when your alarm changes state.

Notification Delete

Whenever this alarm: State is ALARM

Send notification to: monthly-billing-alert Select list

Email list: user1@example.net,

+ Notification
+ AutoScaling Action
+ EC2 Action

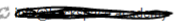
Click Create Alarm.

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

### Confirm new email addresses

Check your email inbox for a message with the subject "AWS Notification - Subscription Confirmation" and click the included link to confirm that you are willing to receive alerts to that address. AWS can only send notifications to confirmed addresses

Waiting for confirmation of 1 new email address

 [Resend confirmation link](#)

Note: You have 72 hours to confirm these email addresses

I will do it later
View Alarm


Click on confirm subscription in the email you receive.

You have chosen to subscribe to the topic:  
arn:aws:sns:us-east-1:950302654420:monthly-billing-alert

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):  
[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](mailto:sns-opt-out)

If all goes well you will see this page


Simple Notification Service

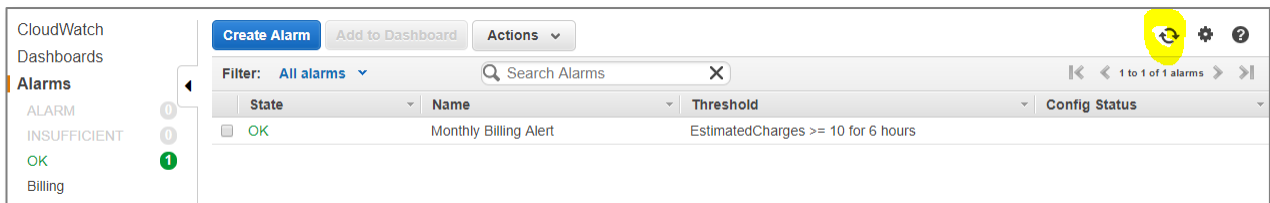
### Subscription confirmed!

You have subscribed info@backspace.academy to the topic:  
**monthly-billing-alert.**

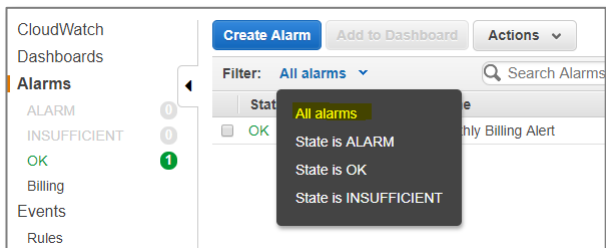
Your subscription's id is:  
arn:aws:sns:us-east-1:950302654420:monthly-billing-alert:69686c20-8f1d-472d-8ae2-37242a448d9a

If it was not your intention to subscribe, [click here to unsubscribe.](#)

Go back to the CloudWatch console and refresh the screen.



If you can't see your alarm then make sure "All alarms" is selected for the filter.

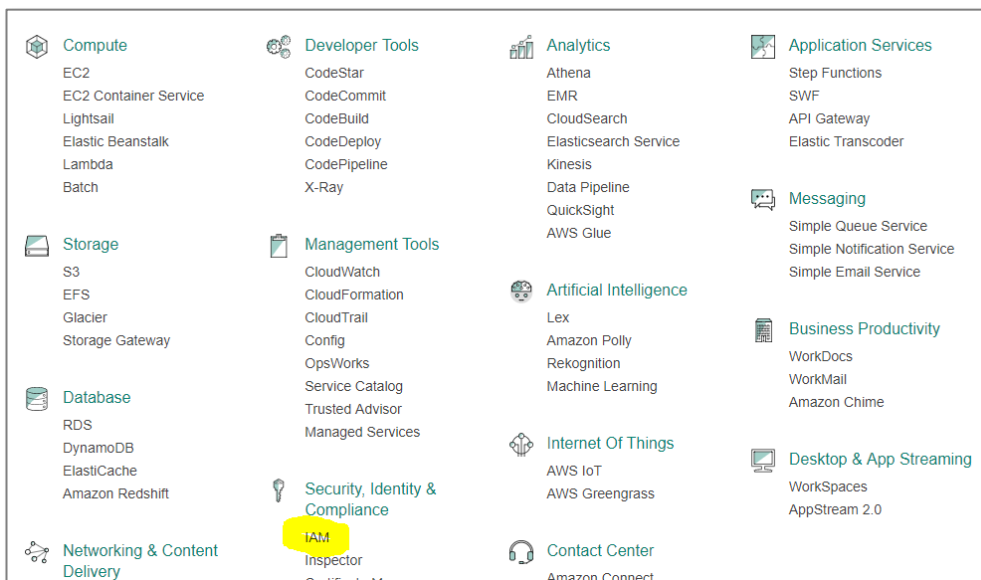


# Creating an IAM User

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

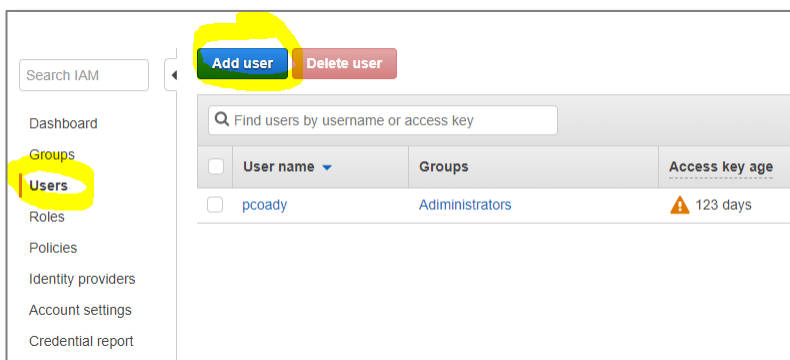
From the AWS console click “Services”

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\* test-user

+ 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

Cancel Previous **Next: Review**

Click "Create user"

### Add user

1 Details 2 Permissions 3 **Review** 4 Complete

#### 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

1 Details 2 Permissions 3 Review 4 **Complete**

✓ **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**

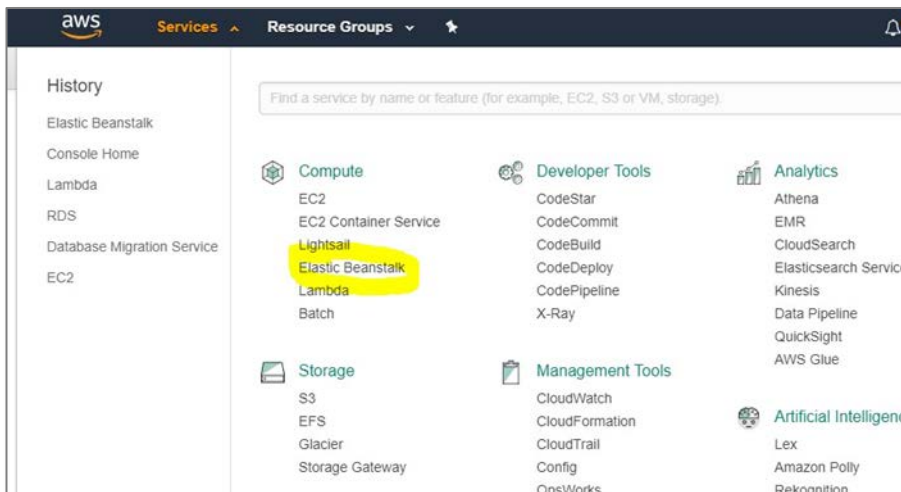
	User	Access key ID	Secret access key	Password	Email login instructions
▶	✓ test-user	AKIAJZGZ6UMOZT3U5V6Q	***** <a href="#">Show</a>	***** <a href="#">Show</a>	<a href="#">Send email</a> <a href="#">↗</a>

Close

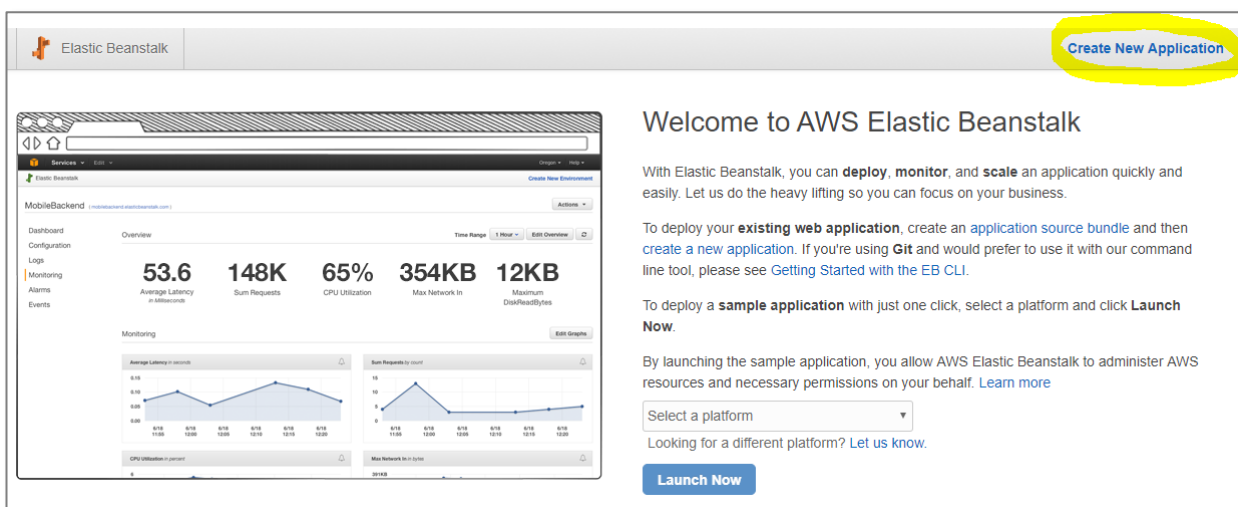
# **Creating** a Highly Available Architecture with Elastic Beanstalk

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*

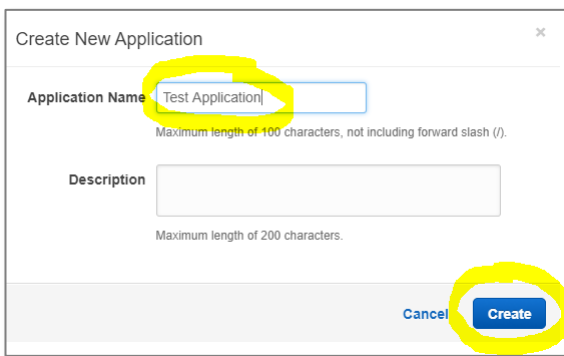


Click "Create New Application"



Give your application a name *Test Application*.

Click "Create"



Create New Application

Application Name

Maximum length of 100 characters, not including forward slash (/).

Description

Maximum length of 200 characters.

Cancel Create

Your application will now be created.

Select “Actions” - “Create Environment”



All Applications > Test Application

Environments

Application versions

Saved configurations

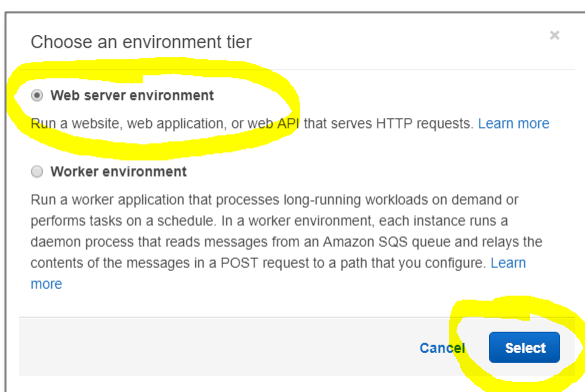
No environments currently exist for this application. [Create one now.](#)

Actions

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

Select “Web server environment”

Click “Select”



Choose an environment tier

☒ Web server environment

Run a website, web application, or web API that serves HTTP requests. [Learn more](#)

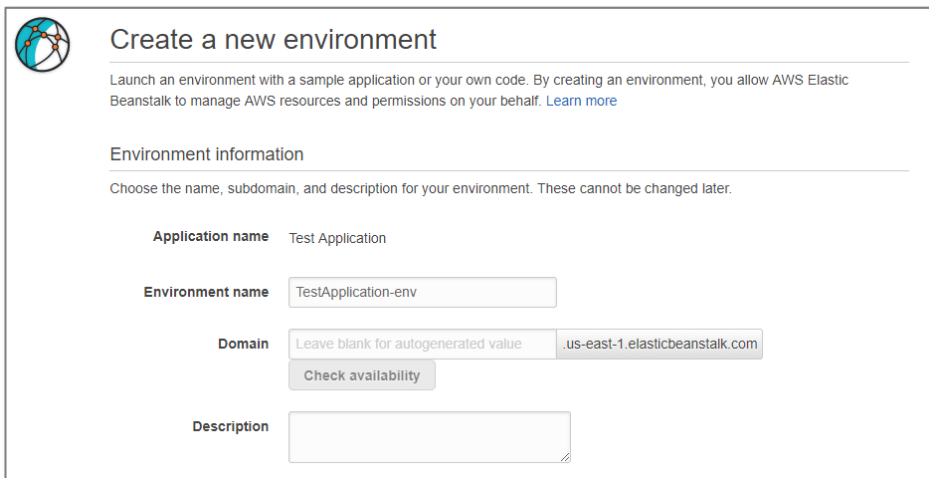
☐ Worker environment

Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. In a worker environment, each instance runs a daemon process that reads messages from an Amazon SQS queue and relays the contents of the messages in a POST request to a path that you configure. [Learn more](#)

Cancel 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**

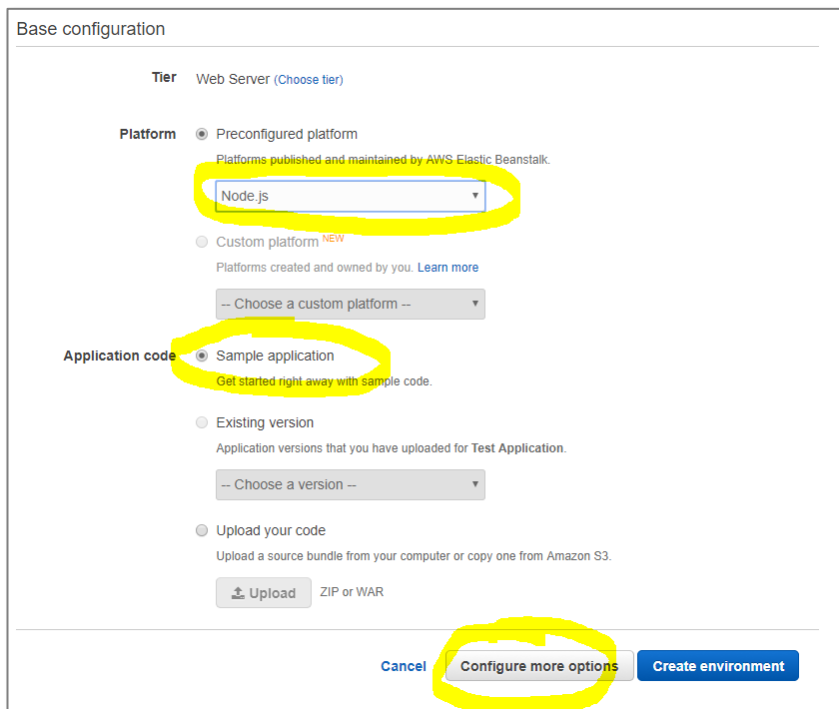
**Domain**

**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.

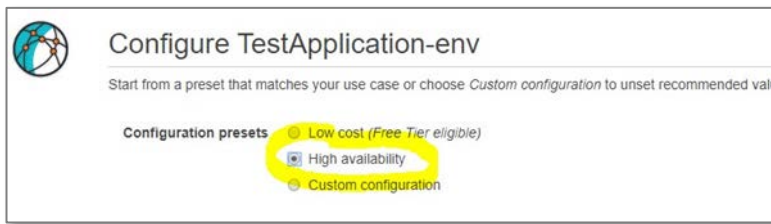
☐ Custom platform <sup>NEW</sup>  
Platforms created and owned by you. [Learn more](#)

**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.  
 ZIP or WAR

Select *High availability*



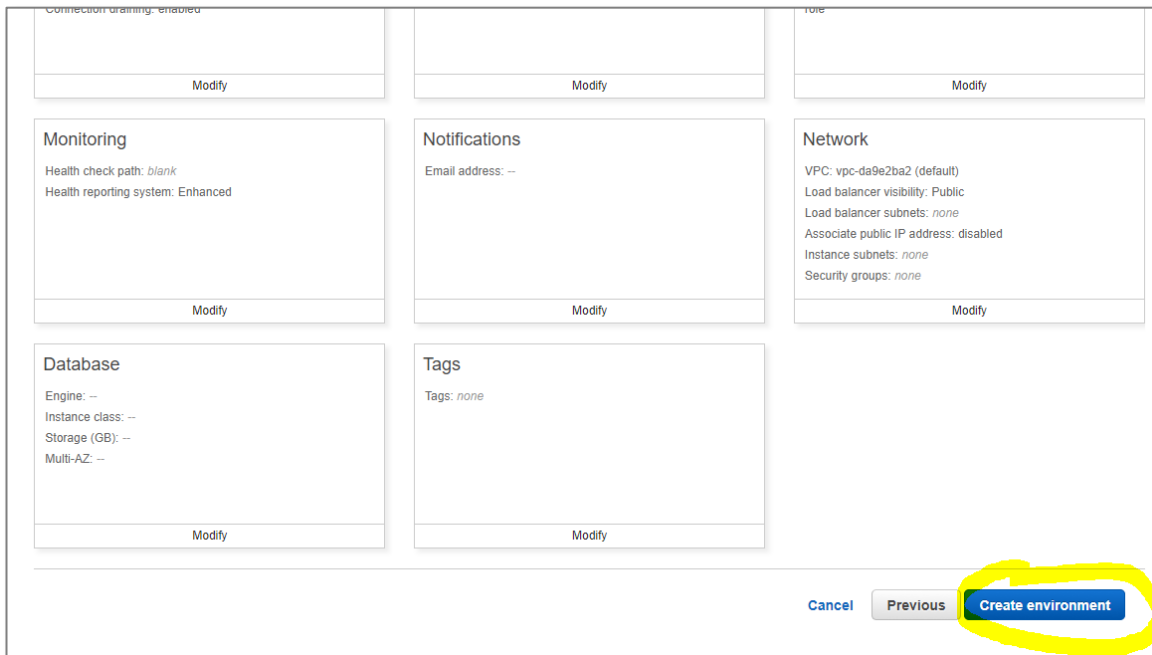
**Configure TestApplication-env**

Start from a preset that matches your use case or choose *Custom configuration* to unset recommended values.

**Configuration presets**

- ☐ Low cost (Free Tier eligible)
- ☒ High availability
- ☐ Custom configuration

Scroll down and click *Create environment*



Connection draining: enabled

Modify

**Monitoring**

Health check path: blank

Health reporting system: Enhanced

Modify

**Notifications**

Email address: --

Modify

**Network**

VPC: vpc-da9e2ba2 (default)

Load balancer visibility: Public

Load balancer subnets: none

Associate public IP address: disabled

Instance subnets: none

Security groups: none

Modify

**Database**

Engine: --

Instance class: --

Storage (GB): --

Multi-AZ: --

Modify

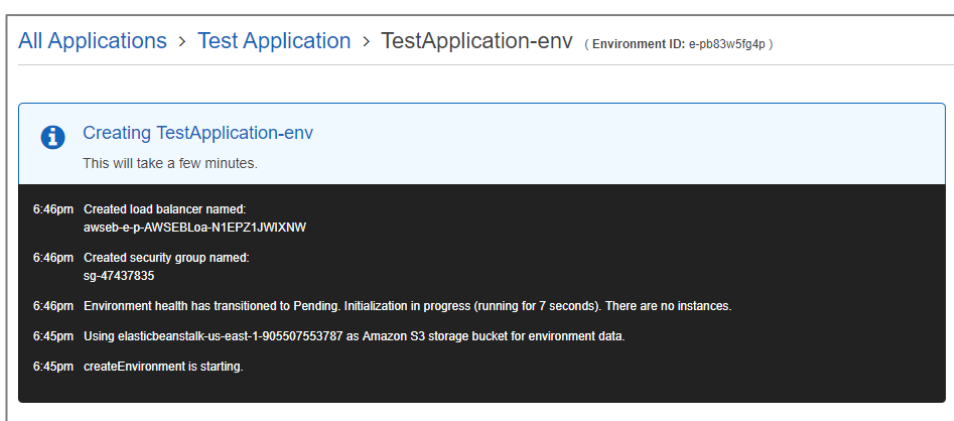
**Tags**

Tags: none

Modify

Cancel Previous **Create environment**

Your environment will start being created



All Applications > Test Application > TestApplication-env (Environment ID: e-pb83w5fg4p)

**Creating TestApplication-env**

This will take a few minutes.

```

6:46pm Created load balancer named:
awseb-e-p-AWSEBLoa-N1EPZ1JWIXNW
6:46pm Created security group named:
sg-47437835
6:46pm Environment health has transitioned to Pending. Initialization in progress (running for 7 seconds). There are no instances.
6:45pm Using elasticbeanstalk-us-east-1-905507553787 as Amazon S3 storage bucket for environment data.
6:45pm createEnvironment is starting.
  
```

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**) Actions

Dashboard Overview Refresh

Configuration

Logs

Health  Health **Ok** Causes

Monitoring

Alarms

Managed Updates

Events

Tags

Running Version

Sample Application

Upload and Deploy

nodejs

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.mxafx3y3j9.us-east-1.elasticbeanstalk.com](https://TestApplication-env.mxafx3y3j9.us-east-1.elasticbeanstalk.com)) Actions ▾

Dashboard Overview Refresh

Configuration

Logs Health Monitoring Alarms Managed Updates Events Tags

Health **Ok** Causes

Running Version Sample Application Upload and Deploy

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

Recent Events Show All

Click Actions

Select Delete Application

All Applications > Test Application

Environments Application versions Saved configurations

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 18:50:40 UTC+1100  
URL: TestApplication-env.mxafx3y3j9.us-east-1.elasticb...

Actions ▾

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

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.mxafx3y3j9.us-east-1....

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

 Elastic Beanstalk is terminating your environment.  
[View Events](#)

Overview Refresh



Health


Ok

Causes

Running Version

Sample Application

Upload and Deploy



Configuration

64bit Amazon Linux 2017.03  
v4.3.0 running Node.js

Change

Copyright 2017 all rights reserved - [BackSpace.Academy](#)

52