



lab title

**Introduction to AWS  
V1.09**



Course title

**BackSpace Academy  
AWS Certified Associate**



# Table of Contents

## Contents

Table of Contents.....	1
About the Lab .....	2
Checking your AWS Usage and Monthly Bill.....	3
Creating an S3 Bucket and Uploading Files.....	4
Uploading Files to your Bucket.....	7
Downloading files from your bucket .....	9
Clean Up.....	10
Creating a SQL Database with RDS.....	13
Connecting to your RDS Instance .....	16
Troubleshooting Connection Issues .....	20
Connecting to your RDS Instance using the Command Line .....	20
Clean Up.....	22
Creating a Web Server with EC2 .....	24
Viewing your web server .....	28
Finding the Username and Password for your WordPress application.....	29
Clean up .....	31
Sending emails with Amazon SES.....	32
Requesting full access to SES.....	34
Creating a Billing Alert with CloudWatch and SNS.....	35
Enabling Billing Alerts .....	35
Creating a CloudWatch Alarm .....	35
Creating a Highly Available Architecture with Elastic Beanstalk .....	40
Clean Up.....	44

## 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.**

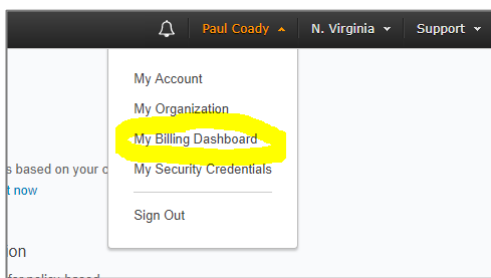
These lab notes are to support the hands on instructional videos of the Introduction to AWS section of the AWS Certified Associate Course.

**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.**

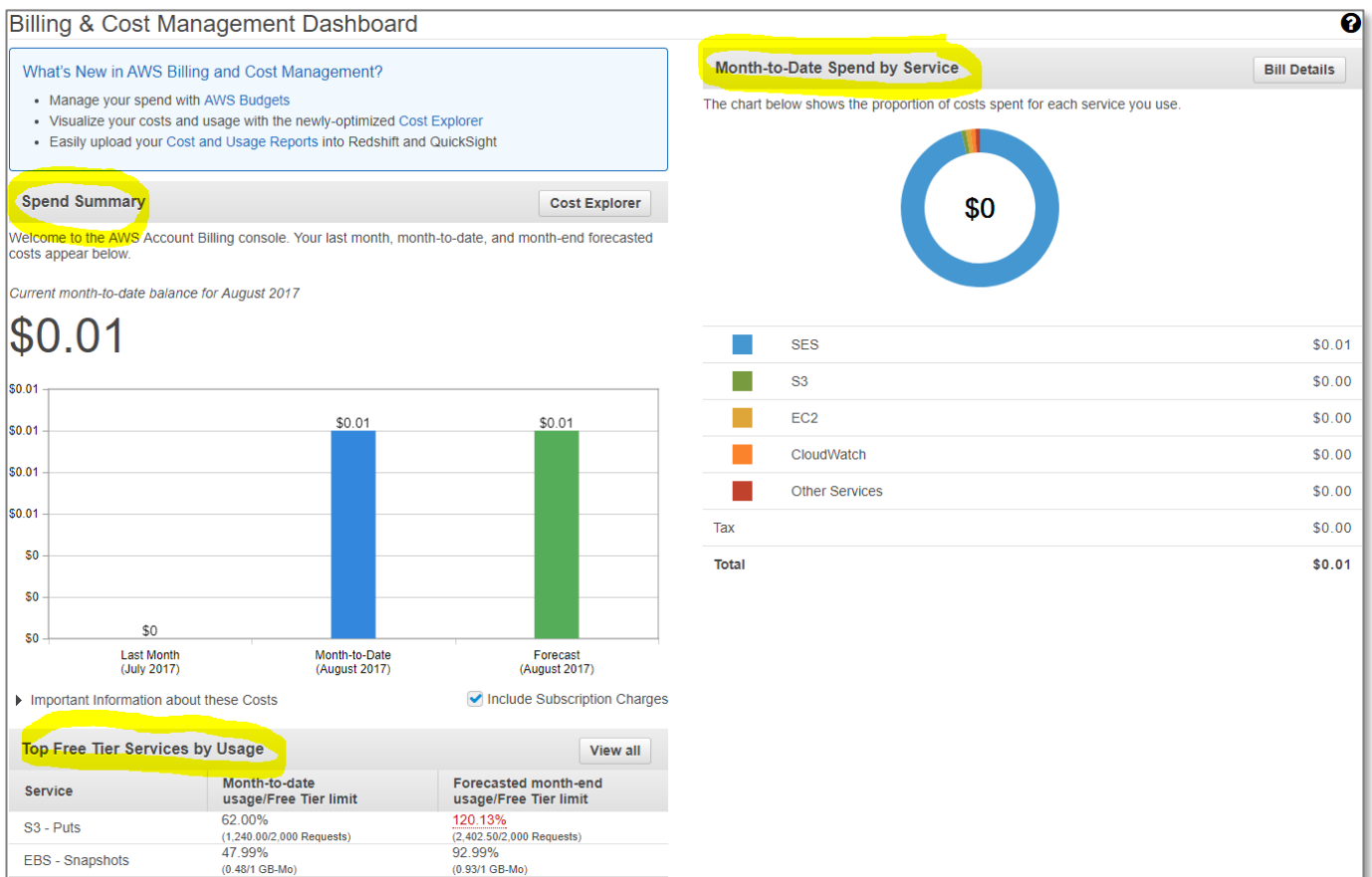
# Checking your AWS Usage and Monthly Bill

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.



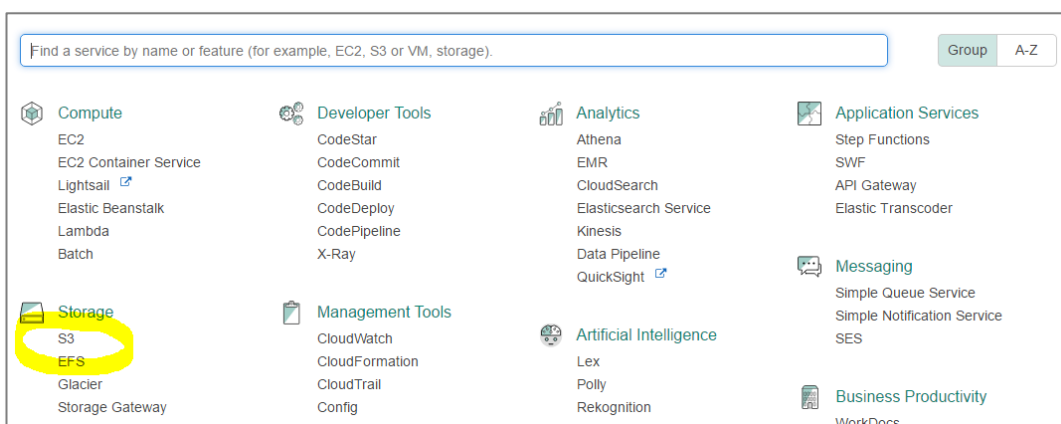
You will now see your total spend summary, spend by service and forecast spend.



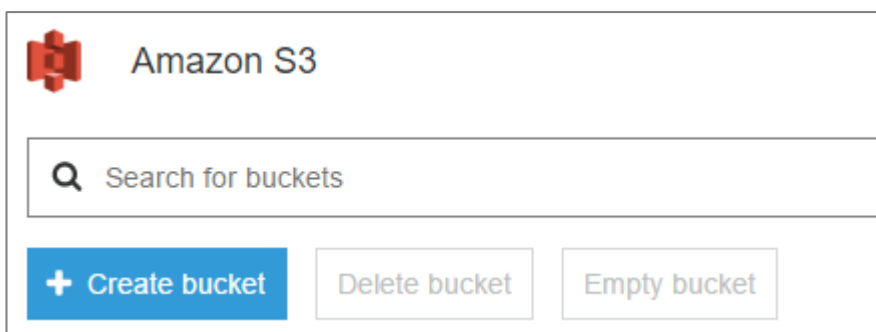
# Creating an S3 Bucket and Uploading Files

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. It has a section 'Name and region' with a 'Bucket name' field containing 'backspace-intro-aws' and a 'Region' dropdown menu set to 'US East (N. Virginia)'. Below this is a section 'Copy settings from an existing bucket' with a 'Select bucket (optional)' dropdown menu showing '2 Buckets'. At the bottom are three buttons: 'Create', 'Cancel', and 'Next'.

Leave as is and click “Next”

The screenshot shows the 'Create bucket' wizard in the AWS Management Console, Step 2: Set properties. 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, 2. Set properties (active), 3. Set permissions, and 4. Review. The main content area is dark blue. It has three sections: 'Versioning' with a description 'Keep multiple versions of an object in the same bucket.' and a 'Learn more' link, 'Logging' with a description 'Set up access log records that provide details about access requests.' and a 'Learn more' link, and 'Tags' with a description 'Use tags to track your cost against projects or other criteria.' and a 'Learn more' link. Each section has a radio button and the text 'Disabled'. At the bottom are two buttons: 'Previous' and 'Next'.

Leave as is and click “Next”

Create bucket

✓ Name and region

✓ Set properties

3 Set permissions

4 Review

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 bucket (Recommended) ▾

Manage system permissions

Do not grant Amazon S3 Log Delivery group write access to this bucket ▾

Previous

Next

Click “Create Bucket”

Create bucket

✓ Name and region

✓ Set properties

✓ Set permissions

4 Review

Name and region

Edit

Bucket name backspace-intro-aws Region US East (N. Virginia)

Properties

Edit

Versioning	Disabled
Logging	Disabled
Tagging	0 Tags

Permissions

Edit

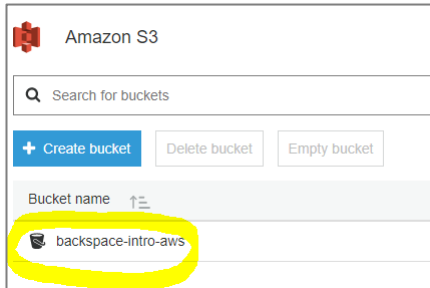
Users	1
Public permissions	Disabled
System permissions	Disabled

Previous

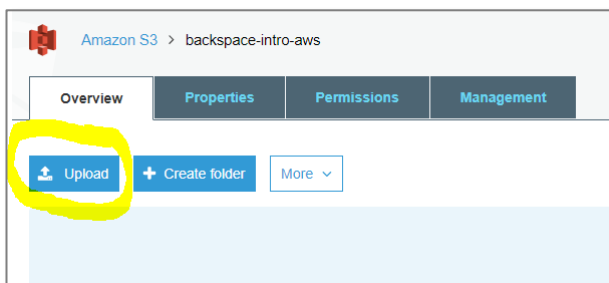
Create bucket

## Uploading Files to your Bucket

Click on the link to the bucket

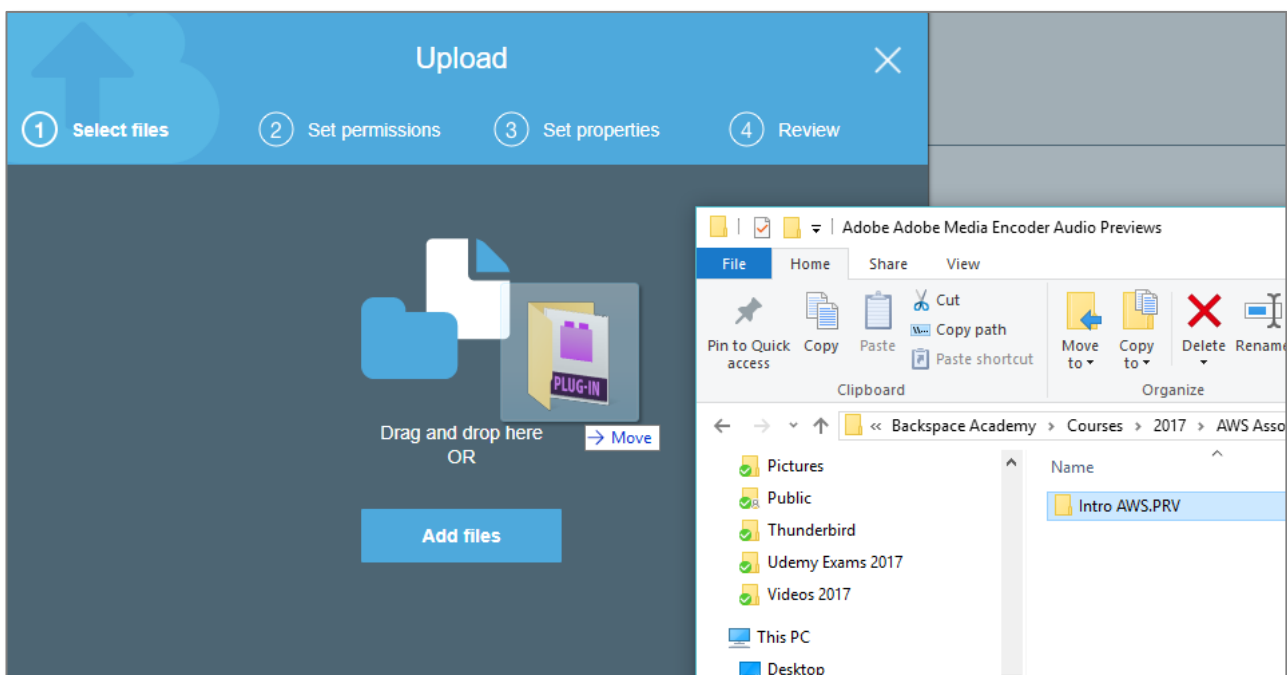


Select "Upload"



Drag a folder with files onto the form.

Click Next



Leave as is and click "Next"



**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
--------	-------

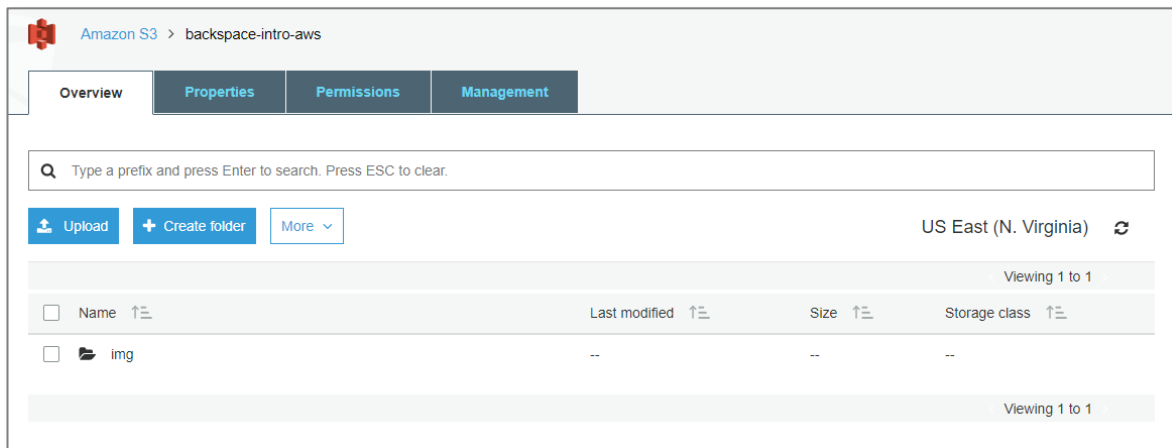
Upload Previous Next

Click “Upload”

**Upload** (0/s)

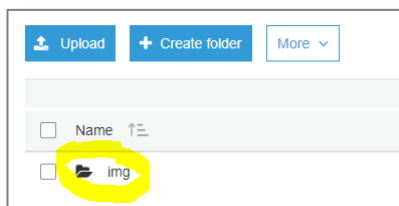
Operations 1 In progress 0 Success 0 Error

Your upload will eventually complete.

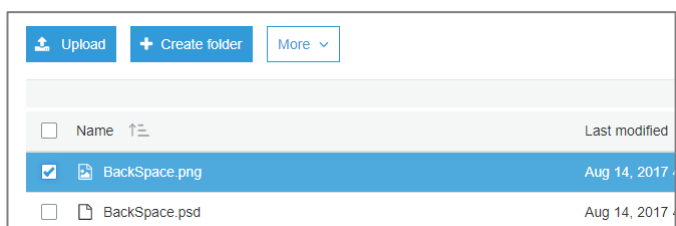


## Downloading files from your bucket

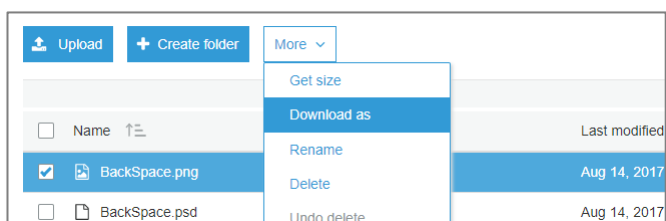
Click the link for your folder



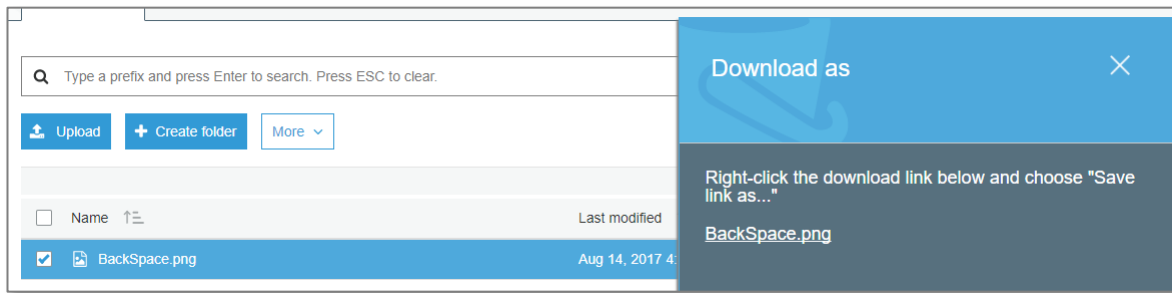
Select a file



Select "More", "Download As"



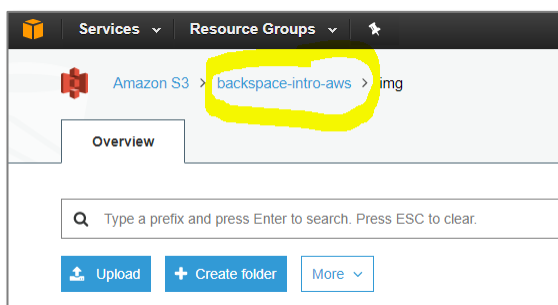
Click the download link to download the file.



## Clean Up

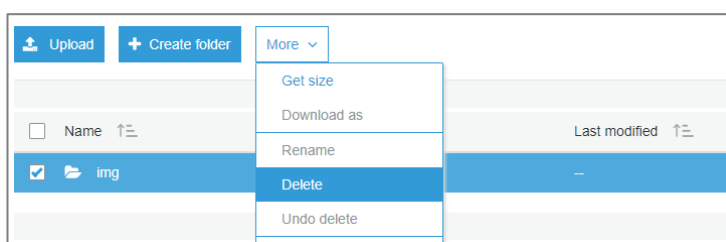
We will now delete the files and bucket so that you will not be billed by AWS.

Go back to your bucket by clicking its link.

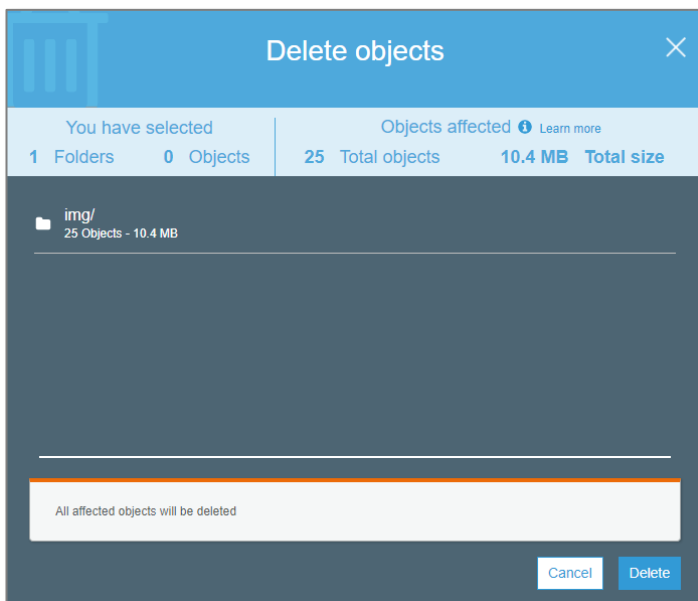


Select the folder

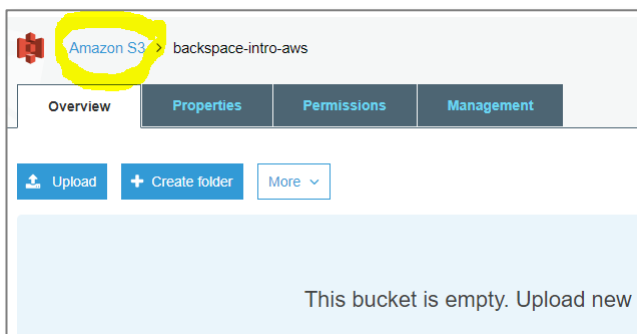
Select "More", "Delete"



Click "Delete"

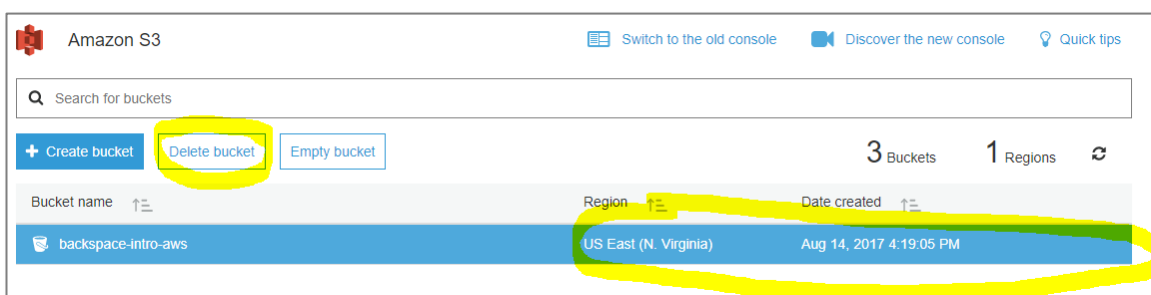


Go back to the S3 dashboard by clicking the link



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.

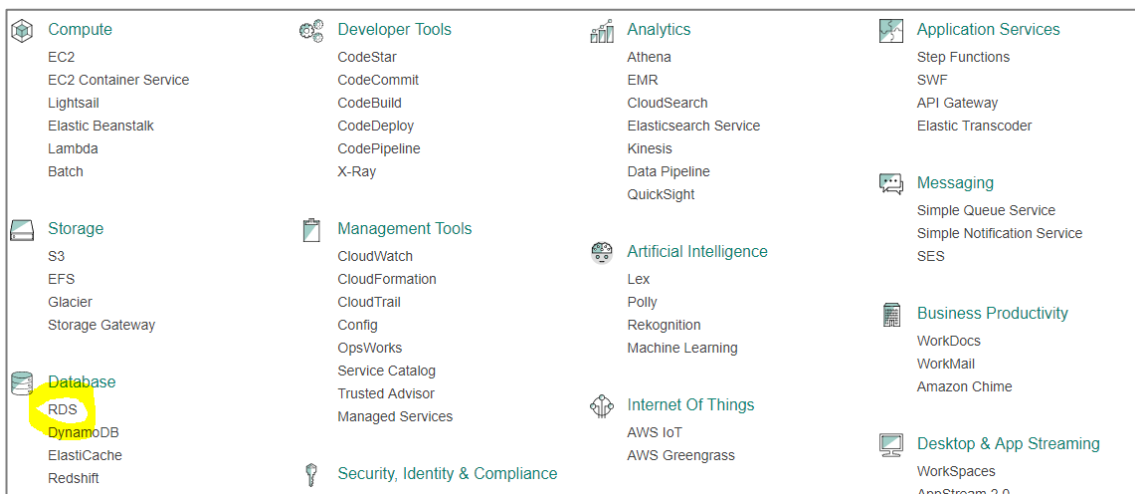
Cancel

Confirm

# Creating a SQL Database with RDS

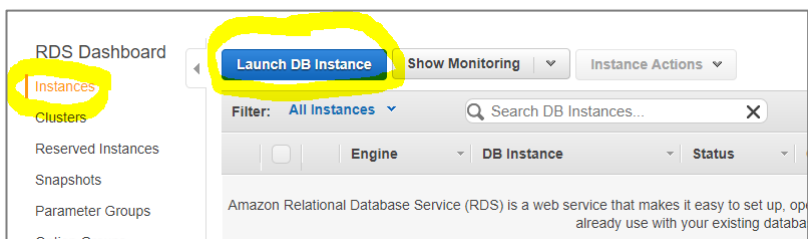
In this section, we will use the Relational Database Service to create a database. We will also connect in to the database.

From the AWS console select “RDS” from the Database services.

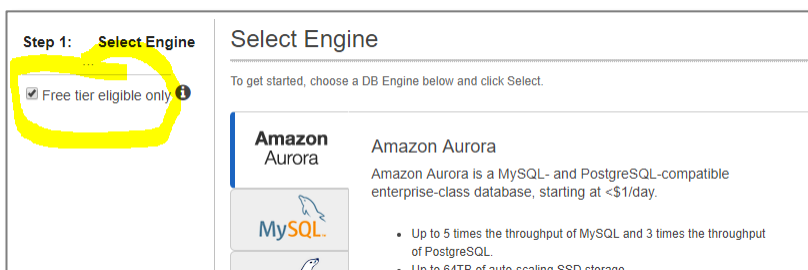


Select “instances”

Select “Launch DB Instance”



Select “Free tier eligible only”



Select the MySQL Community Edition

**Step 1: Select Engine**

☐ Free tier eligible only ⓘ

**Select Engine**

To get started, choose a DB Engine below and click Select.

**MySQL**

MySQL Community Edition

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 6 TB.
- Instances offer up to 32 vCPUs and 244 GiB Memory.
- Supports automated backup and point-in-time recovery.
- Supports cross-region read replicas.
- Free tier eligible

**Select**

Make sure “Only show options that are eligible for RDS Free Tier” is selected

**Specify DB Details**

**Free Tier**

The Amazon RDS Free Tier provides a single db.t2.micro instance as well as up to 20 GB of storage, allowing new AWS customers to gain hands-on experience with Amazon RDS. Learn more about the RDS Free Tier and the instance restrictions [here](#).

☒ Only show options that are eligible for RDS Free Tier

Select db.t2.micro instance class

**Instance Specifications**

DB Engine: mysql

License Model: general-public-license

DB Engine Version: MySQL 5.6.35

Review the [Known Issues/Limitations](#) to learn about potential compatibility issues with specific database versions.

DB Instance Class: db.t2.micro — 1 vCPU, 1 GiB RAM

Multi-AZ Deployment: No

Storage Type: General Purpose (SSD)

Allocated Storage\*: 5 GB

Give your instance a name/identifier.

Fill in a master username and password

Click “Next Step”

**Settings**

DB Instance Identifier\*: backspace-intro-aws

Master Username\*: admin

Master Password\*: .....

Confirm Password\*: .....


your AWS account in the current region. DB instance identifier is case insensitive, but stored as all lower-case, as in "mydbinstance". [Learn More](#).

\* Required

Cancel Previous **Next Step**

Leave settings for Network and Security as below.

### Configure Advanced Settings

Network & Security 

VPC\* Default VPC (vpc-72d25a0b)

Subnet Group default

Publicly Accessible Yes

Availability Zone No Preference

VPC Security Group(s) Create new Security Group

- HTTP (VPC)
- WordPress powered by Bitnami-4-8-0
- default (VPC)

Enter a database name.

Leave 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 3306

DB Parameter Group default.mysql5.6

Option Group default.mysql-5-6

Copy Tags To Snapshots ☐

Enable IAM DB Authentication No Preference

Enable Encryption No

Change “Backup Retention Period” to disable automated backups.

Click “Launch DB Instance”

### 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 0 days

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

Backup Window No Preference

### Monitoring

Enable Enhanced Monitoring No

### Maintenance

Auto Minor Version Upgrade Yes

Maintenance Window No Preference

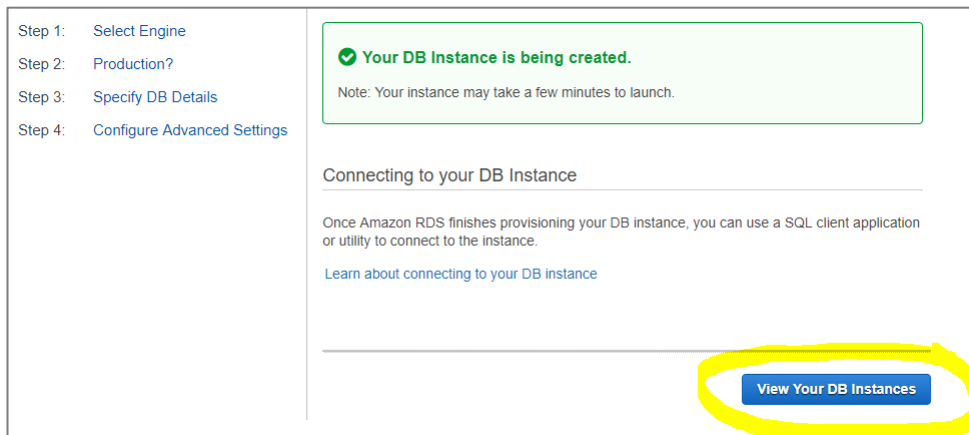
\* Required

Cancel Previous Launch DB Instance

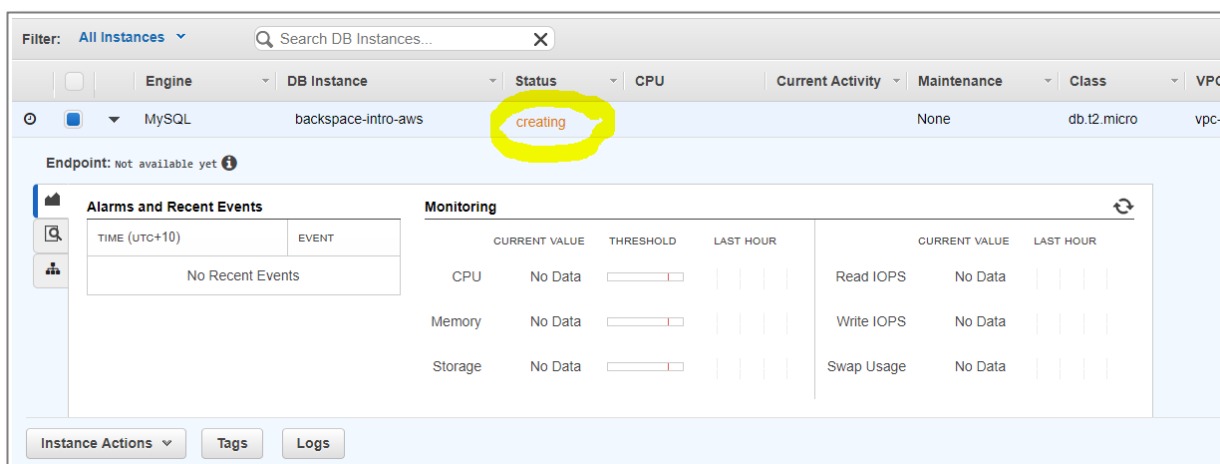
The number of days for which automated backups are retained. Setting this parameter to a positive number enables backups. Setting this parameter to 0 disables automated backups.

Click “View your DB Instances





Your instance will show status “creating”.

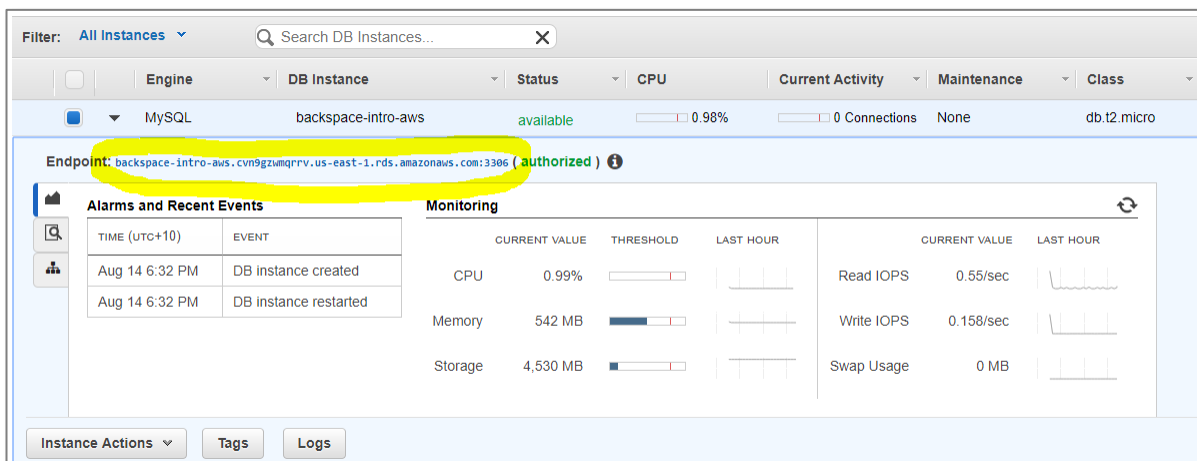


## Connecting to your RDS Instance

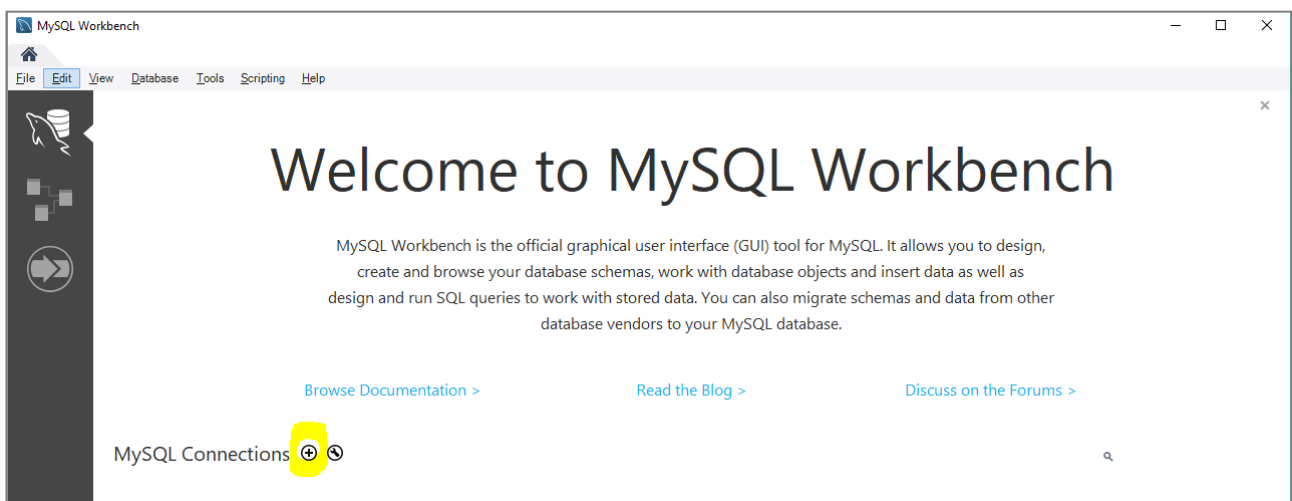
To connect to your MySQL Database you will need to download and install the MySQL Workbench from

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

When your instance status is “available”, copy the database server endpoint



Open the MySQL Workbench application click to add a new connection



Give the connection a name.

The Hostname will be the RDS server endpoint 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

Setup New Connection

Connection Name: Backspace Test Type a name for the connection

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters | SSL | Advanced

Hostname: gzwmgrv.us-east-1.rds.amazonaws.com Port: 3306 Name or IP address of the server host - and TCP/IP port.

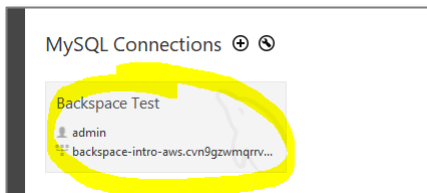
Username: admin Name of the user to connect with.

Password: Store in Vault ... Clear The user's password. Will be requested later if it's not set.

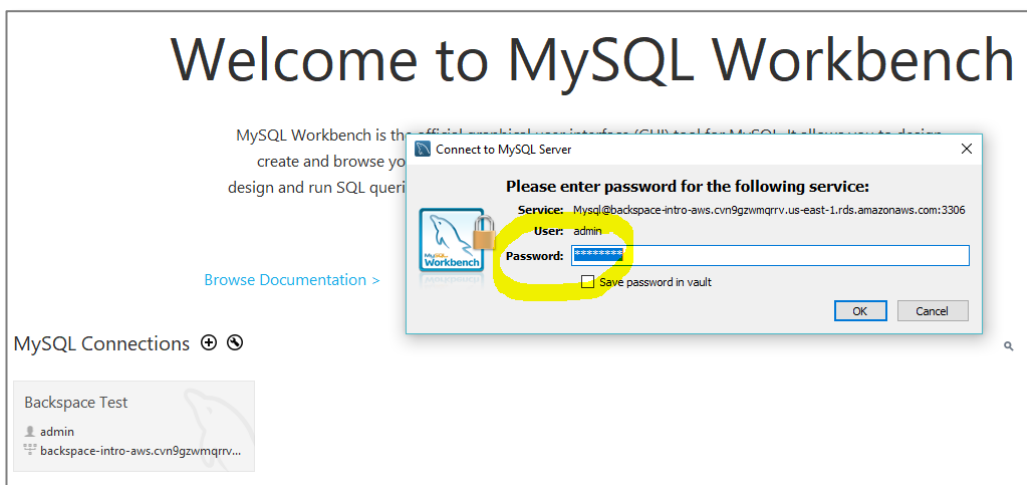
Default Schema: The schema to use as default schema. Leave blank to select it later.

Configure Server Management... Test Connection Cancel OK

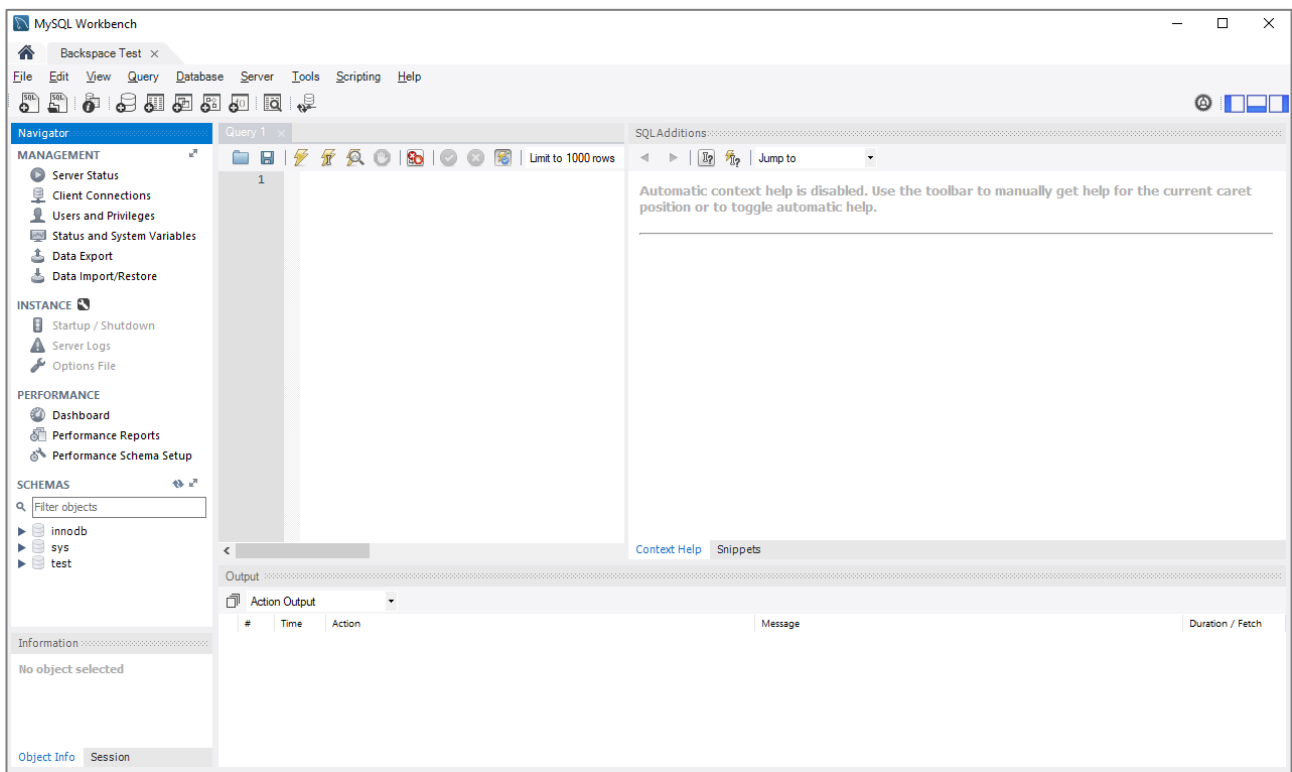
Click on the Connection



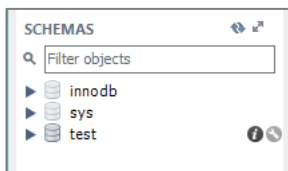
Enter the password you created in RDS for your master username



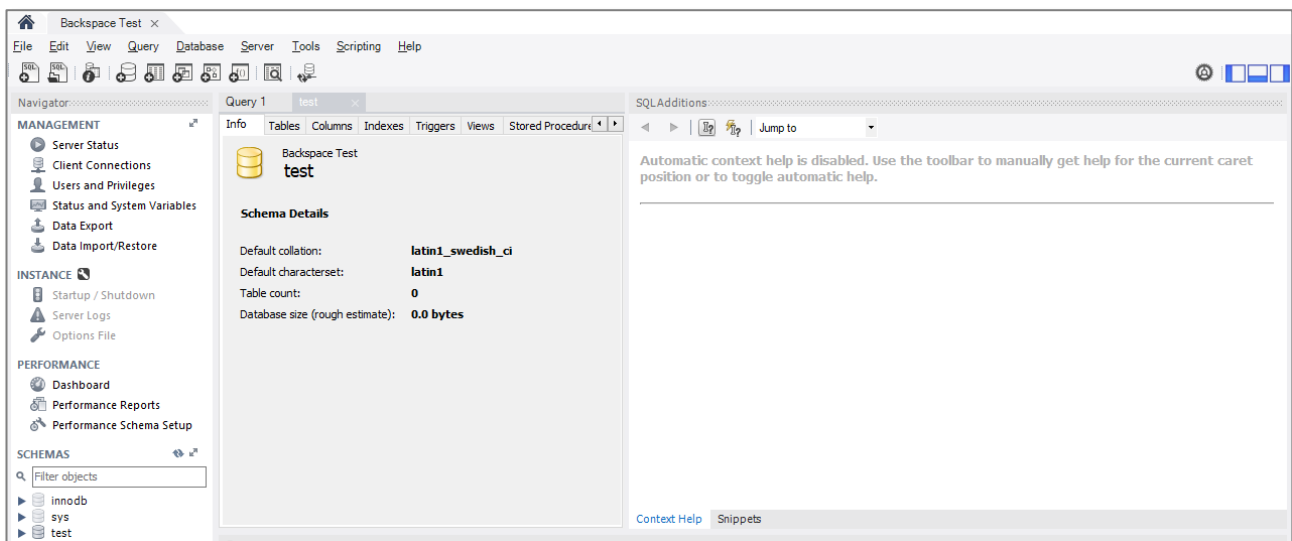
You will soon be connected to your database server



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.

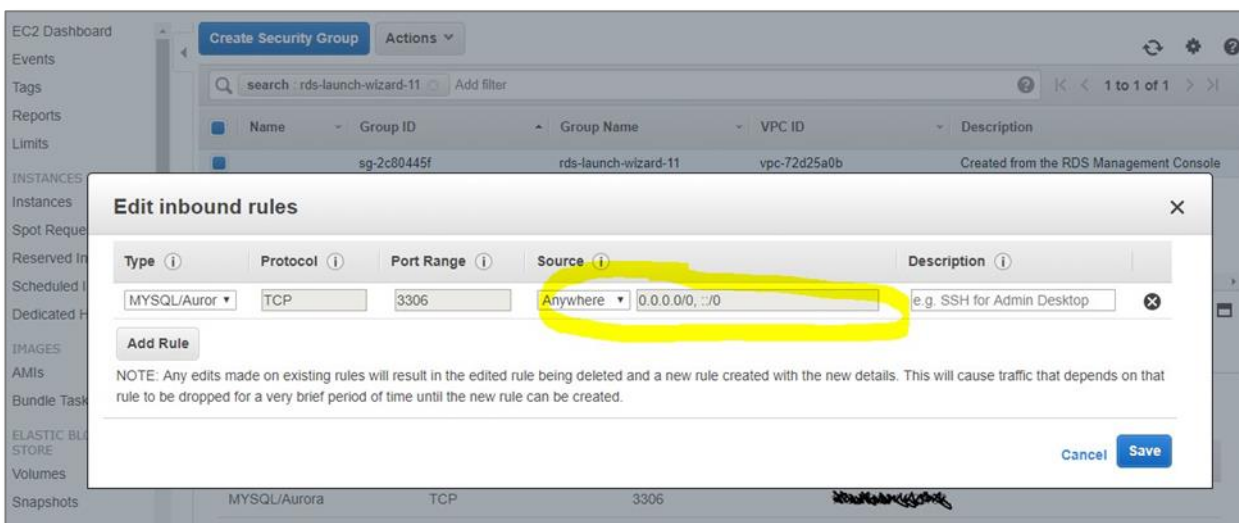


## Troubleshooting Connection Issues

If you are getting connection errors then check the following:

### Security Group Inbound Rules

The security group should 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.



### Database Username and Password

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

The screenshot shows the 'Settings' page for an AWS RDS instance. The 'DB Instance Identifier' is 'backspace-intro-aws', 'Master Username' is 'admin', 'Master Password' is masked with dots, and 'Confirm Password' is also masked with dots. A tooltip on the right explains that the DB instance identifier is case insensitive but stored as all lower-case. At the bottom, there are 'Cancel', 'Previous', and 'Next Step' buttons.

### Hostname

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

## Connecting to your RDS Instance using the Command Line

To connect to your MySQL Database using the command line you will need to download and install the MySQL

Shell from

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

Download and Unzip the file.

Go to the bin folder and run mysqlsh.exe

This will open the MySQL Shell



```

F:\Backspace Technology\Backspace Academy\Courses\2017\AWS Associate\09 - RDS\mysql-shell-1.0.10-...
MySQL Shell 1.0.10

Copyright (c) 2016, 2017, 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 '?' for help; '\quit' to exit.

Currently in JavaScript mode. Use \sql to switch to SQL mode and execute queries.
mysql-js>


```

Connect your database using the following command (if the username is admin):

```
\connect admin@your-connection-hostname-goes-here
```

Enter your password when requested.

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



```

F:\Backspace Technology\Backspace Academy\Courses\2017\AWS Associate\09 - RDS\mysql-shell-1.0.10-...
MySQL Shell 1.0.10

Copyright (c) 2016, 2017, 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 '?' for help; '\quit' to exit.

Currently in JavaScript mode. Use \sql to switch to SQL mode and execute queries.
mysql-js> \connect admin@backspace-intro-aws.cvn9gzwmqrrv.us-east-1.rds.amazonaws.com
Creating a Session to 'admin@backspace-intro-aws.cvn9gzwmqrrv.us-east-1.rds.amazonaws.com'
Enter password: *****
Your MySQL connection id is 47
Server version: 5.6.35 MySQL Community Server (GPL)
No default schema selected; type \use <schema> to set one.
mysql-js>

```

Enter SQL mode with the following command:

```
\sql
```

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

```
show databases;
```

```
mysql-js> \sql
Switching to SQL mode... Commands end with ;
mysql-sql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| innodb |
| mysql |
| performance_schema |
| sys |
| test_database |
+-----+
6 rows in set (0.22 sec)
mysql-sql>
```

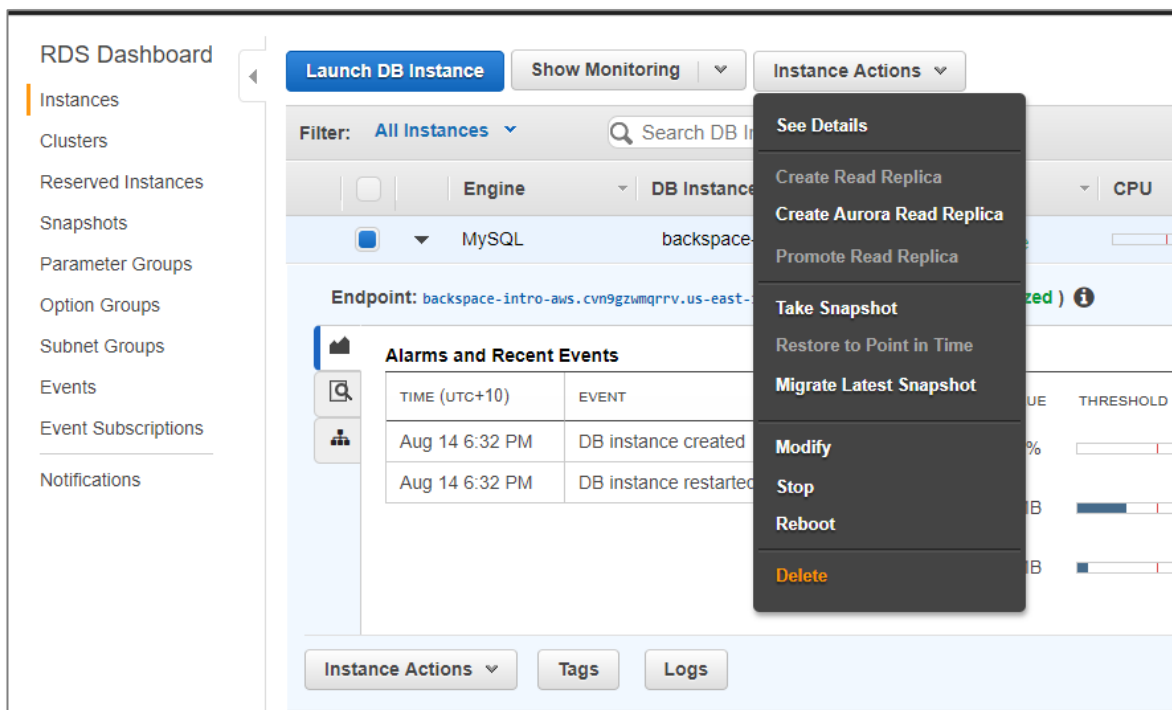
Type \quit to leave the command line

## 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”

**Delete DB Instance** ✕

Are you sure you want to Delete the **backspace-intro-aws** DB Instance?

Create final Snapshot? No ⓘ

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

 We strongly recommend taking a final snapshot before instance deletion since after your instance is deleted, automated backups will no longer be available.

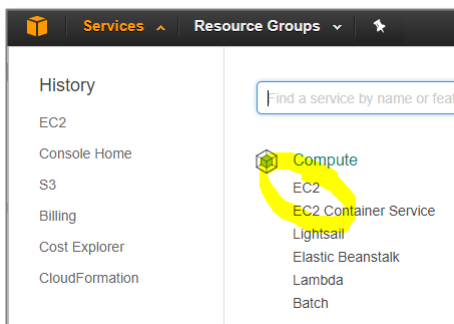
Cancel Delete



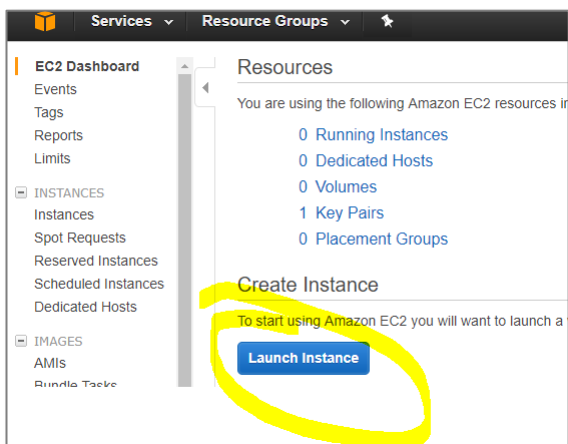
# Creating a Web Server with EC2

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

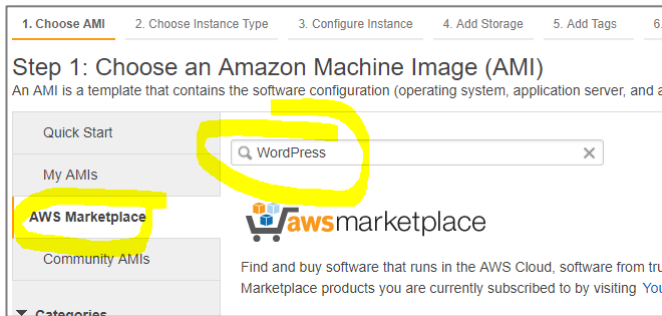
From the AWS console select “EC2” from the Compute services.



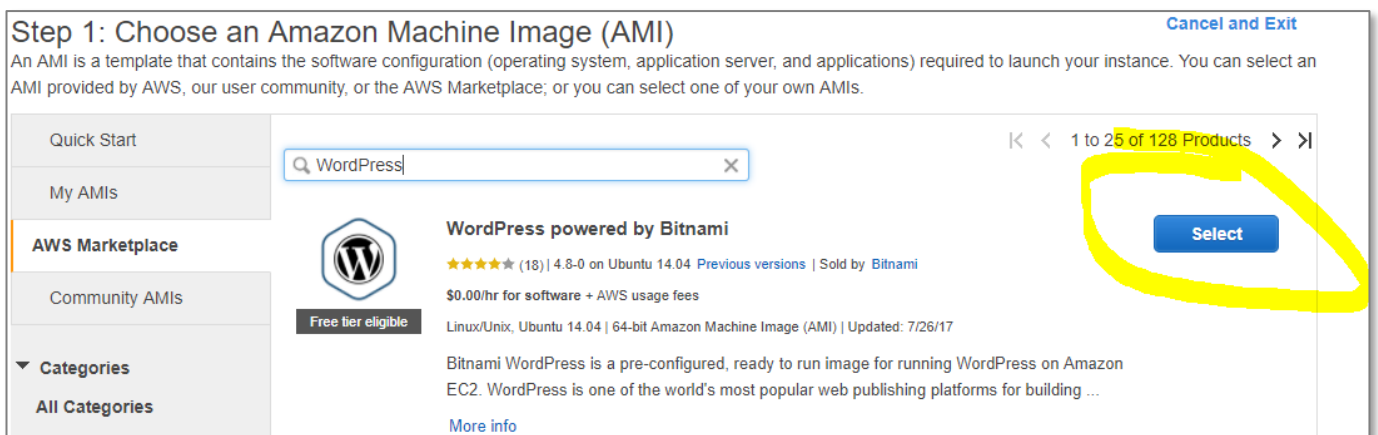
Select “Launch Instance”



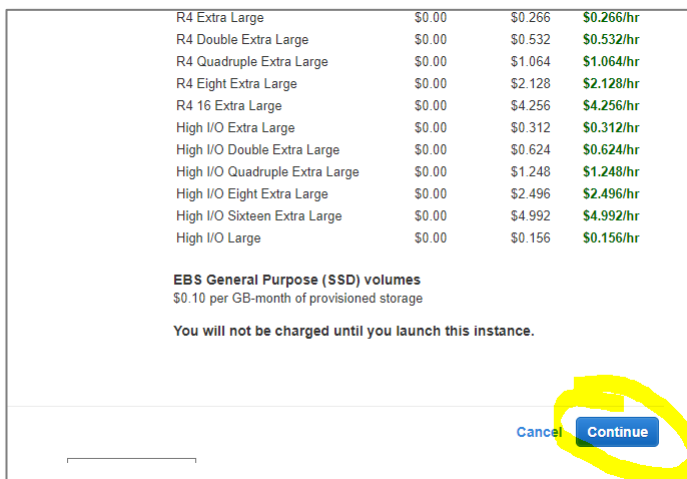
Select the “AWS Marketplace” and search for WordPress



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>

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**

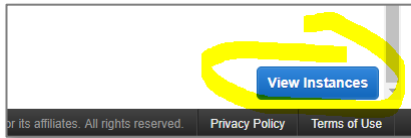
Please do not close your browser while this is loading

Creating security groups... Successful

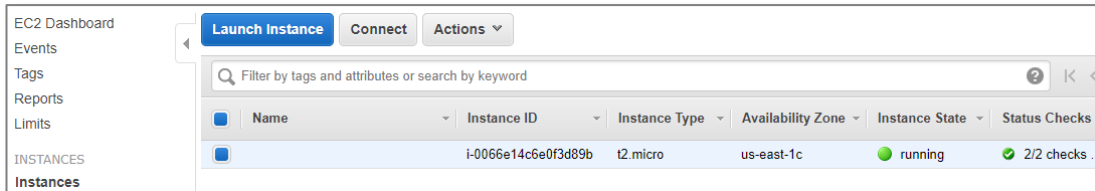
Authorizing inbound rules... Successful

Subscribing to Product...

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

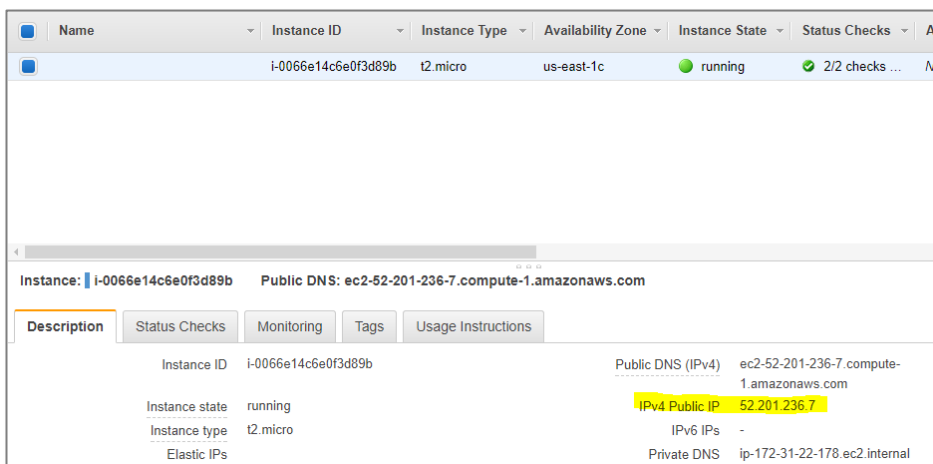


After a few minutes, the status of the instance will change to running.

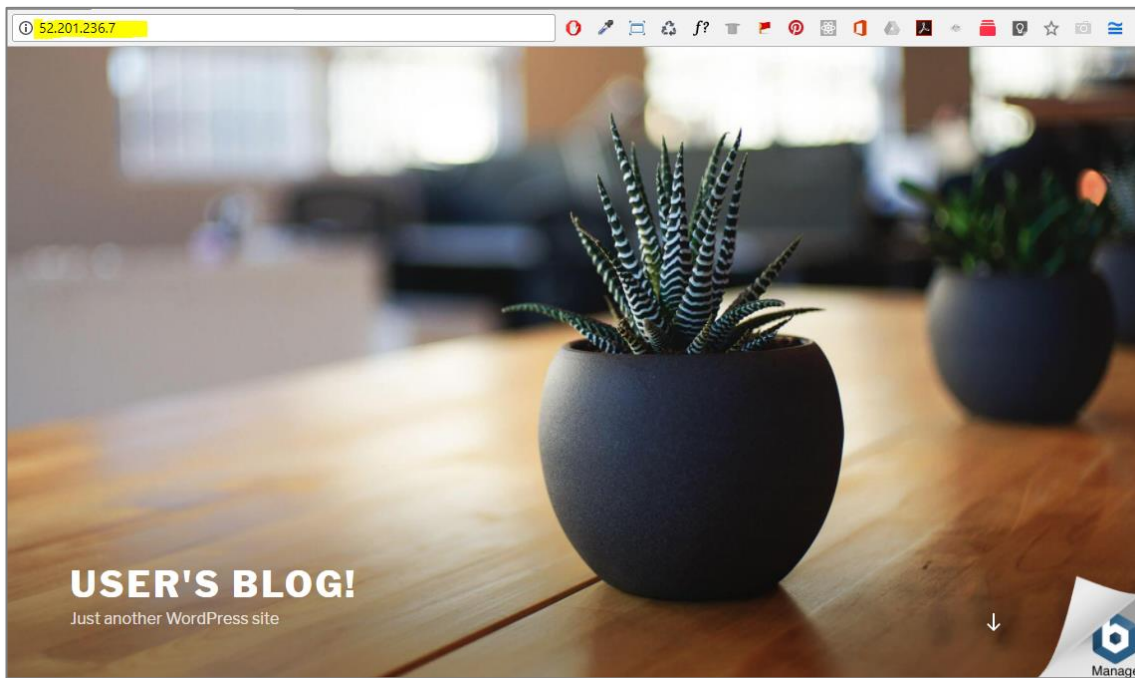


## Viewing your web server

Copy the public IP address of your web server.

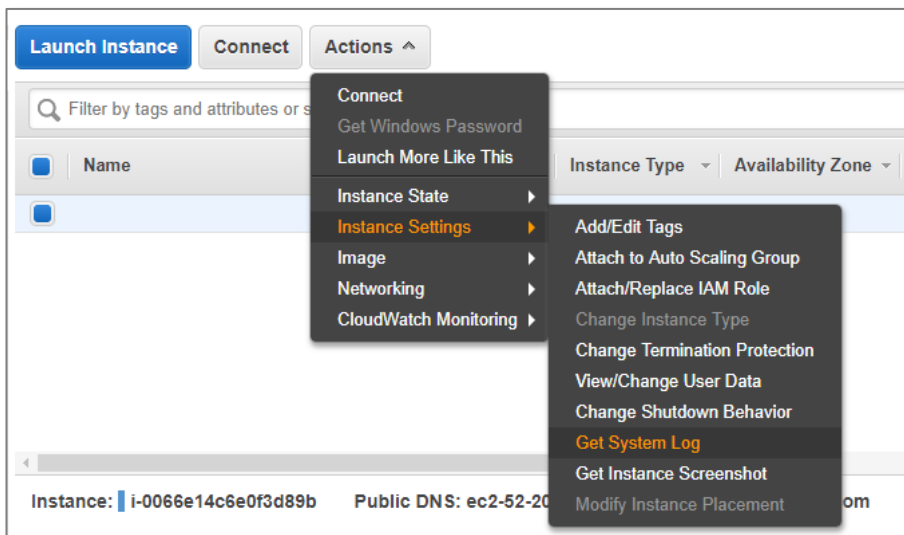


Navigate to the IP address in your browser.



## Finding the Username and Password for your WordPress application

Go back to the EC2 console and select “Instance Settings”, “Get System Log”.



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-85e2-4642e45ee7a0
micro

#####
#
#   Setting Bitnami application password to 'maIdmWey0QPS'
#
#####

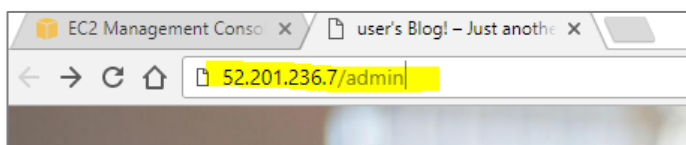
[Mon Aug 14 05:29:19 UTC 2017] Regenerating keys for wordpress
[Mon Aug 14 05:29:24 UTC 2017] Regenerating keys for wordpress finished
[Mon Aug 14 05:29:24 UTC 2017] Finished regenerating keys
[Mon Aug 14 05:29:24 UTC 2017] Setting up password for mysql service
[Mon Aug 14 05:29:44 UTC 2017] Setting up password for mysql service finished
[Mon Aug 14 05:29:44 UTC 2017] Setting up password for wordpress application
[Mon Aug 14 05:29:50 UTC 2017] Setting up password for wordpress application finished
[Mon Aug 14 05:29:50 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 compatibility[746[ 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: -----BEGIN SSH HOST KEY FINGERPRINTS-----

```

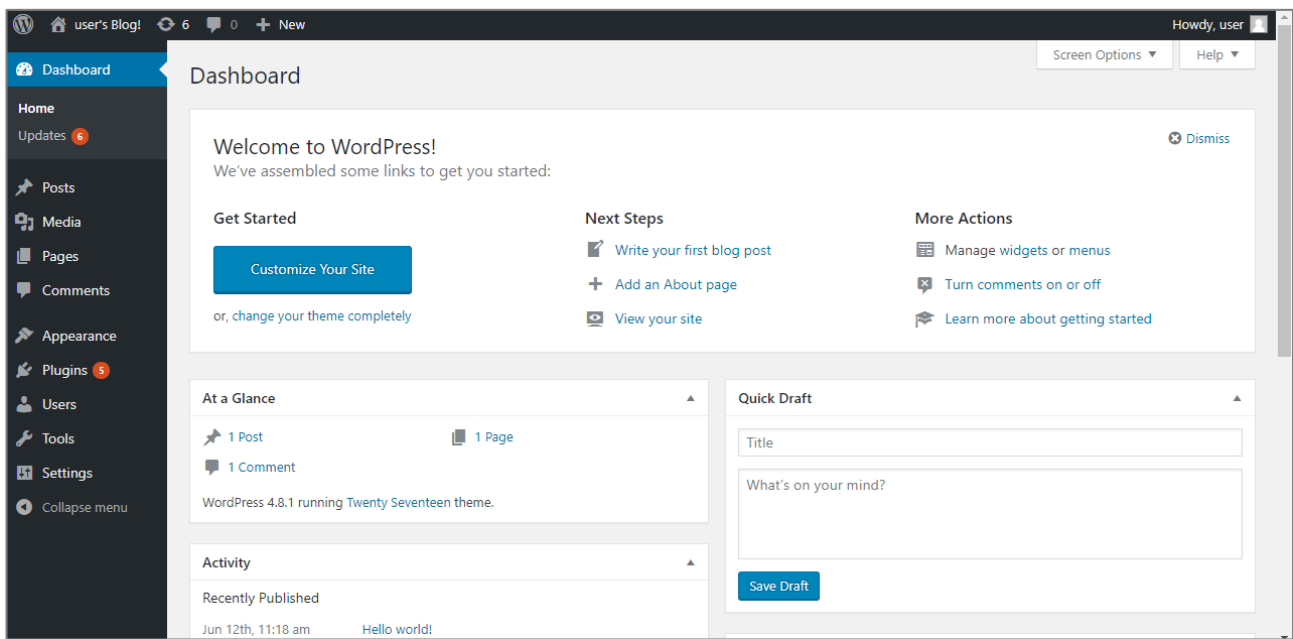
Close

Go to the admin subdirectory of your website in your browser



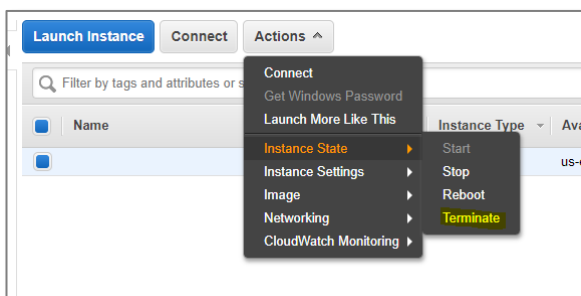
Enter Username "user" and paste in the password

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

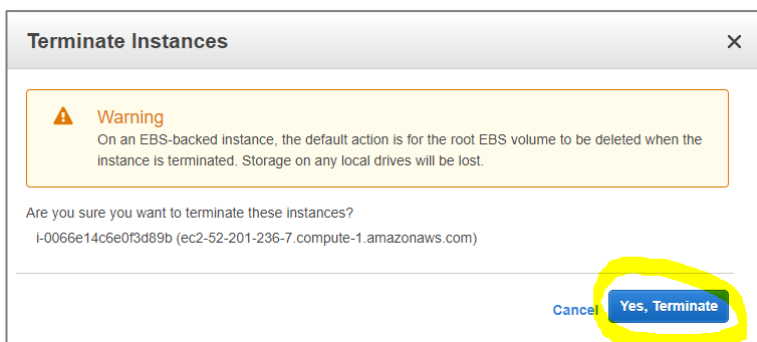


## Clean up

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



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

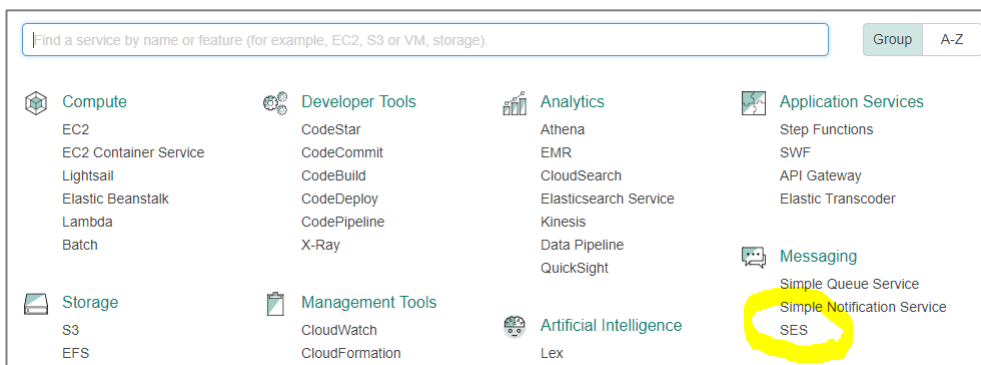




# Sending emails with Amazon SES

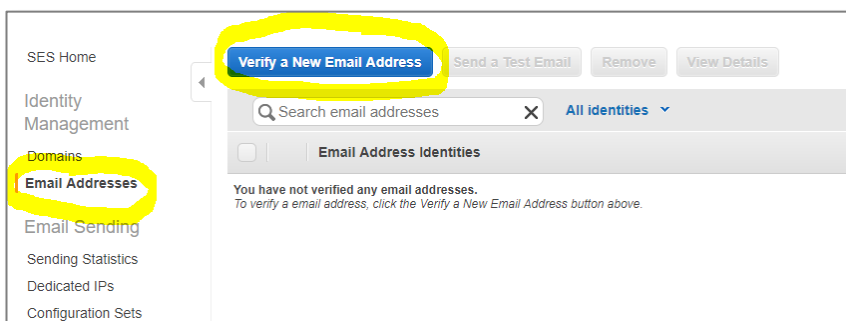
In this section, we will use the Simple Email Service to send an email.

From the AWS console select “SES” from the Messaging services.

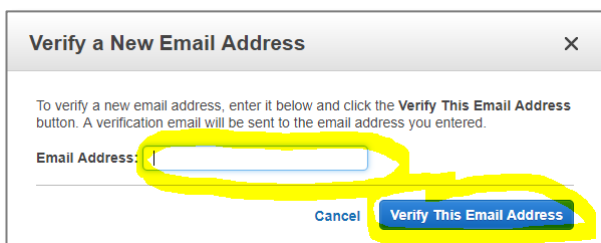


Click on “Email addresses”

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.

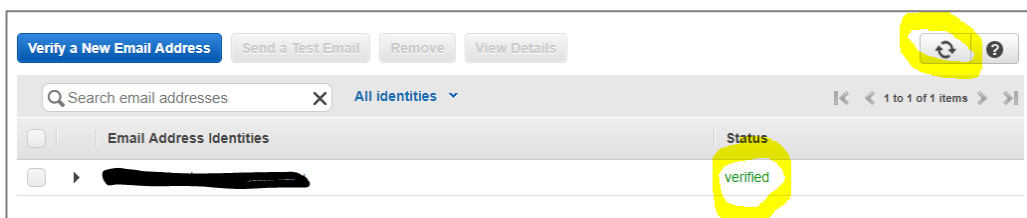
**For new Amazon SES users**—If you have not yet applied for a sending limit increase, then you are still in the [sandbox environment](#), and you can only send email to addresses that have been verified. To verify a new email address or domain, see the **Identity Management** section of the [Amazon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

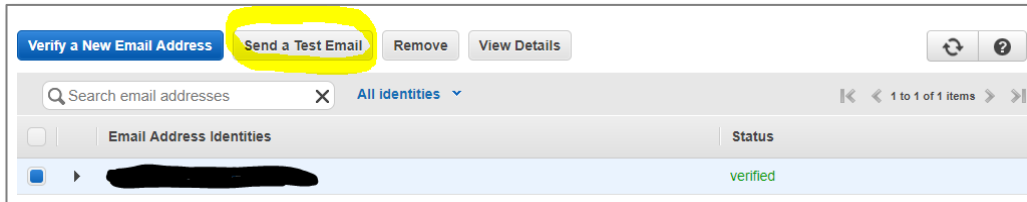
If you have already been approved for a sending limit increase, then you can start sending email to non-verified addresses.

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. The 'Send Test Email' button is highlighted with a yellow circle.

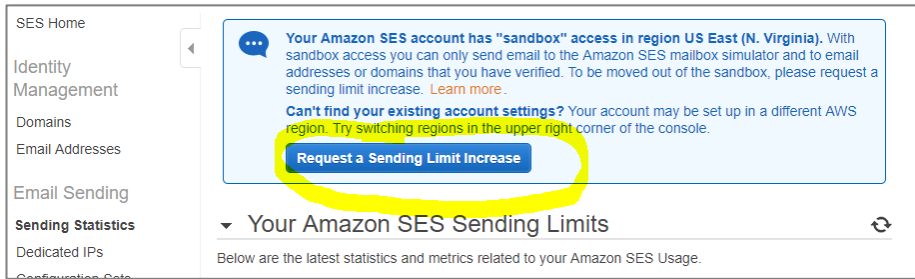
Check your email to see if it worked.

## Requesting full access to SES

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

Click on "Sending Statistics"

Click on "Request a Sending Limit Increase"

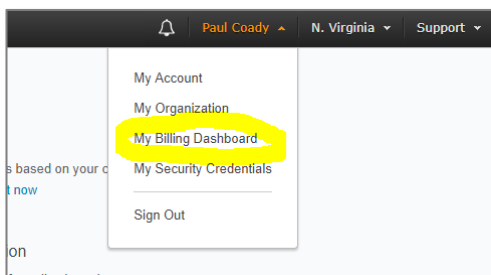


# Creating a Billing Alert with CloudWatch and SNS

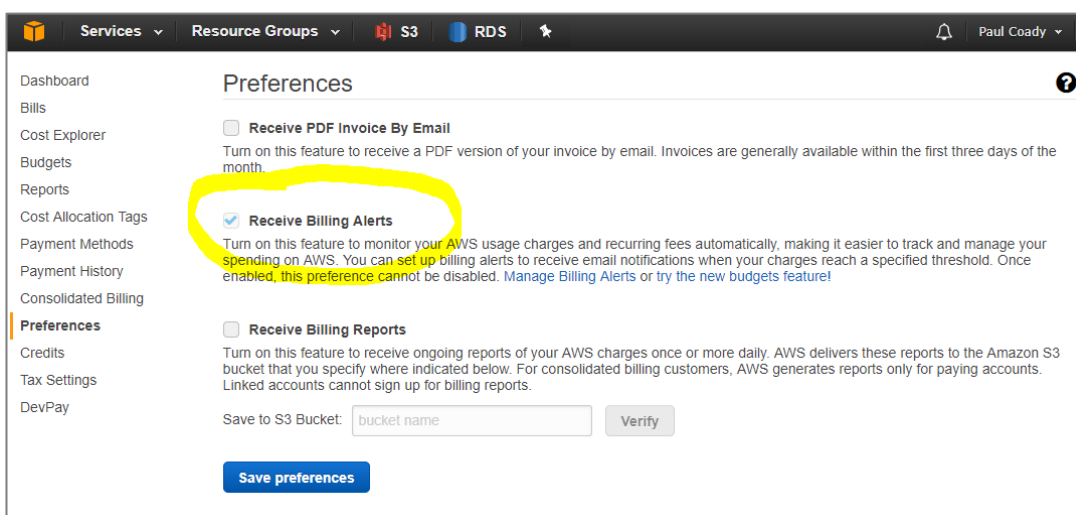
In this section, we will create a CloudWatch billing alert that will send an email through the Simple Notification Service whenever our estimated monthly bill exceeds a certain level.

## Enabling Billing Alerts

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

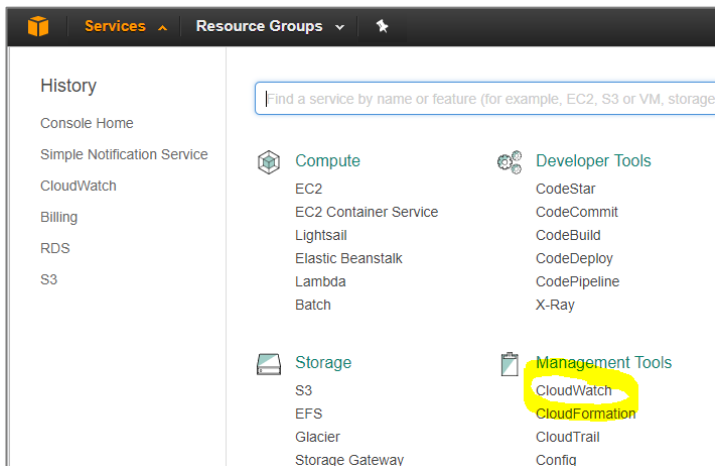


Select “Preferences” and check “Receive Billing Alerts”

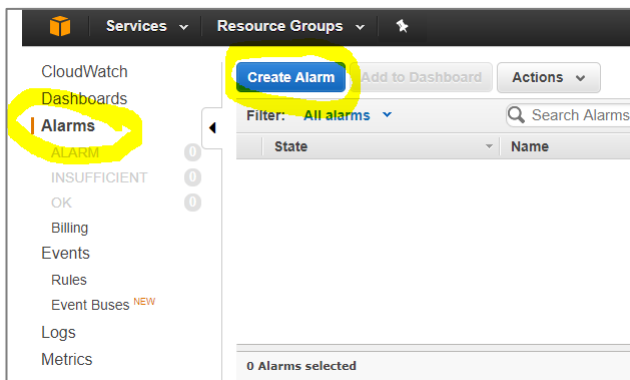


## Creating a CloudWatch Alarm

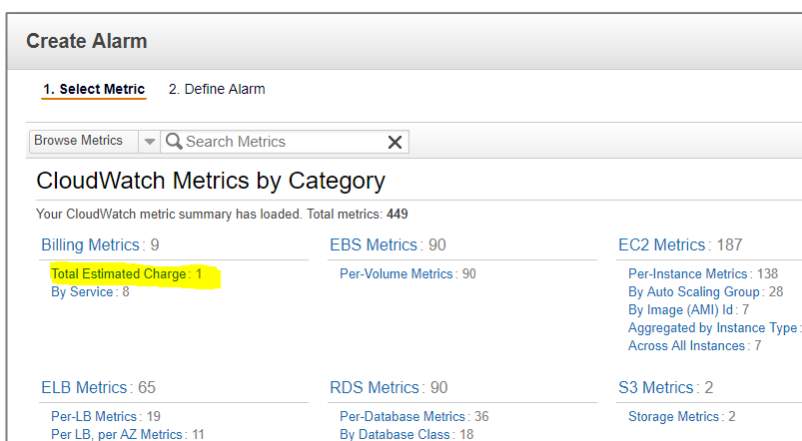
Click the Services menu and select “CloudWatch” from “Management Tools”



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 > Total Estimated Charge

Currency: **USD** Metric Name: EstimatedCharges

Title: EstimatedCharges Maximum 6 Hours

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

Namespace: AWS/Billing

Currency: USD

Metric Name: EstimatedCharges

Next Create Alarm

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

Next Create Alarm

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

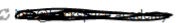
Click Create Alarm.

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

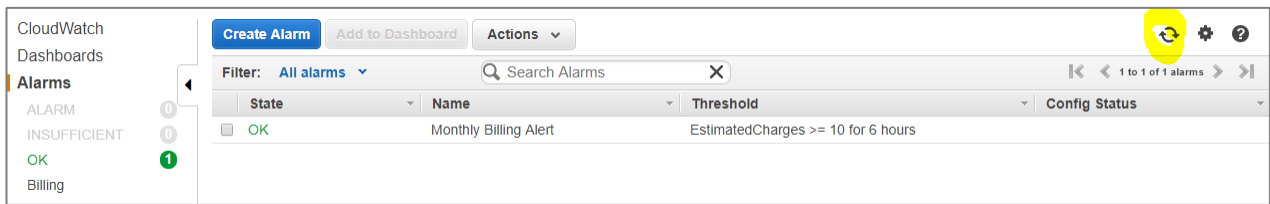
**Subscription confirmed!**

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

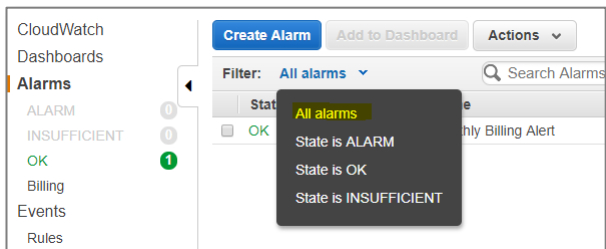
Your subscription's id is:  
arn:aws:sns:us-east-1:950302654420:monthly-billing-alert:69686c20-8f1d-472d-8ae2-37242a448d9a

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.

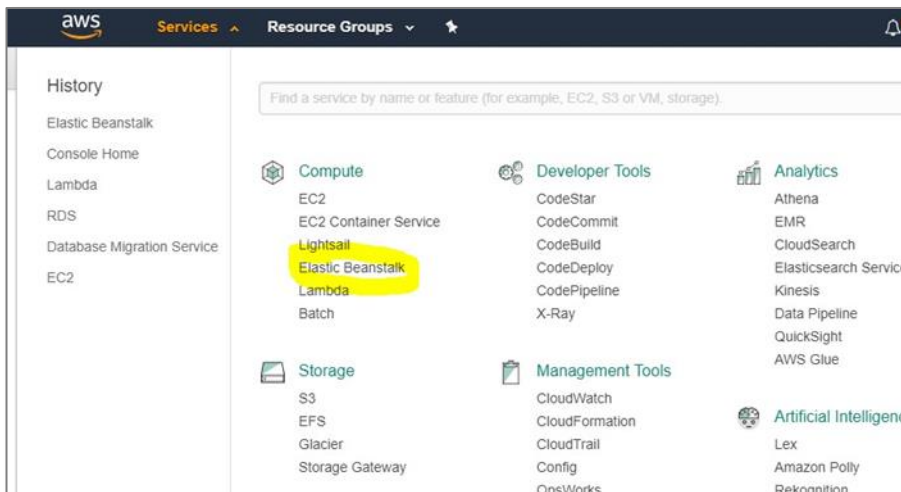




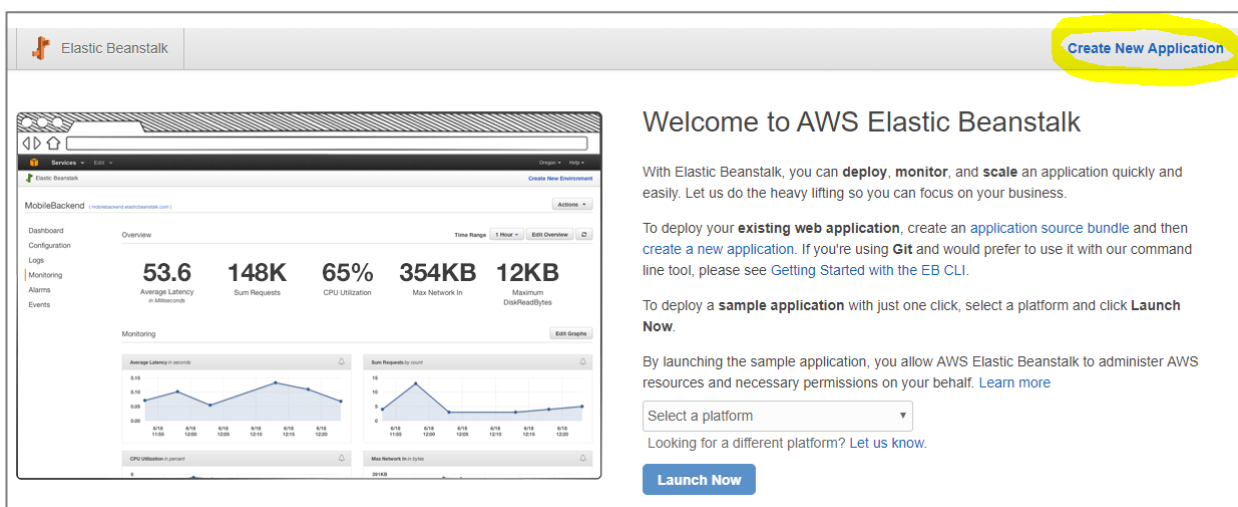
# Creating a Highly Available Architecture with Elastic Beanstalk

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*

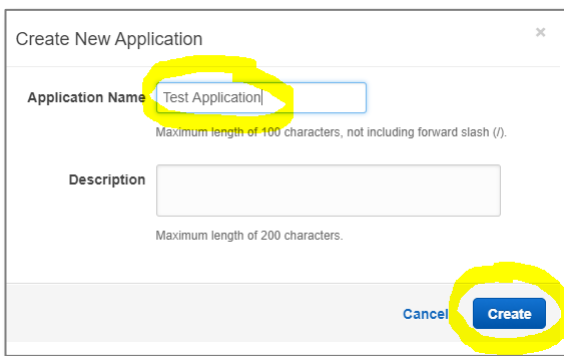


Click "Create New Application"



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

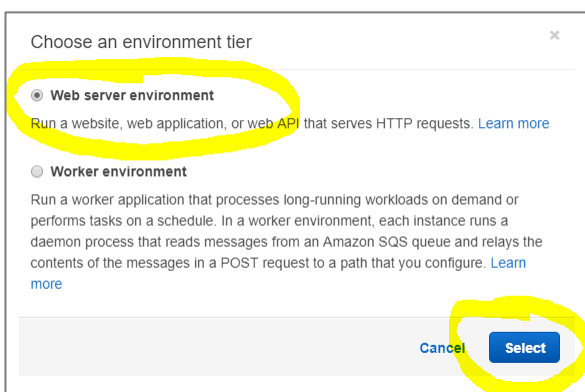
No environments currently exist for this application. [Create one now.](#)

Actions

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

Select “Web server environment”

Click “Select”



Choose an environment tier

☒ Web server environment

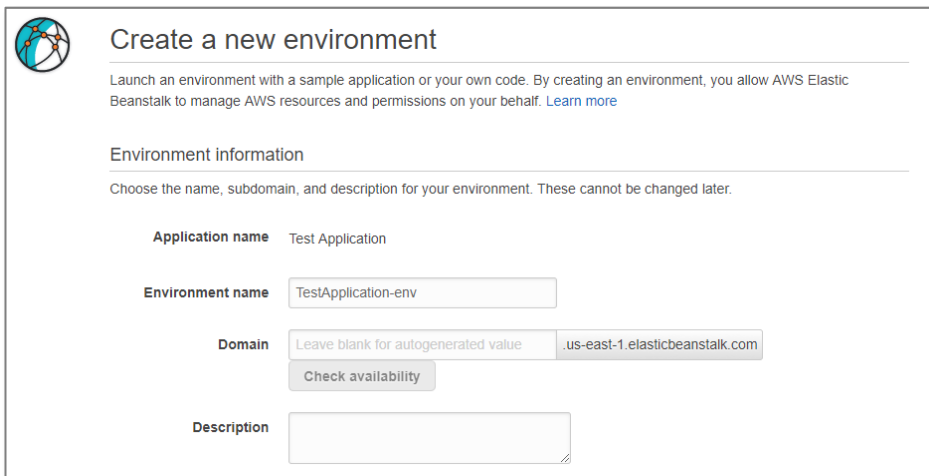
Run a website, web application, or web API that serves HTTP requests. [Learn more](#)

☐ 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**

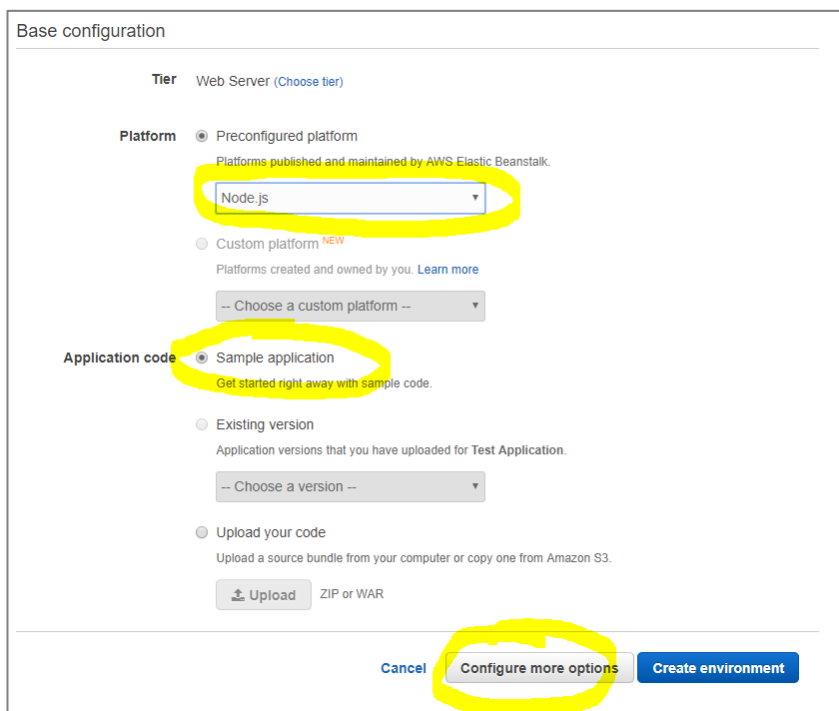
**Domain**

**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*



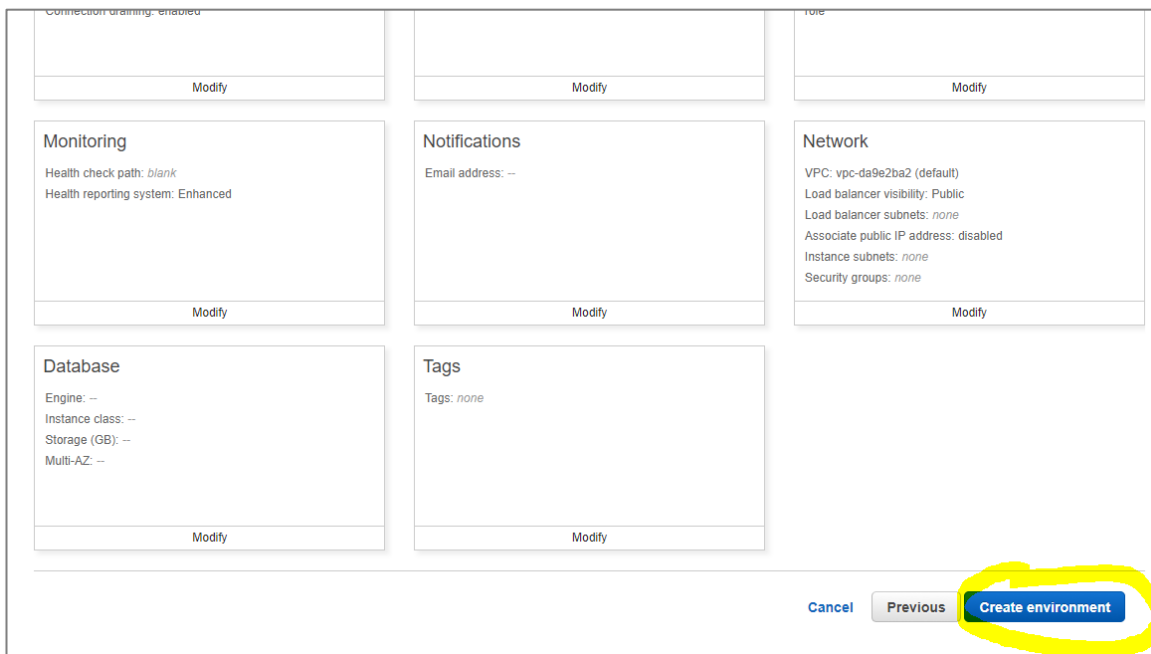
**Configure TestApplication-env**

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

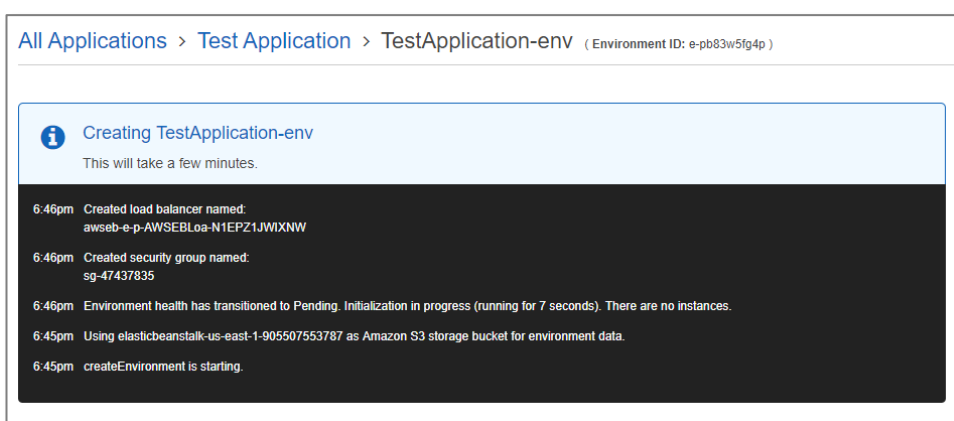
**Tags**

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](#)) 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](#)

46