Compute

Launch a Linux Virtual Machine

using Amazon EC2

Launch a Windows Virtual Machine

with Amazon EC2

Run a Serverless "Hello, World!"

with AWS Lambda

Deploy Docker Containers

on Amazon EC2 Container Service

Websites & Web Apps

Launch a WordPress Website

with Amazon EC2 and AWS Marketplace

Register a Domain Name

with Amazon Route 53

Launch a Web Application

with AWS Elastic Beanstalk

Update a Web Application

with AWS Elastic Beanstalk

Storage & Content Delivery

Store and Retrieve a File

with Amazon S3

Create a Network File System

with Amazon EFS

Deliver Content Faster

with Amazon CloudFront

Database

Create and Connect to a MySQL Database

with Amazon RDS

Create and Query a NoSQL Table

with Amazon Dynamo DB

Create and Connect to a Microsoft SQL Server Database

with Amazon RDS

Create and Connect to a MariaDB Database

with Amazon RDS

Create and Connect to a PostgreSQL Database

with Amazon RDS

Deploy Code to a Virtual Machine

with AWS CodeDeploy

Set up a Continuous Deployment Pipeline

with AWS CodePipeline

Send an Email

with Amazon SES

Send Messages Between Distributed Applications

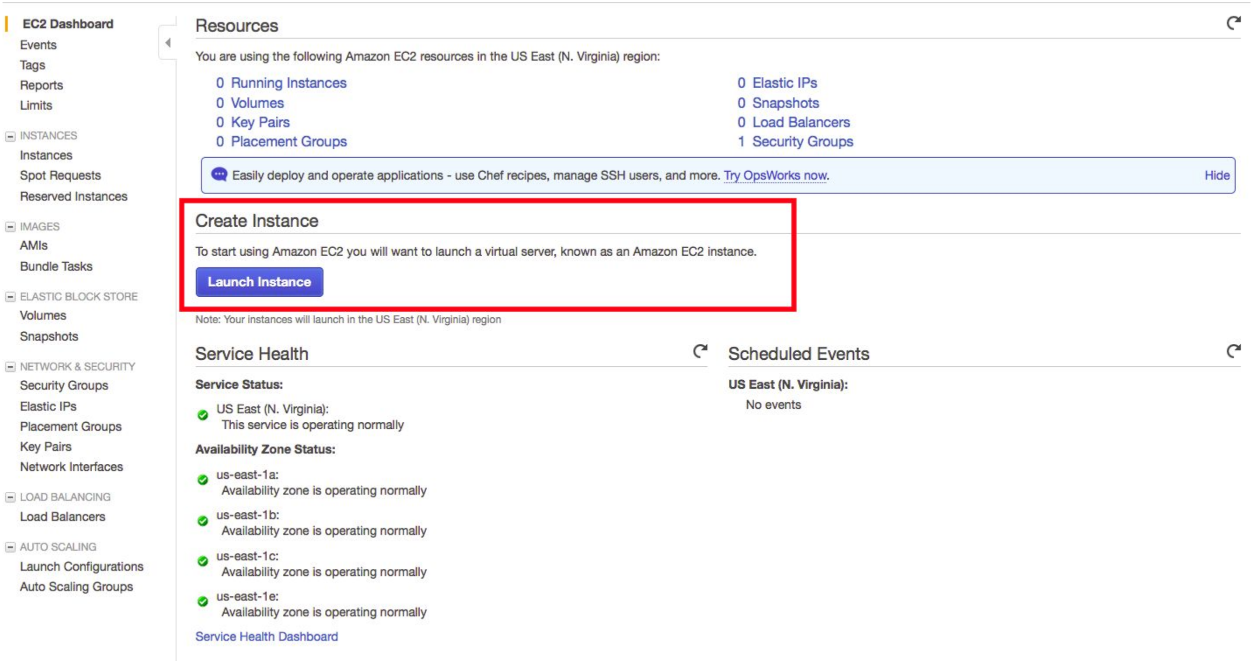
with Amazon SQS

Launch a Linux Virtual Machine

Amazon Elastic Compute Cloud (EC2) is the Amazon Web Service you use to create and run virtual machines in the cloud (we call these virtual machines 'instances'). This step-by-step guide will help you successfully launch a Linux virtual machine on Amazon EC2 within our AWS Free Tier.

Step 1: Launch an Amazon EC2 Instance

a. Click here to open the Amazon EC2 console and then click **Launch Instance** to create and configure your virtual machine.



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

0 Running Instances
0 Volumes
0 Key Pairs
0 Placement Groups
0 Elastic IPs
0 Snapshots
0 Load Balancers
1 Security Groups

Easily deploy and operate applications - use Chef recipes, manage SSH users, and more. Try OpsWorks now.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (N. Virginia) region

Service Health

Service Status:

US East (N. Virginia):
This service is operating normally

Availability Zone Status:

us-east-1a:
Availability zone is operating normally

us-east-1b:
Availability zone is operating normally

us-east-1c:
Availability zone is operating normally

us-east-1e:
Availability zone is operating normally

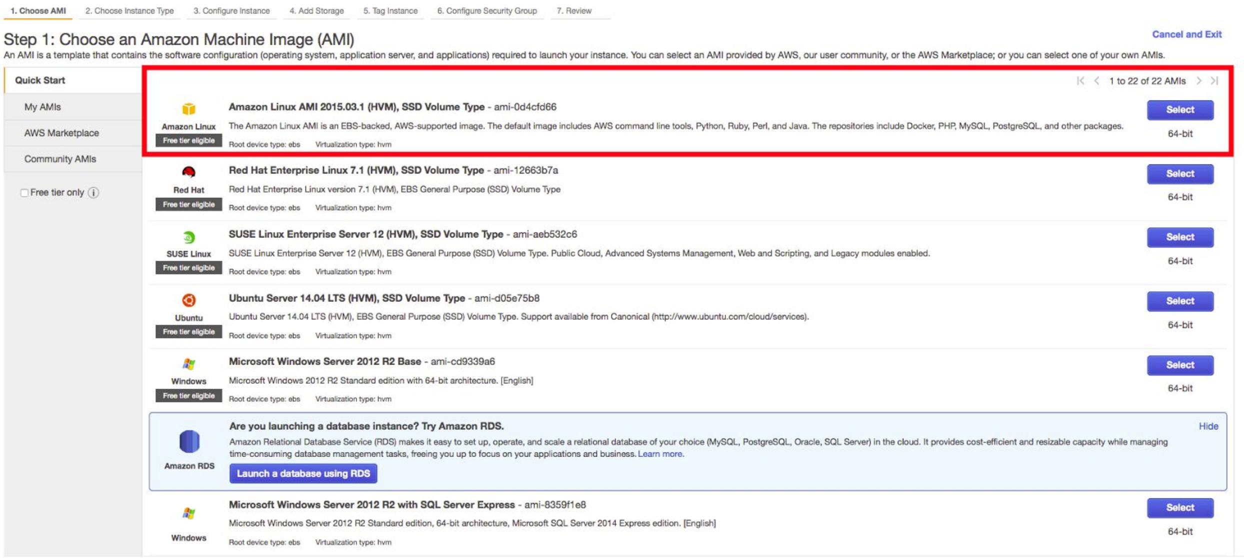
Scheduled Events

US East (N. Virginia):
No events

Step 2: Configure your Instance

You are now in the EC2 Launch Instance Wizard, which will help you configure and launch your instance.

a. With Amazon EC2, you can specify the software and specifications of the instance you want to use. In this screen, you are shown options to choose an Amazon Machine Image (AMI), which is a template that contains the software configuration (e.g. an operating system, an application server, and applications). From an AMI, you launch an instance, which is a copy of the AMI running as a virtual server in the cloud.
For this tutorial, find *Amazon Linux AMI* and click **Select**



Step 1: Choose an Amazon Machine Image (AMI)

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

Amazon Linux — Free tier eligible
Amazon Linux AMI 2015.03.1 (HVM), SSD Volume Type - ami-0d4cfd66
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root device type: ebs   Virtualization type: hvm
64-bit

Red Hat — Free tier eligible
Red Hat Enterprise Linux 7.1 (HVM), SSD Volume Type - ami-12663b7a
Red Hat Enterprise Linux version 7.1 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs   Virtualization type: hvm
64-bit

SUSE Linux — Free tier eligible
SUSE Linux Enterprise Server 12 (HVM), SSD Volume Type - ami-aeb532c6
SUSE Linux Enterprise Server 12 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.
Root device type: ebs   Virtualization type: hvm
64-bit

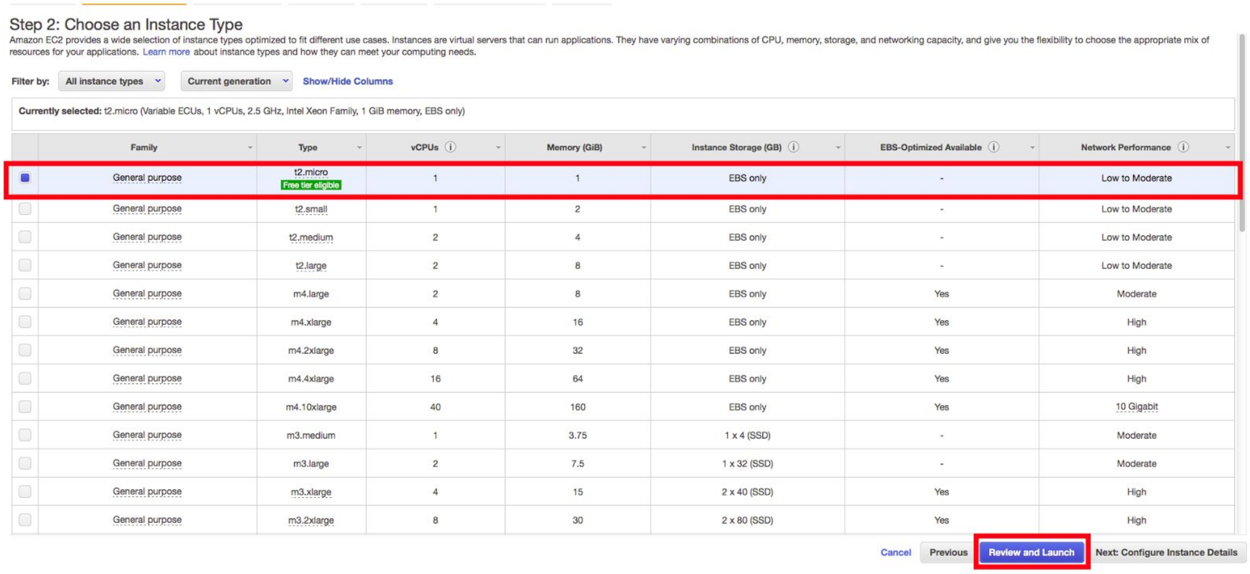
Ubuntu — Free tier eligible
Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-d05e75b8
Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root device type: ebs   Virtualization type: hvm
64-bit

Windows — Free tier eligible
Microsoft Windows Server 2012 R2 Base - ami-cd9339a6
Microsoft Windows 2012 R2 Standard edition with 64-bit architecture. [English]
Root device type: ebs   Virtualization type: hvm
64-bit

Amazon RDS
Are you launching a database instance? Try Amazon RDS.
Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database of your choice (MySQL, PostgreSQL, Oracle, SQL Server) in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database management tasks, freeing you up to focus on your applications and business. Learn more.

Windows
Microsoft Windows Server 2012 R2 with SQL Server Express - ami-8359f1e8
Microsoft Windows Server 2012 R2 Standard edition, 64-bit architecture, Microsoft SQL Server 2014 Express edition. [English]
Root device type: ebs   Virtualization type: hvm
64-bit

b. You will now choose an instance type. Instance types comprise of varying combinations of CPU, memory, storage, and networking capacity so you can choose the appropriate mix for your applications. For more information, see Amazon EC2 Instance Types.
The default option of *t2.micro* should already be checked. This instance type is covered within the Free Tier and offers enough compute capacity to tackle simple workloads. Click Review and Launch at the bottom of the page.



Step 2: Choose an Instance Type

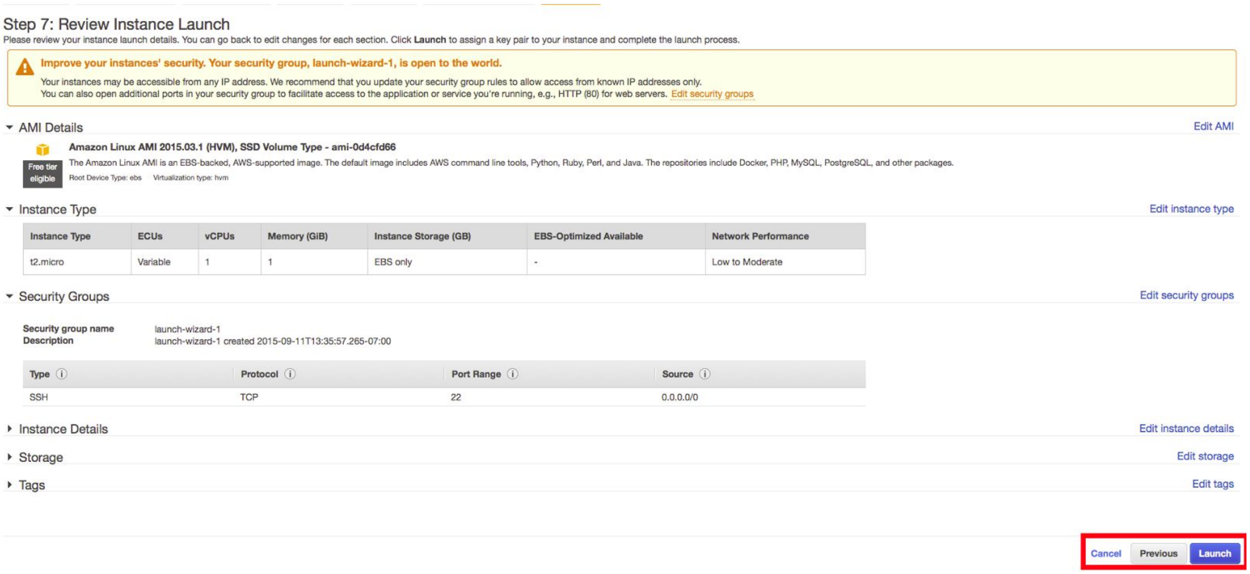
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types | Current generation | Show/Hide Columns

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

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
□	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
□	General purpose	m4.large	2	8	EBS only	Yes	Moderate
□	General purpose	m4.xlarge	4	16	EBS only	Yes	High
□	General purpose	m4.2xlarge	8	32	EBS only	Yes	High
□	General purpose	m4.4xlarge	16	64	EBS only	Yes	High
□	General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit
□	General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate
□	General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate
□	General purpose	m3.xlarge	4	15	2 x 40 (SSD)	Yes	High
□	General purpose	m3.2xlarge	8	30	2 x 80 (SSD)	Yes	High

c. You can review the configuration, storage, tagging, and security settings that have been selected for your instance. While you have the option to customize these settings, we recommend accepting the default values for this tutorial.
Click Launch at the bottom of the page.



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, launch-wizard-1, 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

Amazon Linux AMI 2015.03.1 (HVM), SSD Volume Type - ami-0d4cfd66
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Free tier eligible
Root Device Type: ebs Virtualization type: hvm

Instance Type — Edit instance type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups — Edit security groups

Security group name: launch-wizard-1
Description: launch-wizard-1 created 2015-09-11T13:35:57.265-07:00

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0/0

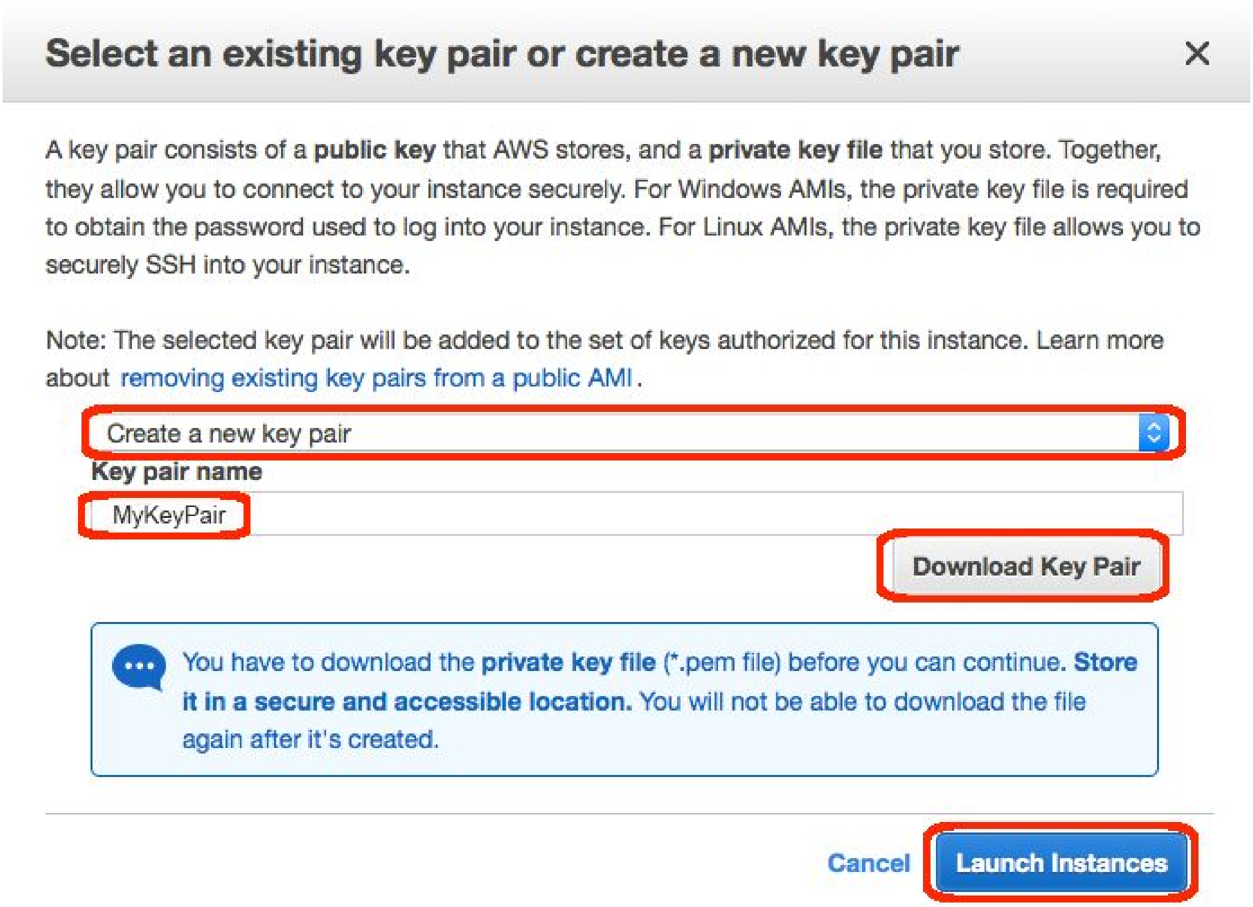
Instance Details — Edit instance details

Storage — Edit storage

Tags — Edit tags

d. On the next screen you will be asked to choose an existing key pair or create a new key pair. A key pair is used to log into your instance (just like your house key is used to enter your home). Select Create a new key pair and give it the name MyKeyPair. Next click the Download Key Pair button.

Be sure to save the key pair in a safe location on your computer. If you don't remember where you store your SSH private key (the file you are downloading), you won't be able to connect to your virtual machine.
Windows users: We recommend saving your key pair in your user directory in a sub-directory called .ssh (ex. *C:\user\{yourusername}\.ssh\MyKeyPair.pem*).
Note: You can't use Windows Explorer to create a folder with a name that begins with a period unless you also end the folder name with a period. After you enter the name (.ssh.), the final period is removed automatically.
Mac/Linux users: We recommend saving your key pair in the .ssh sub-directory from your home directory (ex. *~/.ssh/MyKeyPair.pem*).
Note: On Mac, the key pair is downloaded to your Downloads directory by default. To move the key pair into the .ssh sub-directory, enter the following command in a terminal window: mv ~/Downloads/MyKeyPair.pem ~/.ssh/MyKeyPair.pem
After you have stored your key pair, click Launch Instance to start your Linux instance.



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.

Create a new key pair

Key pair name

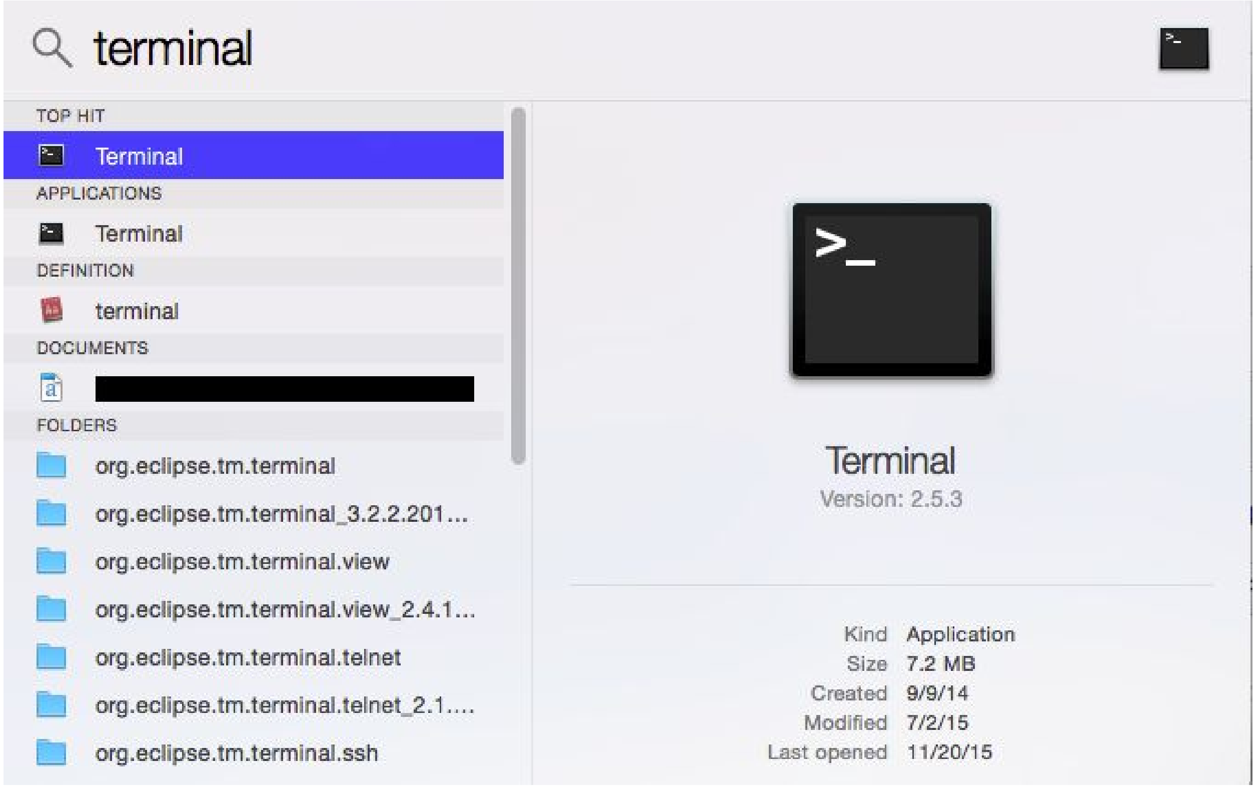
MyKeyPair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

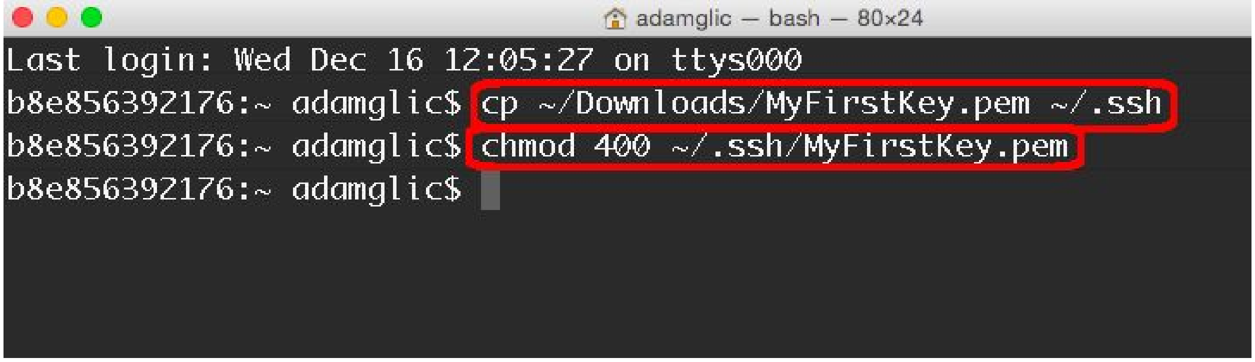
Step 3: Connect to your Instance

After launching your instance, it's time to connect to it.
Windows users: Select Windows below to see instructions for installing Git Bash.
Mac/Linux user: Select Mac / Linux below to see instructions for opening a terminal window.

Your Mac or Linux computer most likely includes an SSH client by default. You can check for an SSH client by typing ssh at the command line. If your computer doesn't recognize the command, the OpenSSH project provides a free implementation of the full suite of SSH tools that you can download.
a. Mac users: Open a terminal window by pressing Command + Space and typing terminal in the search window. Then press enter to open the terminal window.
Linux users: Open a terminal window.



b. Use the chmod command to make sure your private key file is not publicly viewable by entering the following command: chmod 400 ~/.ssh/mykeypair.pem to restrict permissions to your private SSH key.
Note: You do not need to do this every time you connect to you instance, you only need to set this once per SSH key that you have.

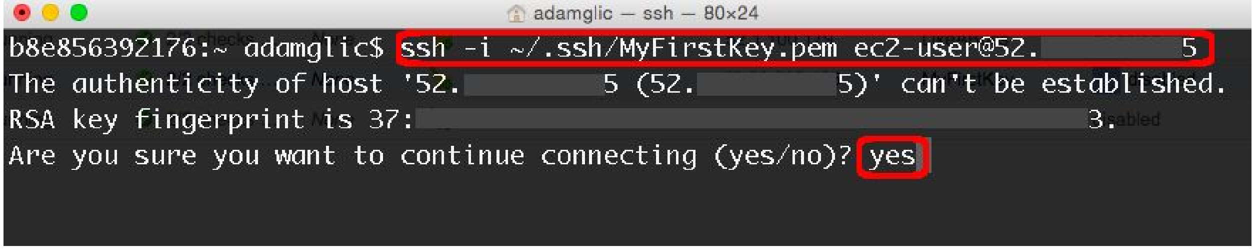


```
Last login: Wed Dec 16 12:05:27 on ttys000
b8e856392176:~ adamglic$ cp ~/Downloads/MyFirstKey.pem ~/.ssh
b8e856392176:~ adamglic$ chmod 400 ~/.ssh/MyFirstKey.pem
b8e856392176:~ adamglic$ 
```

c. Use SSH to connect to your instance. In this case the user name is ec2-user, the SSH key is stored in the directory we saved it to in step 2 part d, and the IP address is from step 2 part f. The format is *ssh -i {full path of your .pem file} ec2-user@{instance IP address}*.
Windows users: Enter ssh -i 'c:\Users\yourusername\.ssh\MyKeyPair.pem' ec2-user@*{IP_Address}* (ex. *ssh -i 'c:\Users\adamglic\.ssh\MyKeyPair.pem' ec2-user@52.27.212.125*)
Mac/Linux users: Enter ssh -i ~/.ssh/MyKeyPair.pem ec2-user@*{IP_Address}* (ex. ssh -i ~/.ssh/MyKeyPair.pem ec2-user@52.27.212.125)

> Note: if you started a Linux instance that isn't Amazon Linux, there may by a different user name that is used. common user names include *ec2-user*, *root*, *ubuntu*, and *fedora*. If you are unsure what the login user name is, check with your AMI provider.

You'll see a response similar to the following:
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)' can't be established. RSA key fingerprint is 1f:51:ae:28:df:63:e9:d8:cf:38:5d:87:2d:7b:b8:ca:9f:f5:b1:6f. Are you sure you want to continue connecting (yes/no)?
Type yes and press enter.

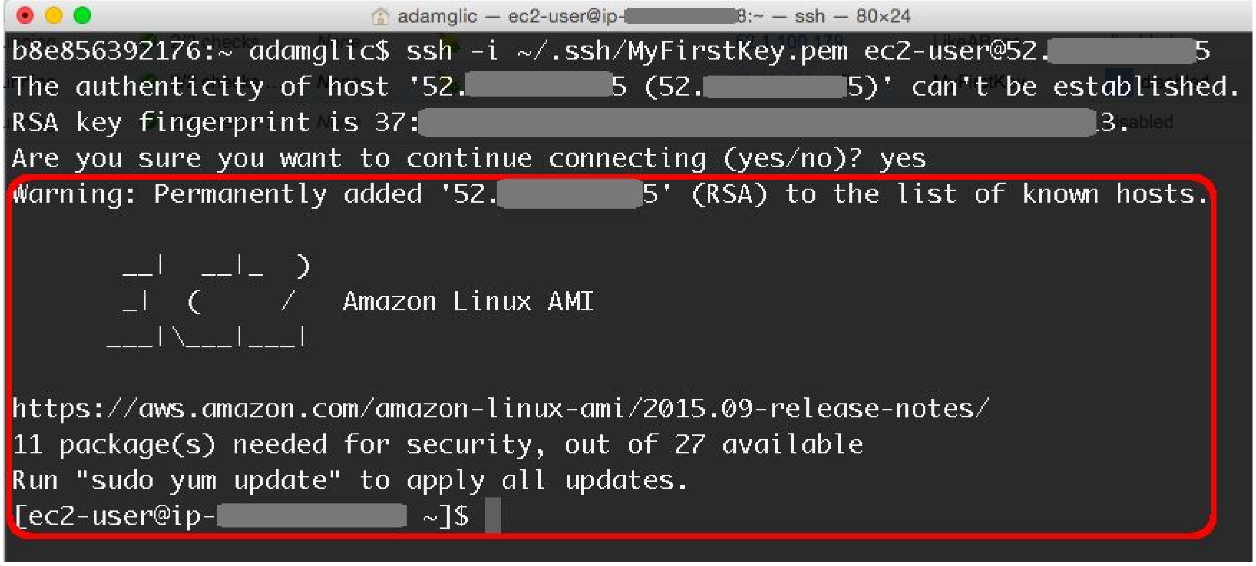


```
b8e856392176:~ adamglic$ ssh -i ~/.ssh/MyFirstKey.pem ec2-user@52.[redacted]5
The authenticity of host '52.[redacted]5 (52.[redacted]5)' can't be established.
RSA key fingerprint is 37:[redacted]3.
Are you sure you want to continue connecting (yes/no)? yes
```

You'll see a response similar to the following:

Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA) to the list of known hosts.

You should then see the welcome screen for your instance and you are now connected to your AWS Linux virtual machine in the cloud.

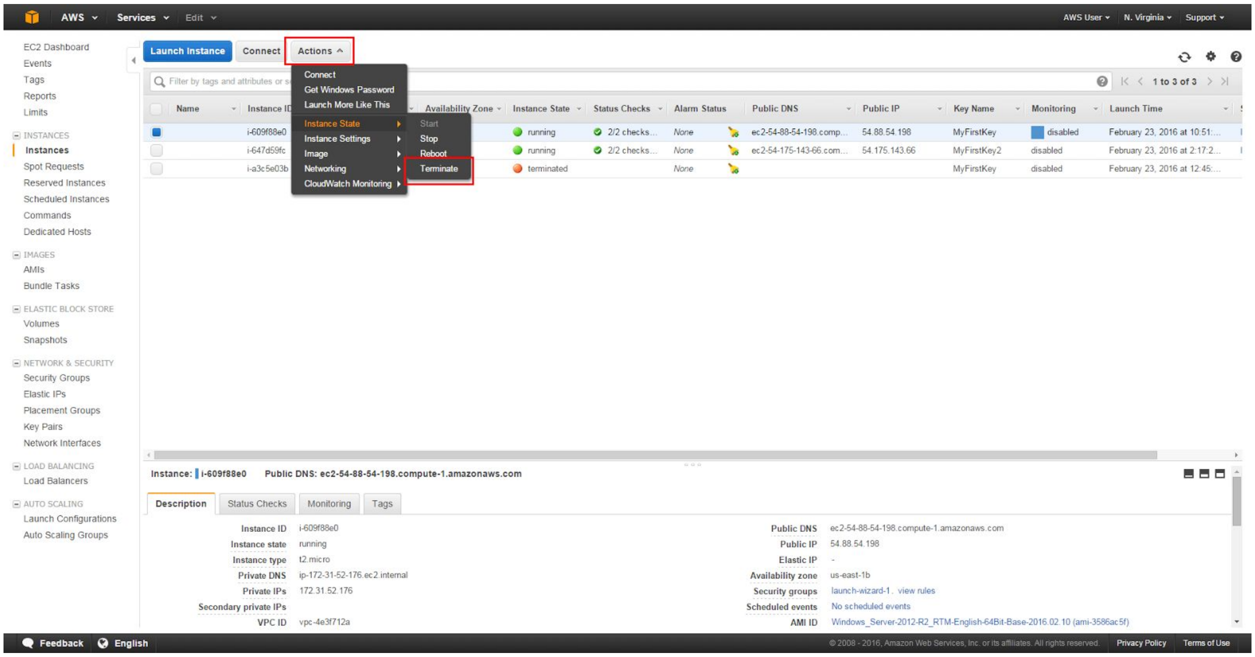


```
b8e856392176:~ adamglic$ ssh -i ~/.ssh/MyFirstKey.pem ec2-user@52.[redacted]5
The authenticity of host '52.[redacted]5 (52.[redacted]5)' can't be established.
RSA key fingerprint is 37:[redacted]3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.[redacted]5' (RSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2015.09-release-notes/
11 package(s) needed for security, out of 27 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-[redacted] ~]$
```

Step 4: Terminate Your Instance

You can easily terminate the instance from the EC2 console. In fact, it is a best practice to terminate instances you are no longer using so you don't keep getting charged for them.

a. Back on the EC2 Console, select the box next to the instance you created. Then click the Actions button, navigate to *Instance State,* and click Terminate.



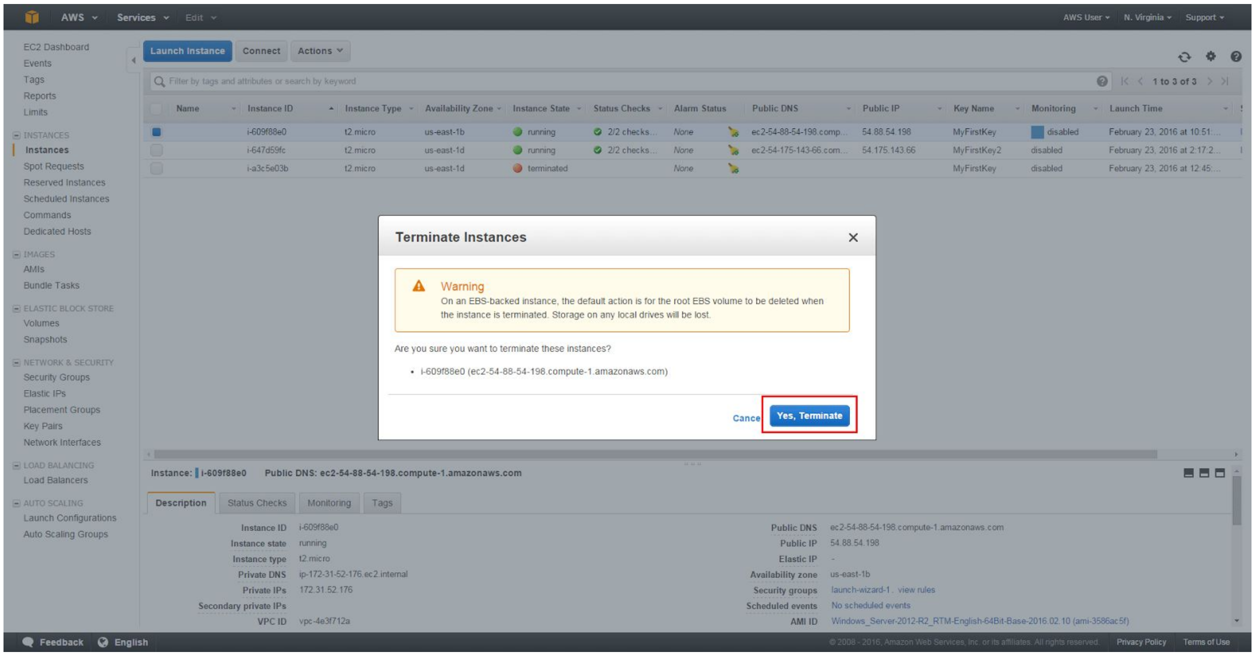
Name	Instance ID	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring	Launch Time
	i-609f88e0		running	2/2 checks...	None	ec2-54-88-54-198.comp...	54.88.54.198	MyFirstKey	disabled	February 23, 2016 at 10:51:...
	i-647d59fc		running	2/2 checks...	None	ec2-54-175-143-66.com...	54.175.143.66	MyFirstKey2	disabled	February 23, 2016 at 2:17:2...
	i-a3c5e03b		terminated		None			MyFirstKey	disabled	February 23, 2016 at 12:45:...

Instance: i-609f88e0 Public DNS: ec2-54-88-54-198.compute-1.amazonaws.com

Instance ID	i-609f88e0	Public DNS	ec2-54-88-54-198.compute-1.amazonaws.com
Instance state	running	Public IP	54.88.54.198
Instance type	t2.micro	Elastic IP	-
Private DNS	ip-172-31-52-176.ec2.internal	Availability zone	us-east-1b
Private IPs	172.31.52.176	Security groups	launch-wizard-1. view rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-4e3f712a	AMI ID	Windows_Server-2012-R2_RTM-English-64Bit-Base-2016.02.10 (ami-3586ac5f)

b. You will be asked to confirm your termination - select Yes, Terminate.

Note: This process can take several seconds to complete. Once your instance has been terminated, the Instance State will change to *terminated* on your EC2 Console.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring	Launch Time
	i-609f88e0	t2.micro	us-east-1b	running	2/2 checks...	None	ec2-54-88-54-198.comp...	54.88.54.198	MyFirstKey	disabled	February 23, 2016 at 10:51:...
	i-647d59fc	t2.micro	us-east-1d	running	2/2 checks...	None	ec2-54-175-143-66.com...	54.175.143.66	MyFirstKey2	disabled	February 23, 2016 at 2:17:2...
	i-a3c5e03b	t2.micro	us-east-1d	terminated		None			MyFirstKey	disabled	February 23, 2016 at 12:45:...

Terminate Instances

Warning

On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.

Are you sure you want to terminate these instances?

- i-609f88e0 (ec2-54-88-54-198.compute-1.amazonaws.com)

Instance: i-609f88e0 Public DNS: ec2-54-88-54-198.compute-1.amazonaws.com

Instance ID	i-609f88e0	Public DNS	ec2-54-88-54-198.compute-1.amazonaws.com
Instance state	running	Public IP	54.88.54.198
Instance type	t2.micro	Elastic IP	-
Private DNS	ip-172-31-52-176.ec2.internal	Availability zone	us-east-1b
Private IPs	172.31.52.176	Security groups	launch-wizard-1. view rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-4e3f712a	AMI ID	Windows_Server-2012-R2_RTM-English-64Bit-Base-2016.02.10 (ami-3586ac5f)

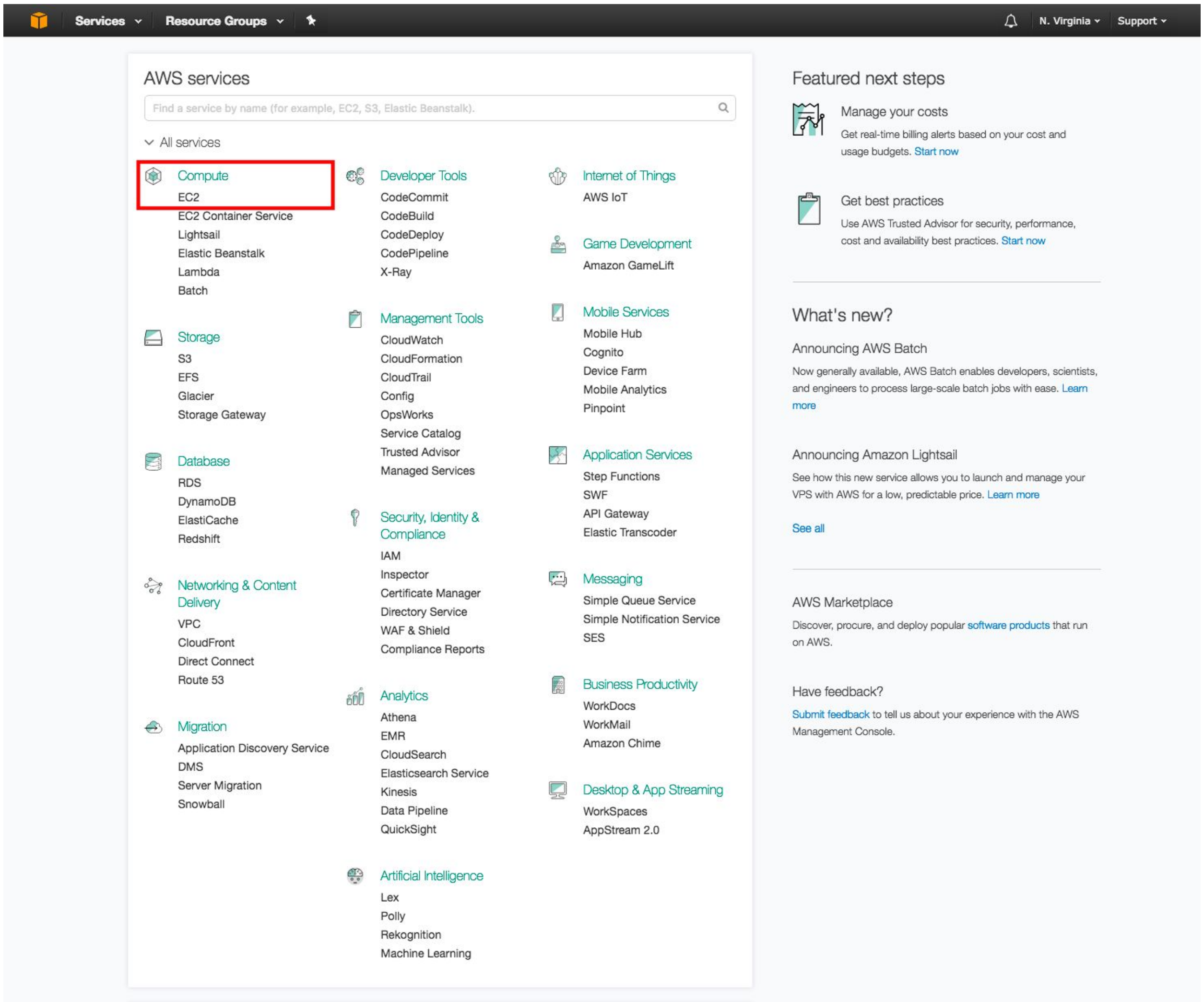
Launch a WordPress Website

This step-by-step guide will help you get a website up and running with WordPress*, installed on an Amazon EC2 virtual machine (also known as an "instance"). You will go through how to configure and launch an EC2 instance, how to get your WordPress username and password, and how to log into your WordPress admin portal. Everything done in this tutorial is free-tier eligible.

> ***Note: This guide results in an architecture for a basic WordPress website intended for personal use or learning. This tutorial is not recommended for business-level websites with high scalability needs. For a more advanced tutorial, see the Build a Wordpress Website Project or Wordpress: Best Practices on AWS.**

Did you know? AWS made it even easier to launch a WordPress website. Jumpstart your website with

Open the AWS Management Console and you can keep this step-by-step guide open. When the screen loads, enter your user name and password to get started. Then find EC2 under *Compute*, and double click to open the dashboard.



AWS services

Find a service by name (for example, EC2, S3, Elastic Beanstalk).

All services

Compute
EC2
EC2 Container Service
Lightsail
Elastic Beanstalk
Lambda
Batch

Storage
S3
EFS
Glacier
Storage Gateway

Database
RDS
DynamoDB
ElastiCache
Redshift

Networking & Content Delivery
VPC
CloudFront
Direct Connect
Route 53

Migration
Application Discovery Service
DMS
Server Migration
Snowball

Developer Tools
CodeCommit
CodeBuild
CodeDeploy
CodePipeline
X-Ray

Management Tools
CloudWatch
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Trusted Advisor
Managed Services

Security, Identity & Compliance
IAM
Inspector
Certificate Manager
Directory Service
WAF & Shield
Compliance Reports

Analytics
Athena
EMR
CloudSearch
Elasticsearch Service
Kinesis
Data Pipeline
QuickSight

Artificial Intelligence
Lex
Polly
Rekognition
Machine Learning

Internet of Things
AWS IoT

Game Development
Amazon GameLift

Mobile Services
Mobile Hub
Cognito
Device Farm
Mobile Analytics
Pinpoint

Application Services
Step Functions
SWF
API Gateway
Elastic Transcoder

Messaging
Simple Queue Service
Simple Notification Service
SES

Business Productivity
WorkDocs
WorkMail
Amazon Chime

Desktop & App Streaming
WorkSpaces
AppStream 2.0

Featured next steps

Manage your costs

Get real-time billing alerts based on your cost and usage budgets. Start now

Get best practices

Use AWS Trusted Advisor for security, performance, cost and availability best practices. Start now

What's new?

Announcing AWS Batch

Now generally available, AWS Batch enables developers, scientists, and engineers to process large-scale batch jobs with ease. Learn more

Announcing Amazon Lightsail

See how this new service allows you to launch and manage your VPS with AWS for a low, predictable price. Learn more

See all

AWS Marketplace

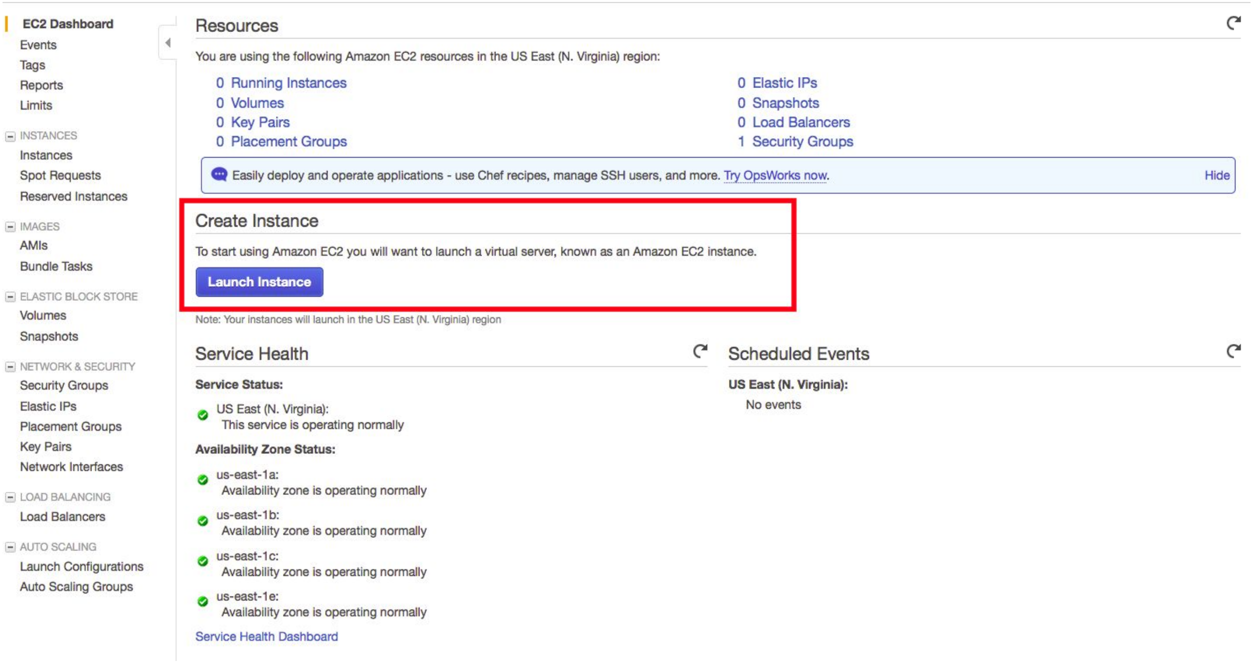
Discover, procure, and deploy popular software products that run on AWS.

Have feedback?

Submit feedback to tell us about your experience with the AWS Management Console.

Step 1: Launch an Amazon EC2 Instance

Now you are in the EC2 dashboard, click Launch Instance from the dashboard to create and configure your virtual machine.



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

0 Running Instances	0 Elastic IPs
0 Volumes	0 Snapshots
0 Key Pairs	0 Load Balancers
0 Placement Groups	1 Security Groups

Easily deploy and operate applications - use Chef recipes, manage SSH users, and more. Try OpsWorks now.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (N. Virginia) region

Service Health

Service Status:

US East (N. Virginia):
This service is operating normally

Availability Zone Status:

us-east-1a:
Availability zone is operating normally

us-east-1b:
Availability zone is operating normally

us-east-1c:
Availability zone is operating normally

us-east-1e:
Availability zone is operating normally

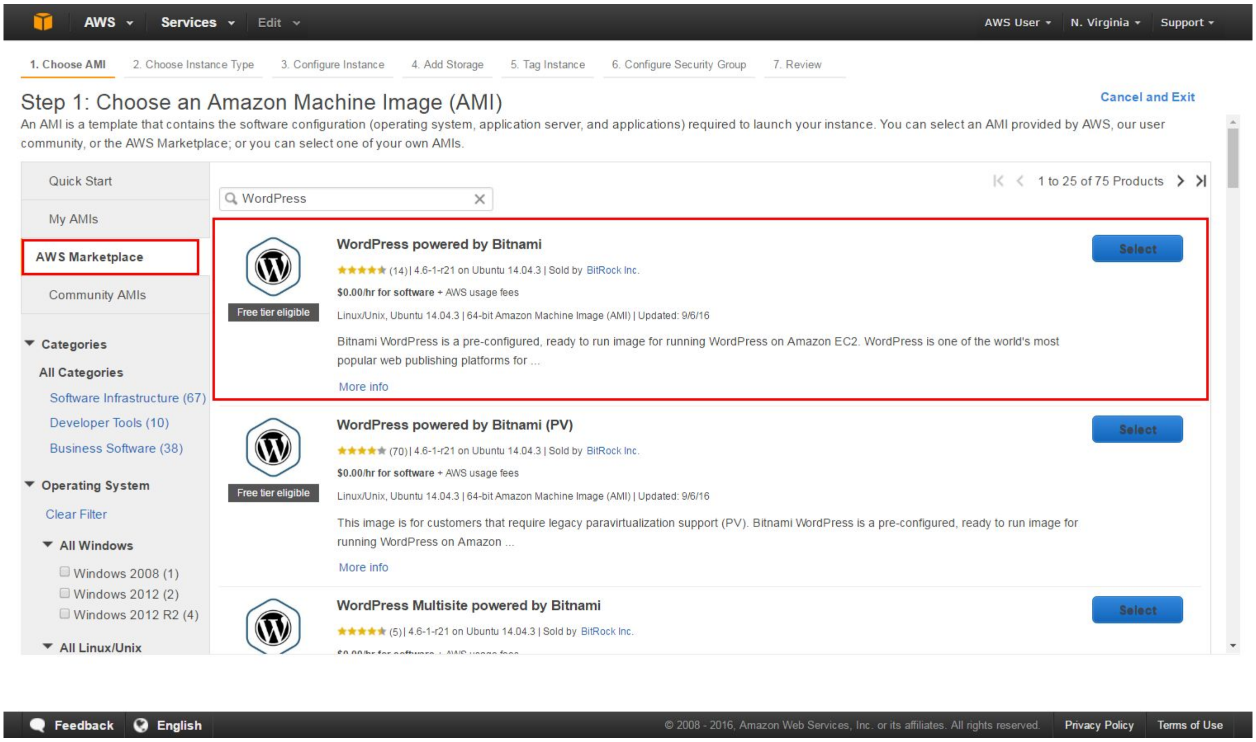
Scheduled Events

US East (N. Virginia):
No events

Step 2: Configure your Instance

Now you're in the Amazon EC2 configuration wizard, we will be using an existing Amazon Machine Image (AMI) from the AWS Marketplace that has WordPress already installed. The AWS Marketplace provides access to thousands of pre-configured images for common pieces of software.

1. Click on AWS Marketplace on the left-hand side, search for WordPress, look for *WordPress powered by BitNami*, then click Select.



Step 1: Choose an Amazon Machine Image (AMI)

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

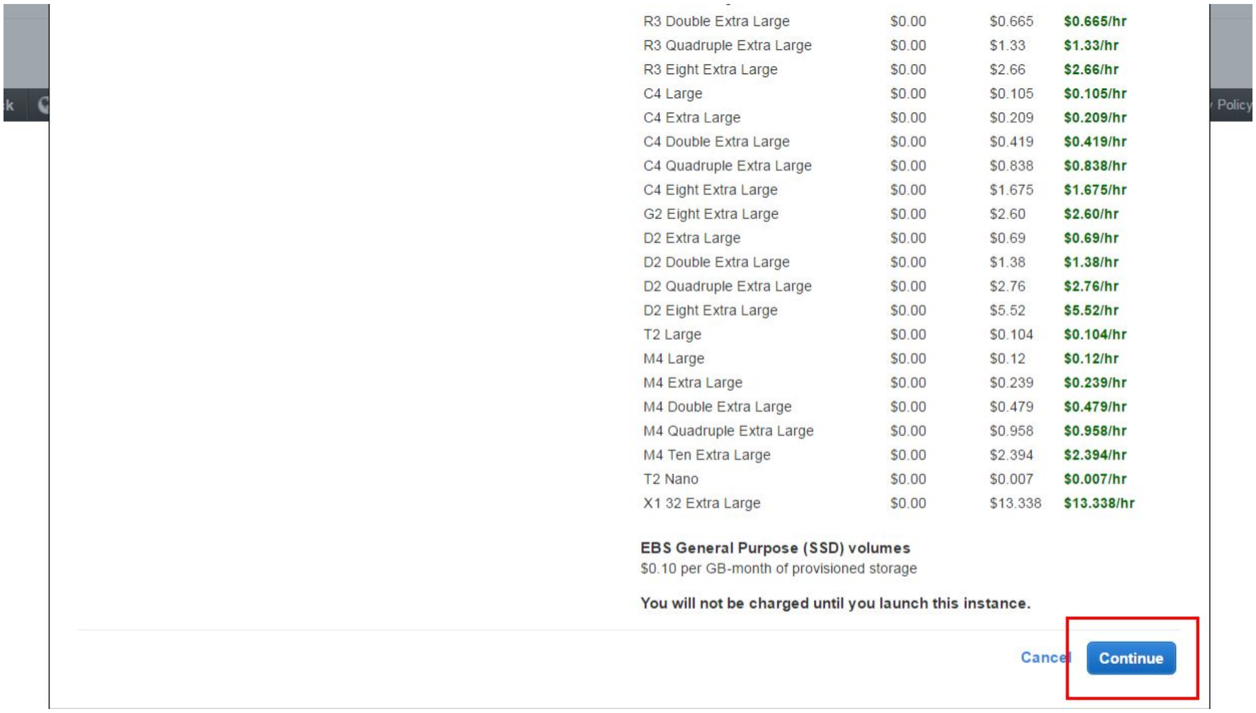
WordPress powered by Bitnami
★★★★★ (14) | 4.6-1-r21 on Ubuntu 14.04.3 | Sold by BitRock Inc.
$0.00/hr for software + AWS usage fees
Linux/Unix, Ubuntu 14.04.3 | 64-bit Amazon Machine Image (AMI) | Updated: 9/6/16
Bitnami WordPress is a pre-configured, ready to run image for running WordPress on Amazon EC2. WordPress is one of the world's most popular web publishing platforms for ...

WordPress powered by Bitnami (PV)
★★★★★ (70) | 4.6-1-r21 on Ubuntu 14.04.3 | Sold by BitRock Inc.
$0.00/hr for software + AWS usage fees
Linux/Unix, Ubuntu 14.04.3 | 64-bit Amazon Machine Image (AMI) | Updated: 9/6/16
This image is for customers that require legacy paravirtualization support (PV). Bitnami WordPress is a pre-configured, ready to run image for running WordPress on Amazon ...

WordPress Multisite powered by Bitnami
★★★★★ (5) | 4.6-1-r21 on Ubuntu 14.04.3 | Sold by BitRock Inc.

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

2. You will be presented a detailed pricing page. In this case, the price will be $0.00 for the software regardless of the size of the instance that you use.
Scroll to the bottom and click Continue.

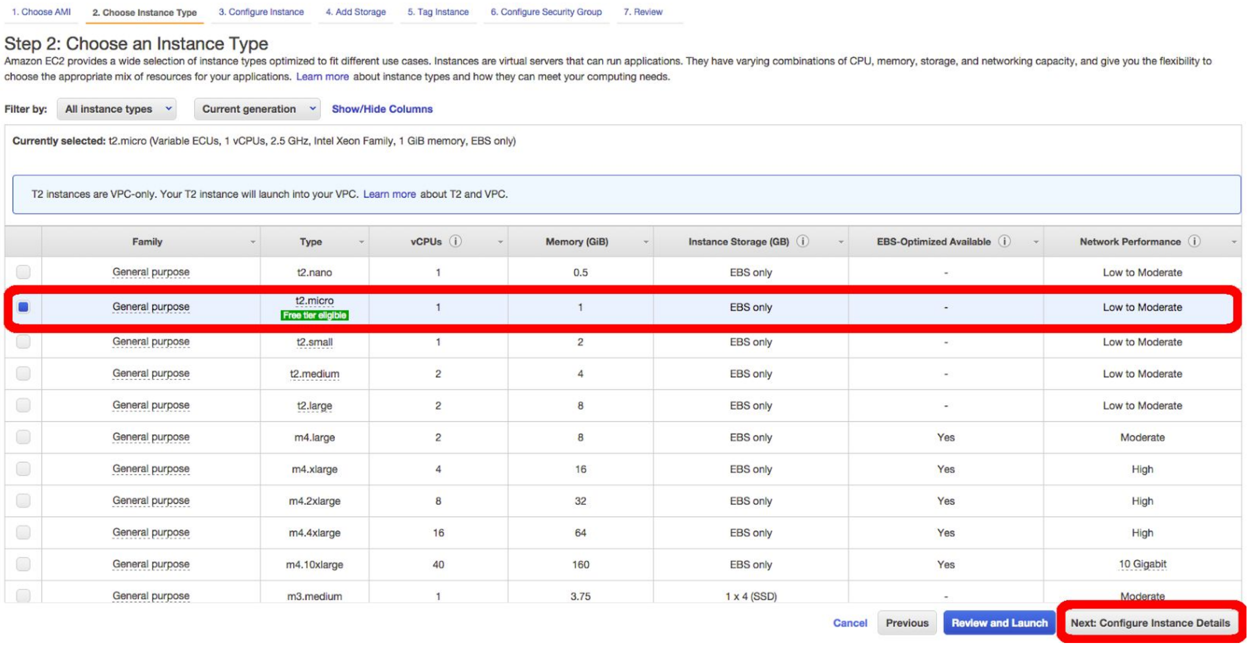


R3 Double Extra Large	$0.00	$0.665	**$0.665/hr**
R3 Quadruple Extra Large	$0.00	$1.33	**$1.33/hr**
R3 Eight Extra Large	$0.00	$2.66	**$2.66/hr**
C4 Large	$0.00	$0.105	**$0.105/hr**
C4 Extra Large	$0.00	$0.209	**$0.209/hr**
C4 Double Extra Large	$0.00	$0.419	**$0.419/hr**
C4 Quadruple Extra Large	$0.00	$0.838	**$0.838/hr**
C4 Eight Extra Large	$0.00	$1.675	**$1.675/hr**
G2 Eight Extra Large	$0.00	$2.60	**$2.60/hr**
D2 Extra Large	$0.00	$0.69	**$0.69/hr**
D2 Double Extra Large	$0.00	$1.38	**$1.38/hr**
D2 Quadruple Extra Large	$0.00	$2.76	**$2.76/hr**
D2 Eight Extra Large	$0.00	$5.52	**$5.52/hr**
T2 Large	$0.00	$0.104	**$0.104/hr**
M4 Large	$0.00	$0.12	**$0.12/hr**
M4 Extra Large	$0.00	$0.239	**$0.239/hr**
M4 Double Extra Large	$0.00	$0.479	**$0.479/hr**
M4 Quadruple Extra Large	$0.00	$0.958	**$0.958/hr**
M4 Ten Extra Large	$0.00	$2.394	**$2.394/hr**
T2 Nano	$0.00	$0.007	**$0.007/hr**
X1 32 Extra Large	$0.00	$13.338	**$13.338/hr**

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

You will not be charged until you launch this instance.

3. For this tutorial, we will be using a free-tier eligible t2.micro instance. Click on t2.micro in the *Type* column (it should be the first one), then click Next: Configure Instance Details. It may take a few seconds to load.
On the following screens, click Next: Add Storage and then Next: Tag Instance.



Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

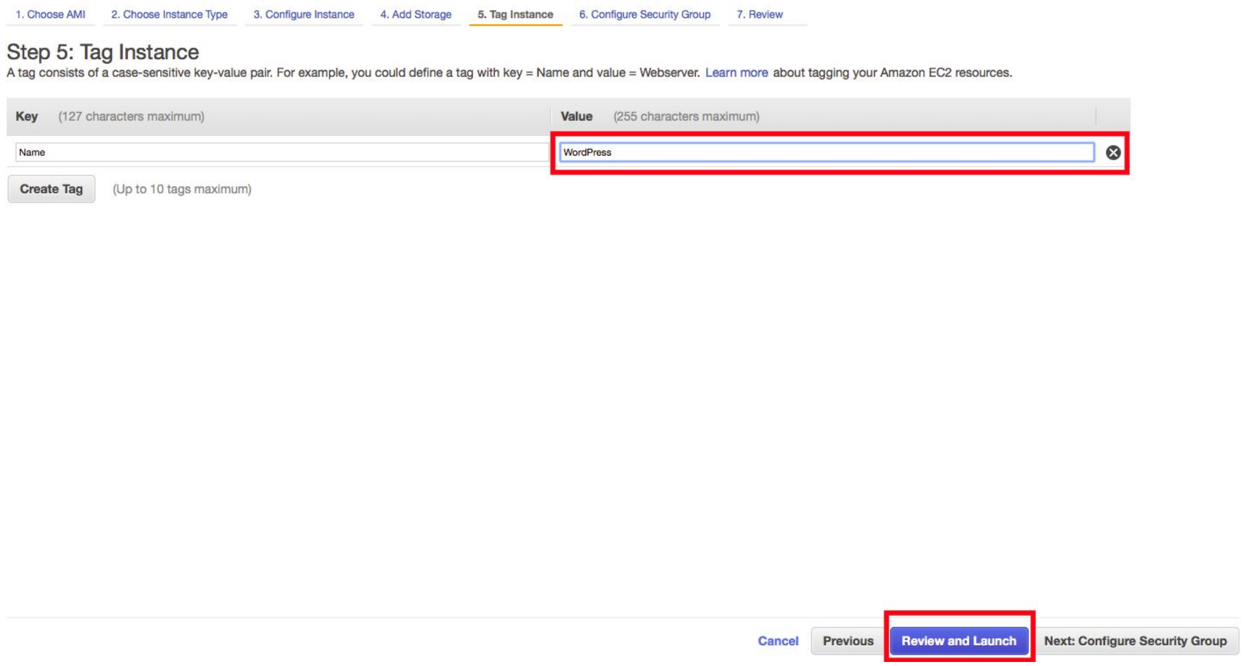
Filter by: All instance types; Current generation; Show/Hide Columns

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

T2 instances are VPC-only. Your T2 instance will launch into your VPC. Learn more about T2 and VPC.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
□	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
□	General purpose	m4.large	2	8	EBS only	Yes	Moderate
□	General purpose	m4.xlarge	4	16	EBS only	Yes	High
□	General purpose	m4.2xlarge	8	32	EBS only	Yes	High
□	General purpose	m4.4xlarge	16	64	EBS only	Yes	High
□	General purpose	m4.10xlarge	40	160	EBS only	Yes	10 Gigabit
□	General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate

4. We will set a name for your instance in this step. Enter WordPress in the Value box next to the Name box. Click Review and Launch to continue.

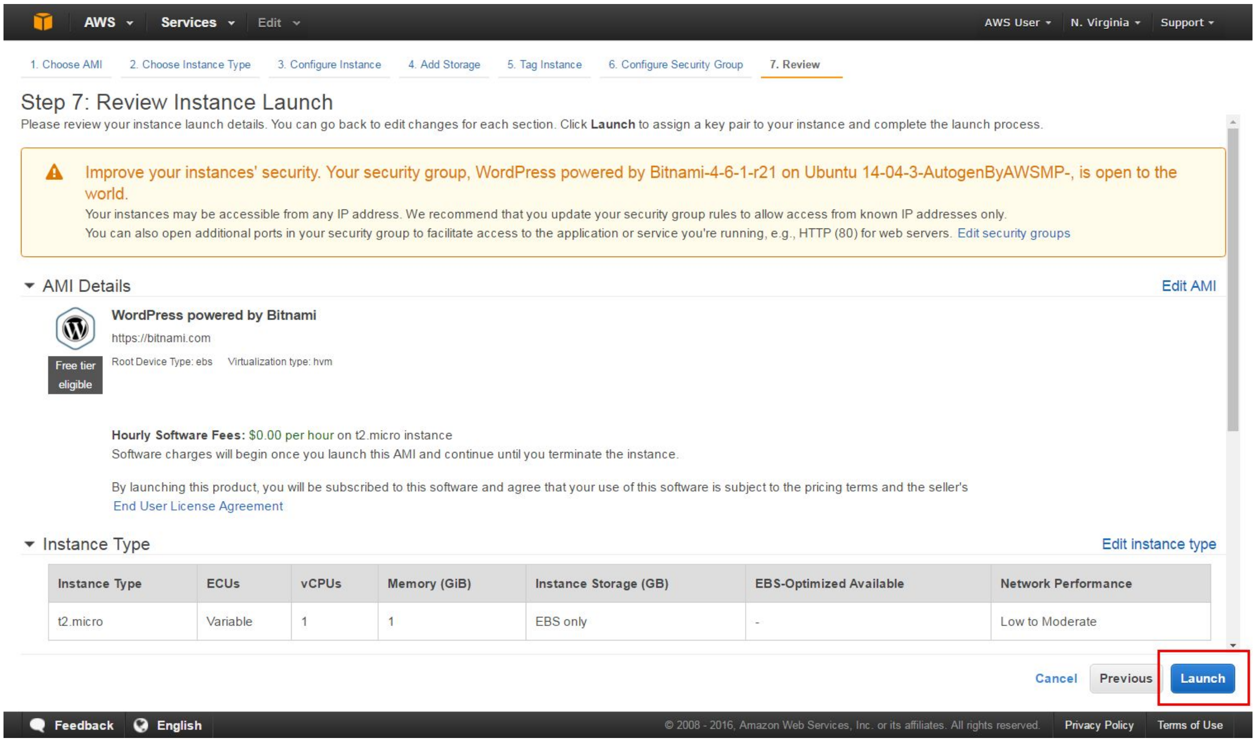


Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	WordPress

5. You can review your instance configurations, then click Launch when you're ready to start your Amazon EC2 instance running WordPress.



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-4-6-1-r21 on Ubuntu 14-04-3-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
Free tier eligible

Hourly Software Fees: $0.00 per hour on t2.micro instance
Software charges will begin once you launch this AMI and continue until you terminate the instance.

By launching this product, you will be subscribed to this software and agree that your use of this software is subject to the pricing terms and the seller's End User License Agreement

Instance Type — Edit instance type

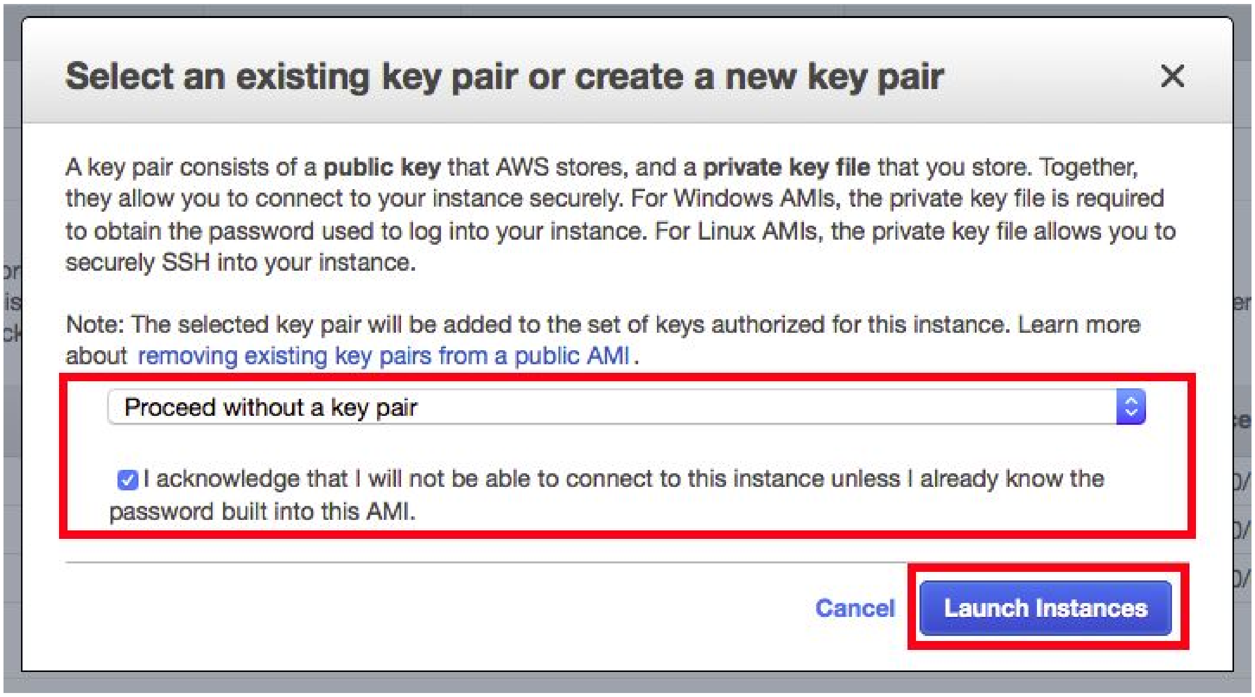
Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

6. The next screen deals with key-pairs. Key-pairs are how you can connect to your EC2 instances via a terminal program using Secure Shell (SSH). Select Proceed without a key pair, and check the box acknowledging that you know you need this key to access your EC2 instance.

Click Launch Instances to launch your instance. Be aware that starting the instance up may take a few minutes.

Note: To connect to your instance directly, you will need to create a new key pair. For instructions on creating a key pair and connecting to an instance, see steps 2 d.-2 f. and 3 of the tutorial: Launch a Linux Virtual Machine.



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.

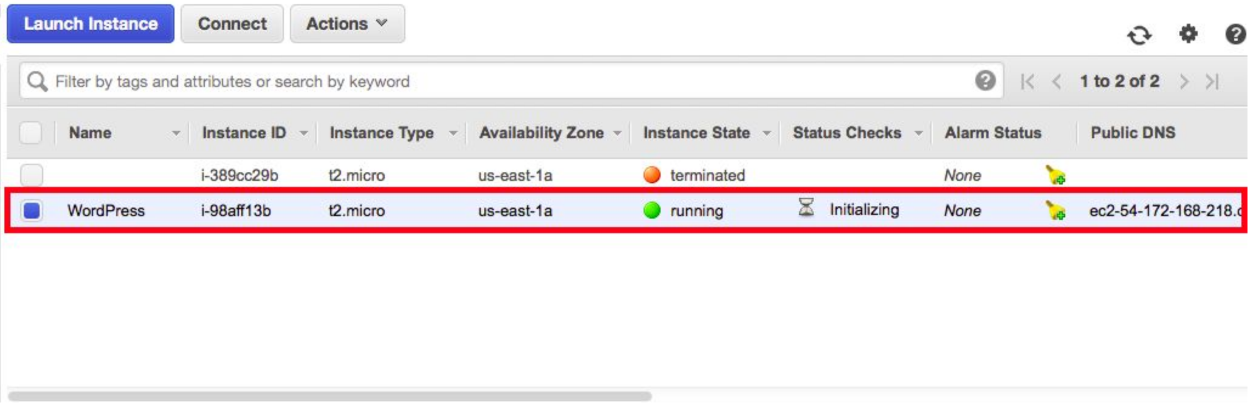
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

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.

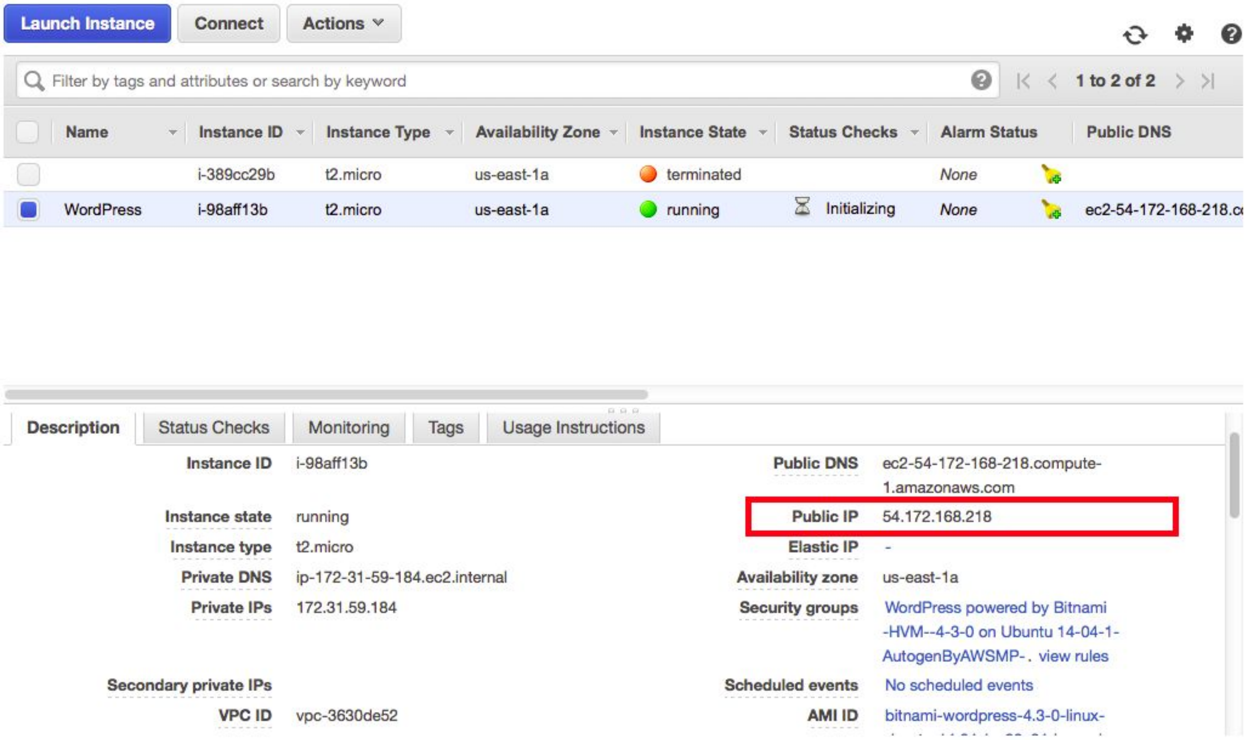
Cancel　Launch Instances

7. Click View *Instances* on the bottom right of the page (you may need to scroll down to see it). Then select the WordPress instance, make sure the *Instance State* says *running*. If Instance State says *launching* then AWS is still preparing your WordPress instance.



□	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status		Public DNS
□		i-389cc29b	t2.micro	us-east-1a	● terminated		None		
■	WordPress	i-98aff13b	t2.micro	us-east-1a	● running	⌛ Initializing	None		ec2-54-172-168-218.c

8. Once your instance is running, you can now test your WordPress website. Find the Public IP for your instance at the bottom of this page.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-389cc29b	t2.micro	us-east-1a	● terminated		*None*	
WordPress	i-98aff13b	t2.micro	us-east-1a	● running	Initializing	*None*	ec2-54-172-168-218.c

Instance ID i-98aff13b

Instance state running

Instance type t2.micro

Private DNS ip-172-31-59-184.ec2.internal

Private IPs 172.31.59.184

Secondary private IPs

VPC ID vpc-3630de52

Public DNS ec2-54-172-168-218.compute-1.amazonaws.com

Public IP 54.172.168.218

Elastic IP -

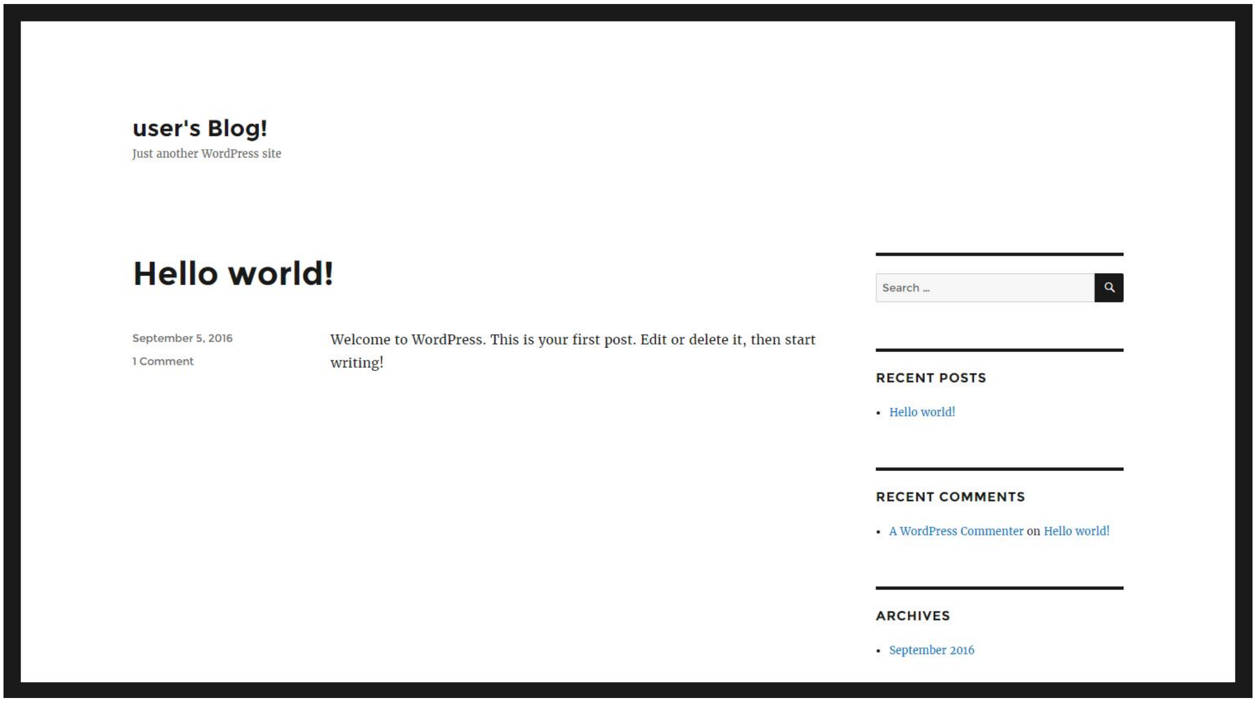
Availability zone us-east-1a

Security groups WordPress powered by Bitnami -HVM--4-3-0 on Ubuntu 14-04-1-AutogenByAWSMP-. view rules

Scheduled events No scheduled events

AMI ID bitnami-wordpress-4.3-0-linux-

9. Copy the Public IP into a new tab in your web browser, and you should see a *Hello World* blog page appear.



user's Blog!

Just another WordPress site

Hello world!

September 5, 2016

1 Comment

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

RECENT POSTS

- Hello world!

RECENT COMMENTS

- A WordPress Commenter on Hello world!

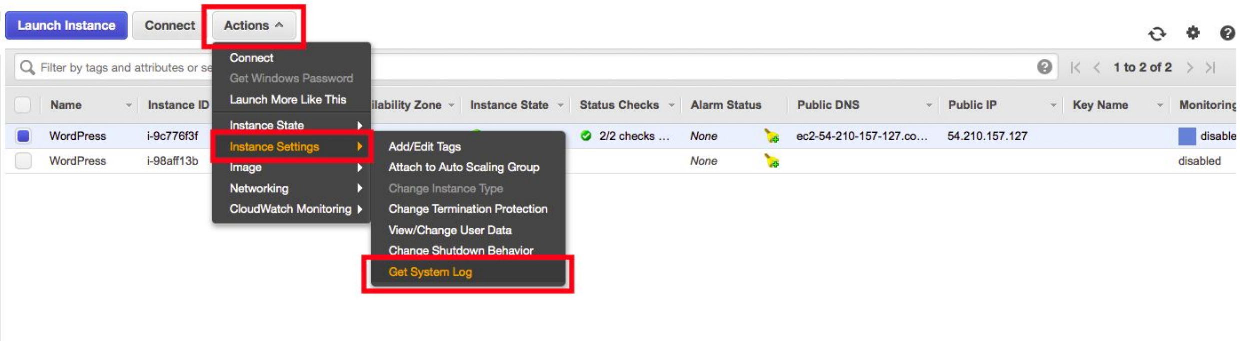
ARCHIVES

- September 2016

Step 3: Make Changes to Your Website

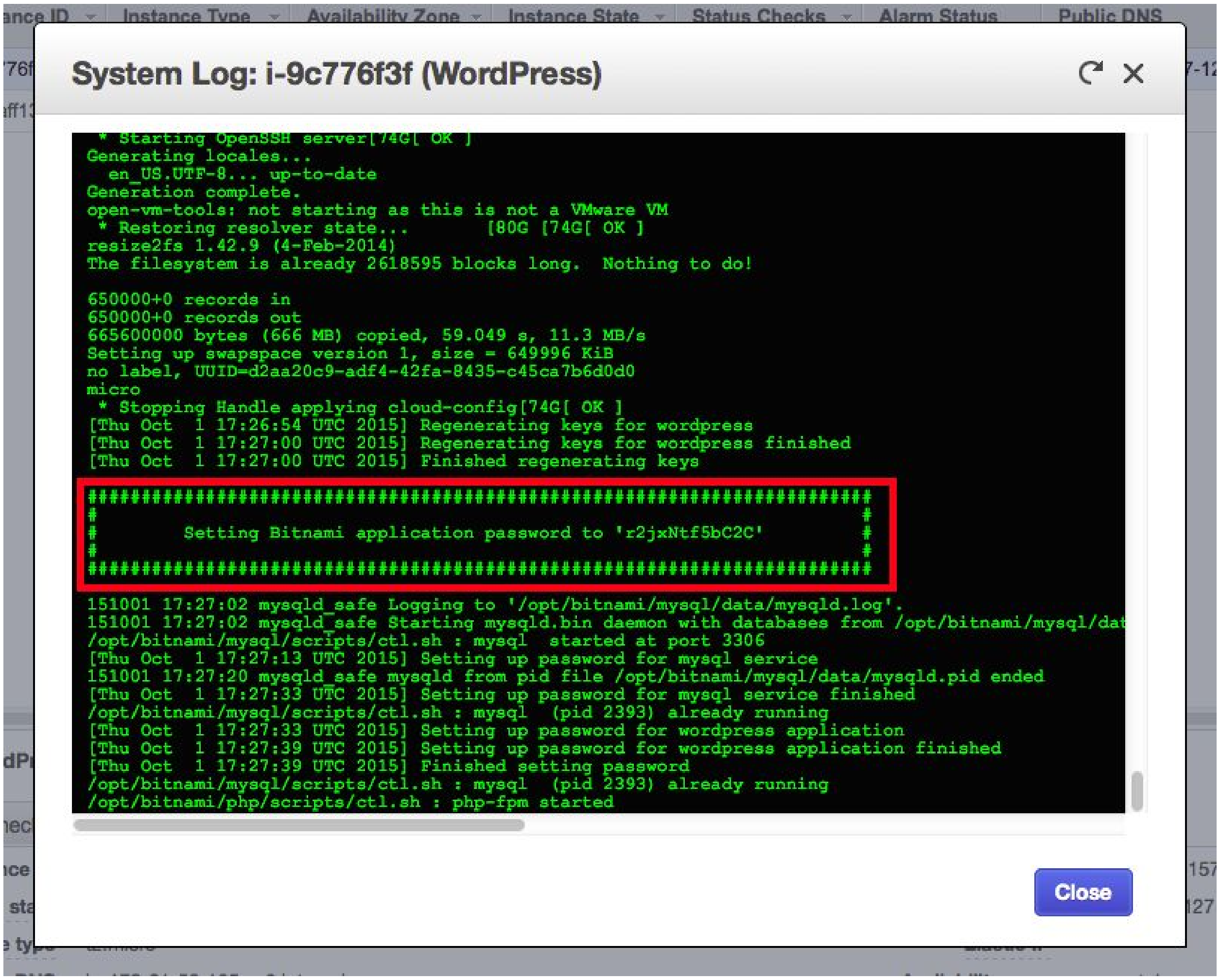
Now that you have your WordPress site up and running, it's time to log into its administration page so you can customize your site. To find your password, please follow the steps below:

1. Switch back to your EC2 management console in your web browser. Select WordPress instance, and click the Actions button. In the drop down menu, select Instance Setting, and choose Get System Log.



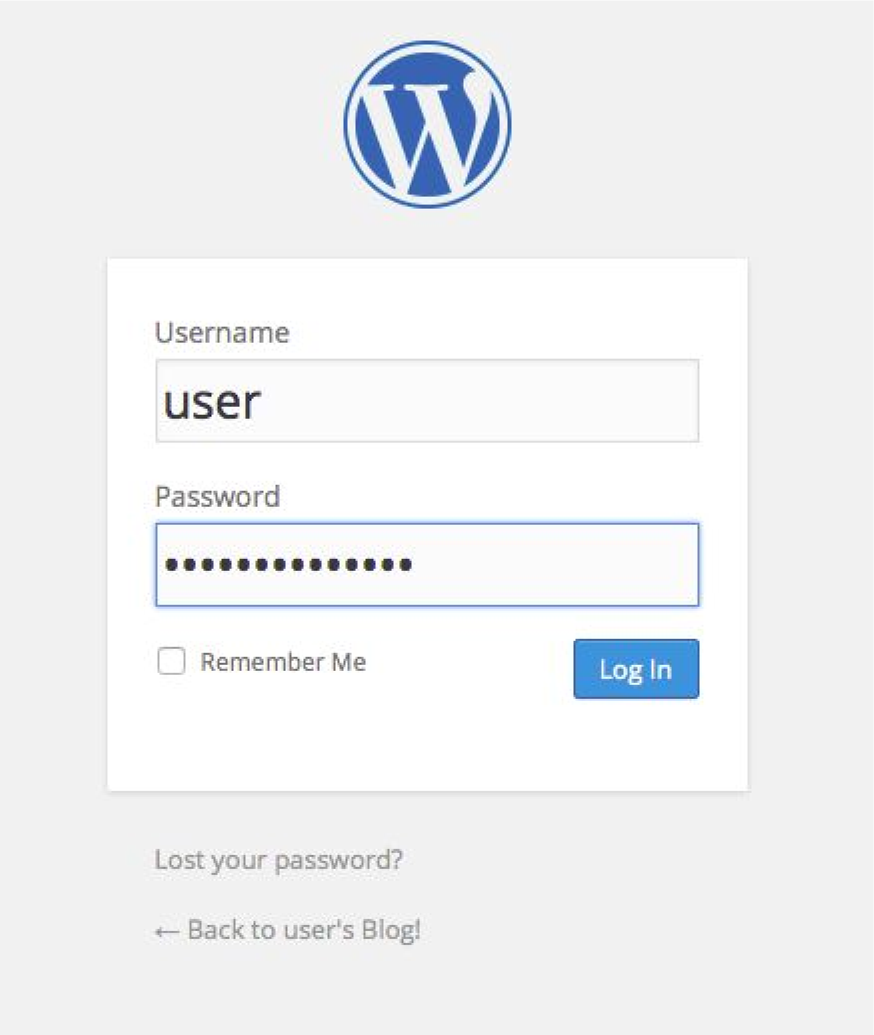
Name	Instance ID	ilability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring
WordPress	i-9c776f3f			2/2 checks ...	None	ec2-54-210-157-127.co...	54.210.157.127		disable
WordPress	i-98aff13b				None				disabled

2. In the system log window, scroll through to the bottom to find the *password* that's surrounded by hash marks.



```
 * Starting OpenSSH server[74G[ OK ]
Generating locales...
  en_US.UTF-8... up-to-date
Generation complete.
open-vm-tools: not starting as this is not a VMware VM
 * Restoring resolver state...       [80G [74G[ OK ]
resize2fs 1.42.9 (4-Feb-2014)
The filesystem is already 2618595 blocks long.  Nothing to do!

650000+0 records in
650000+0 records out
665600000 bytes (666 MB) copied, 59.049 s, 11.3 MB/s
Setting up swapspace version 1, size = 649996 KiB
no label, UUID=d2aa20c9-adf4-42fa-8435-c45ca7b6d0d0
micro
 * Stopping Handle applying cloud-config[74G[ OK ]
[Thu Oct  1 17:26:54 UTC 2015] Regenerating keys for wordpress
[Thu Oct  1 17:27:00 UTC 2015] Regenerating keys for wordpress finished
[Thu Oct  1 17:27:00 UTC 2015] Finished regenerating keys

##########################################################################
#                                                                        #
#        Setting Bitnami application password to 'r2jxNtf5bC2C'          #
#                                                                        #
##########################################################################

151001 17:27:02 mysqld_safe Logging to '/opt/bitnami/mysql/data/mysqld.log'.
151001 17:27:02 mysqld_safe Starting mysqld.bin daemon with databases from /opt/bitnami/mysql/dat
/opt/bitnami/mysql/scripts/ctl.sh : mysql  started at port 3306
[Thu Oct  1 17:27:13 UTC 2015] Setting up password for mysql service
151001 17:27:20 mysqld_safe mysqld from pid file /opt/bitnami/mysql/data/mysqld.pid ended
[Thu Oct  1 17:27:33 UTC 2015] Setting up password for mysql service finished
/opt/bitnami/mysql/scripts/ctl.sh : mysql  (pid 2393) already running
[Thu Oct  1 17:27:33 UTC 2015] Setting up password for wordpress application
[Thu Oct  1 17:27:39 UTC 2015] Setting up password for wordpress application finished
[Thu Oct  1 17:27:39 UTC 2015] Finished setting password
/opt/bitnami/mysql/scripts/ctl.sh : mysql  (pid 2393) already running
/opt/bitnami/php/scripts/ctl.sh : php-fpm started
```

3. Now that you have your password, switch back to the tab that you used to access the WordPress Hello World page. Add /admin to the end of the URL so it looks something like 54.192.32.144/admin. Hit enter.
Enter the Username user and the Password that you read from the log file.



Username

user

Password

••••••••••••••

□ Remember Me

Congratulations! You now have your WordPress site up and running. You can now manage, customize, and configure it as you like

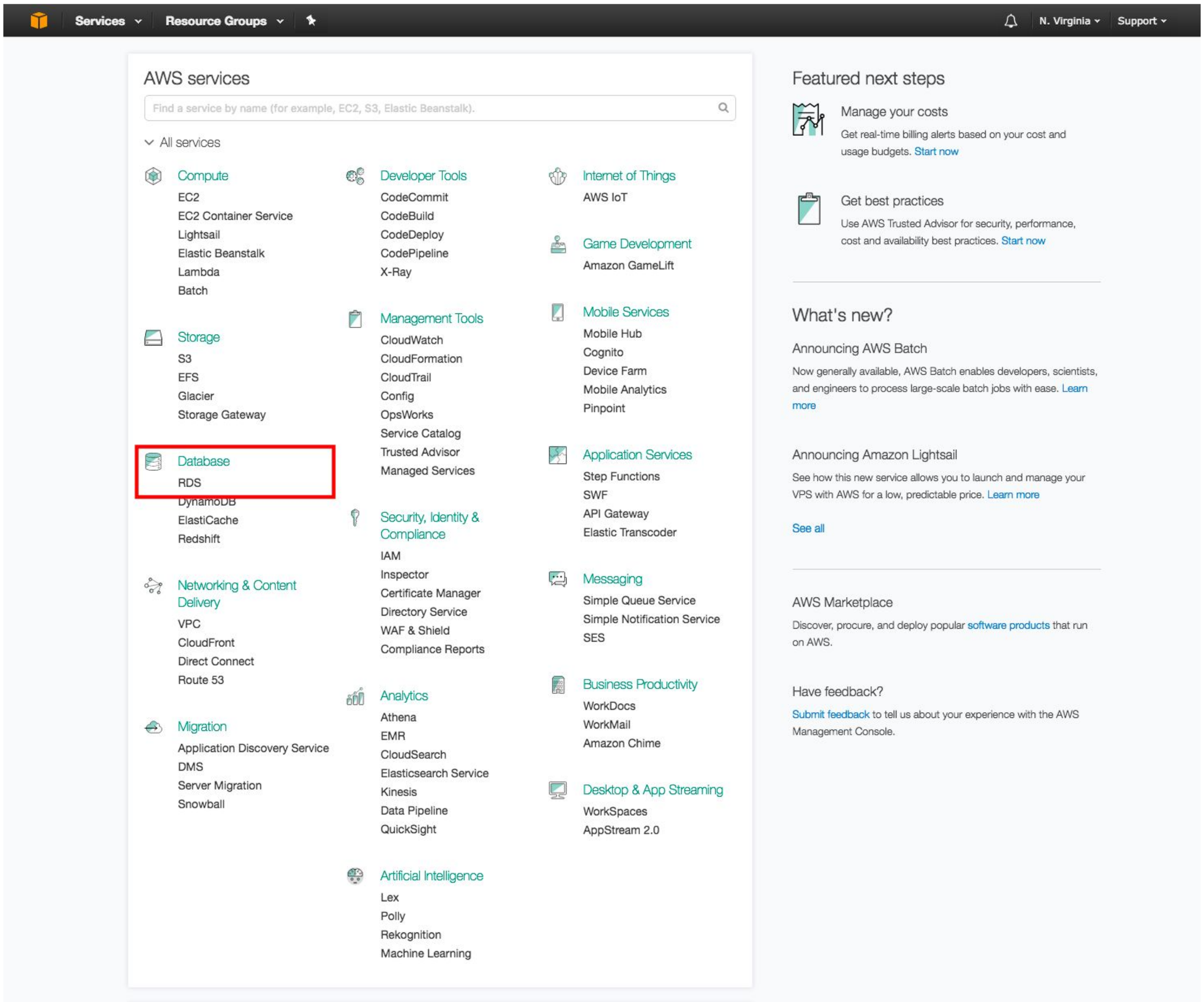
Create and Connect to a MySQL Database

with Amazon RDS or We can use

- **DynamoDB**
- **ElastiCache**
- **Redshift**

In this tutorial, you will learn how to create an environment to run your MySQL database (we call this environment an 'instance'), connect to the database, and delete the DB instance. We will do this using Amazon Relational Database Service (Amazon RDS) and everything done in this tutorial is free-tier eligible.

When you click here, the AWS management console will open in a new browser window, so you can keep this step-by-step guide open. When this screen loads, find RDS under *Database* and click to open the Amazon RDS Console.



Services Resource Groups N. Virginia Support

AWS services

Find a service by name (for example, EC2, S3, Elastic Beanstalk).

All services

Compute
EC2
EC2 Container Service
Lightsail
Elastic Beanstalk
Lambda
Batch

Storage
S3
EFS
Glacier
Storage Gateway

Database
RDS
DynamoDB
ElastiCache
Redshift

Networking & Content Delivery
VPC
CloudFront
Direct Connect
Route 53

Migration
Application Discovery Service
DMS
Server Migration
Snowball

Developer Tools
CodeCommit
CodeBuild
CodeDeploy
CodePipeline
X-Ray

Management Tools
CloudWatch
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Trusted Advisor
Managed Services

Security, Identity & Compliance
IAM
Inspector
Certificate Manager
Directory Service
WAF & Shield
Compliance Reports

Analytics
Athena
EMR
CloudSearch
Elasticsearch Service
Kinesis
Data Pipeline
QuickSight

Artificial Intelligence
Lex
Polly
Rekognition
Machine Learning

Internet of Things
AWS IoT

Game Development
Amazon GameLift

Mobile Services
Mobile Hub
Cognito
Device Farm
Mobile Analytics
Pinpoint

Application Services
Step Functions
SWF
API Gateway
Elastic Transcoder

Messaging
Simple Queue Service
Simple Notification Service
SES

Business Productivity
WorkDocs
WorkMail
Amazon Chime

Desktop & App Streaming
WorkSpaces
AppStream 2.0

Featured next steps

Manage your costs

Get real-time billing alerts based on your cost and usage budgets. Start now

Get best practices

Use AWS Trusted Advisor for security, performance, cost and availability best practices. Start now

What's new?

Announcing AWS Batch

Now generally available, AWS Batch enables developers, scientists, and engineers to process large-scale batch jobs with ease. Learn more

Announcing Amazon Lightsail

See how this new service allows you to launch and manage your VPS with AWS for a low, predictable price. Learn more

See all

AWS Marketplace

Discover, procure, and deploy popular software products that run on AWS.

Have feedback?

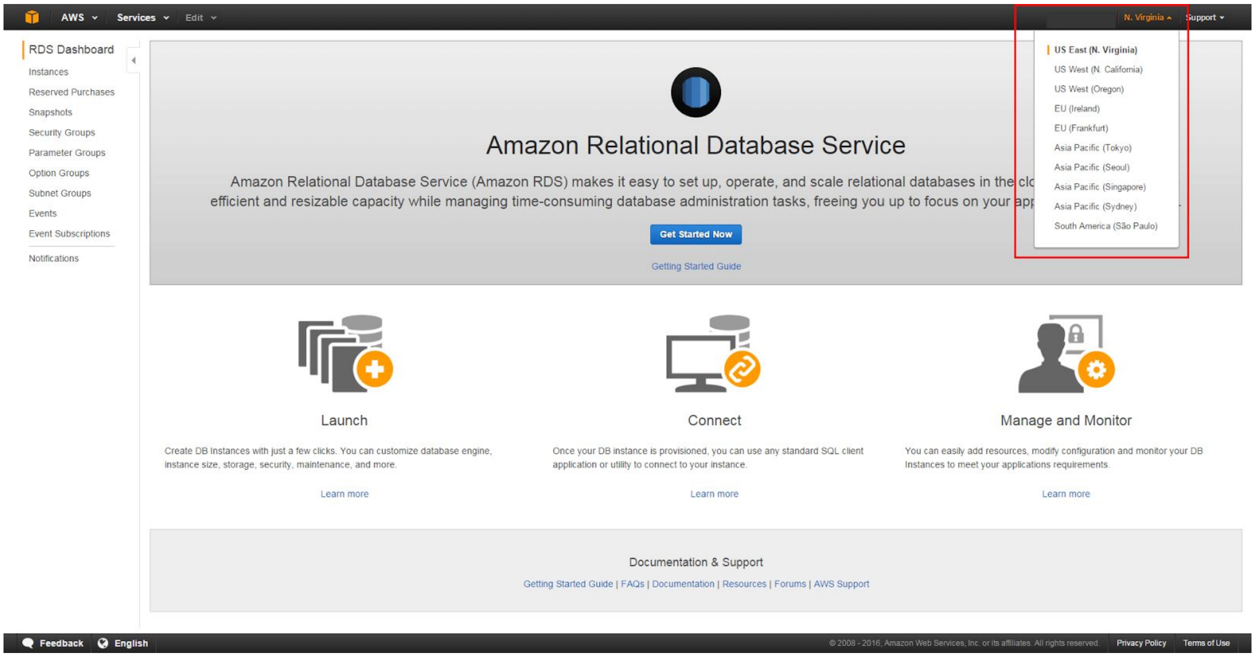
Submit feedback to tell us about your experience with the AWS Management Console.

Step 1: Create a MySQL DB Instance

In this step, we will use Amazon RDS to create a MySQL DB Instance with db.t2.micro DB instance class, 5 GB of storage, and automated backups enabled with a retention period of one day. As a reminder, all of this is free tier eligible.

a. In the top right corner of the Amazon RDS console, select the *Region* in which you want to create the DB instance.

> **Note: Amazon cloud computing resources are housed in highly available data center facilities in different areas of the world. Each Region contains multiple distinct locations called 'Availability Zones,' or AZs. You have the ability to choose which Region to host your Amazon RDS activity in.**



Amazon Relational Database Service

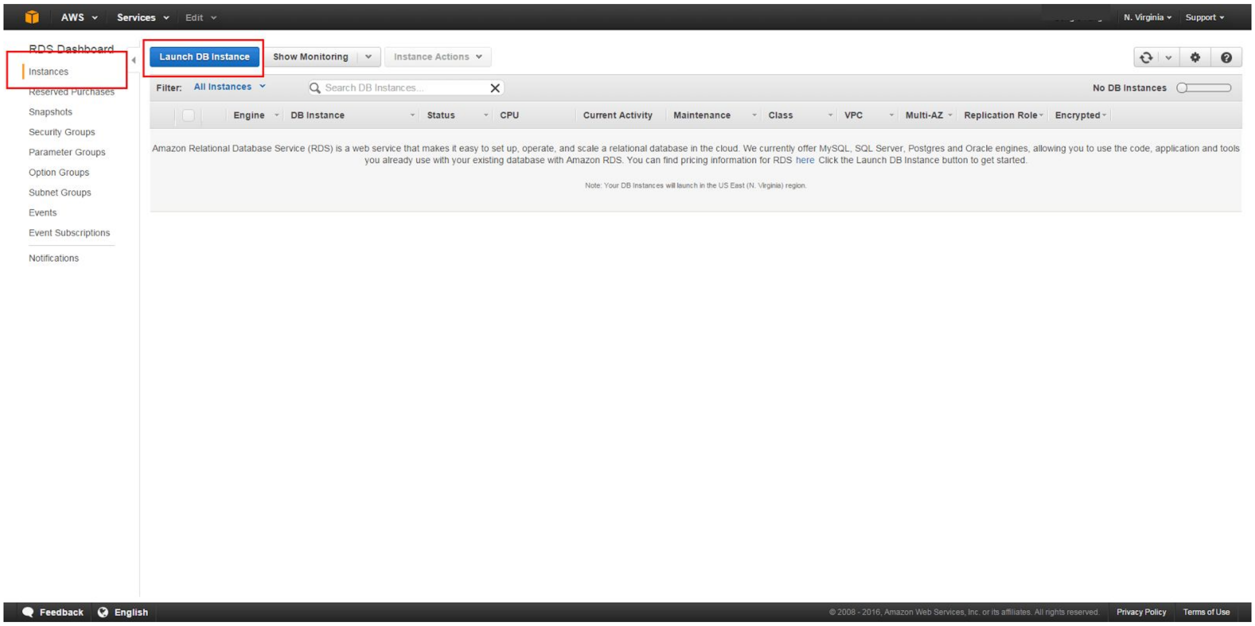
Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale relational databases in the clc efficient and resizable capacity while managing time-consuming database administration tasks, freeing you up to focus on your app

Create DB Instances with just a few clicks. You can customize database engine, instance size, storage, security, maintenance, and more.

Once your DB instance is provisioned, you can use any standard SQL client application or utility to connect to your instance.

You can easily add resources, modify configuration and monitor your DB Instances to meet your applications requirements.

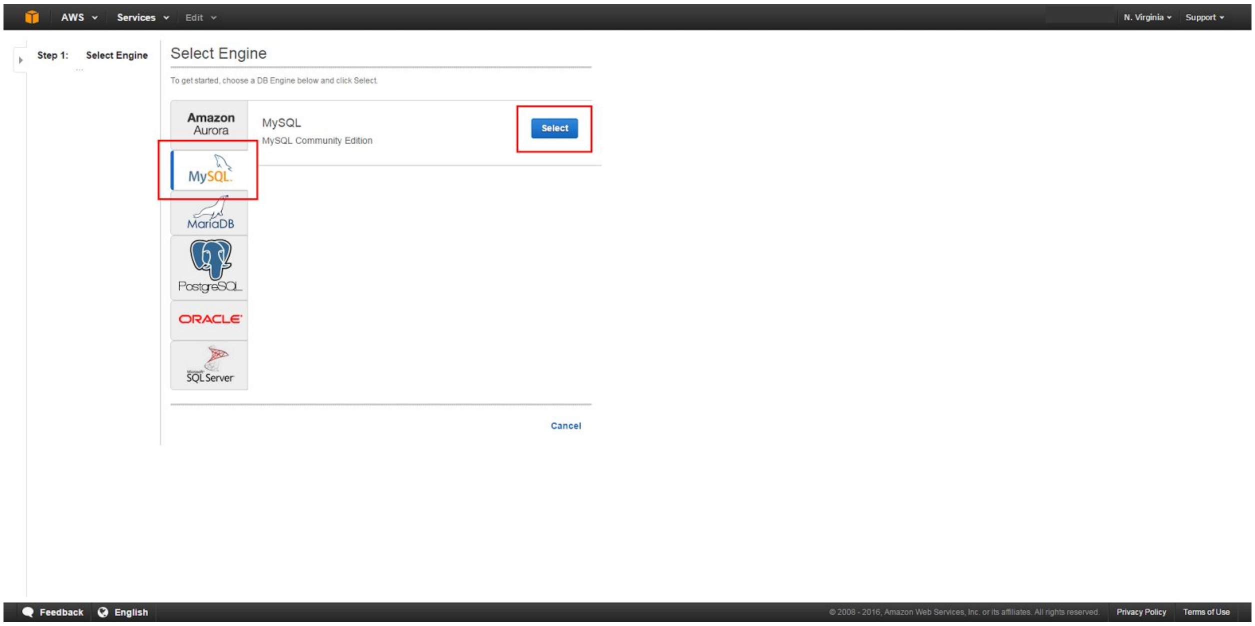
b. In the navigation pane on the left, click Instances. Then click Launch DB Instance.



Amazon Relational Database Service (RDS) is a web service that makes it easy to set up, operate, and scale a relational database in the cloud. We currently offer MySQL, SQL Server, Postgres and Oracle engines, allowing you to use the code, application and tools you already use with your existing database with Amazon RDS. You can find pricing information for RDS here Click the Launch DB Instance button to get started.

Note: Your DB Instances will launch in the US East (N. Virginia) region.

c. You now have options to select your engine. For this tutorial, click the *MySQL icon* and then click Select.



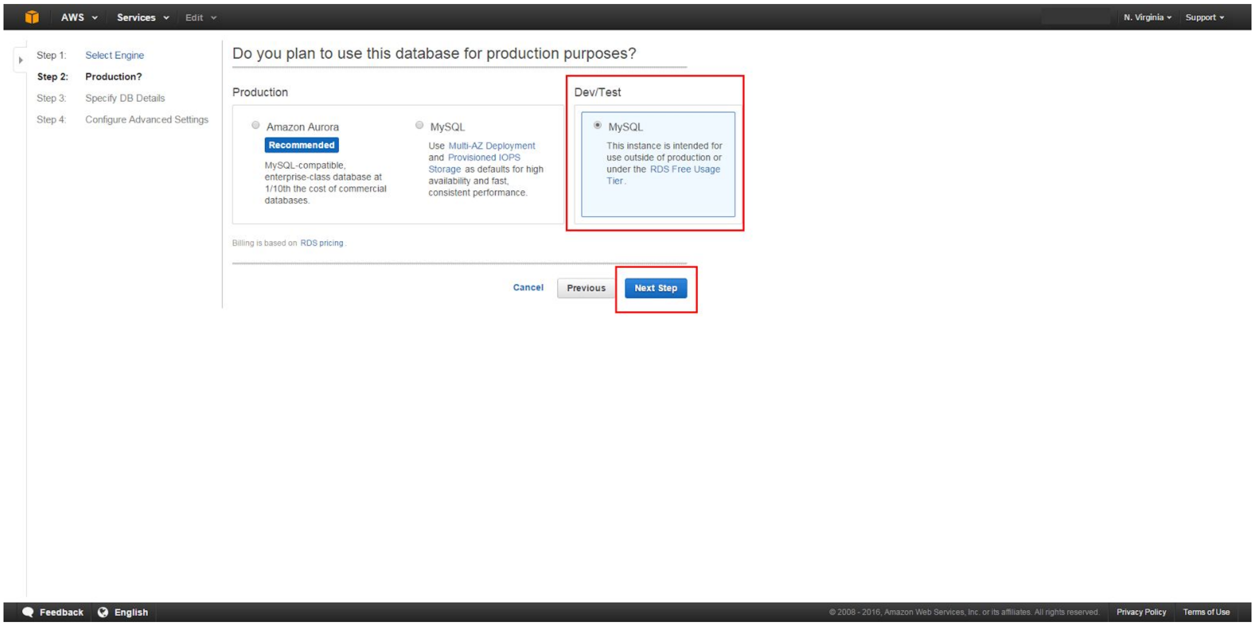
Select Engine

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

MySQL

MySQL Community Edition

d. The next screen gives you options to choose your environment. Select the MySQL option under Dev/Test and click Next Step.



Do you plan to use this database for production purposes?

○ Amazon Aurora

MySQL-compatible, enterprise-class database at 1/10th the cost of commercial databases.

○ MySQL

Use Multi-AZ Deployment and Provisioned IOPS Storage as defaults for high availability and fast, consistent performance.

● MySQL

This instance is intended for use outside of production or under the RDS Free Usage Tier.

Billing is based on RDS pricing.

e. You will now configure your DB instance. The list below shows the example settings you can use for this tutorial:

Instance Specifications:

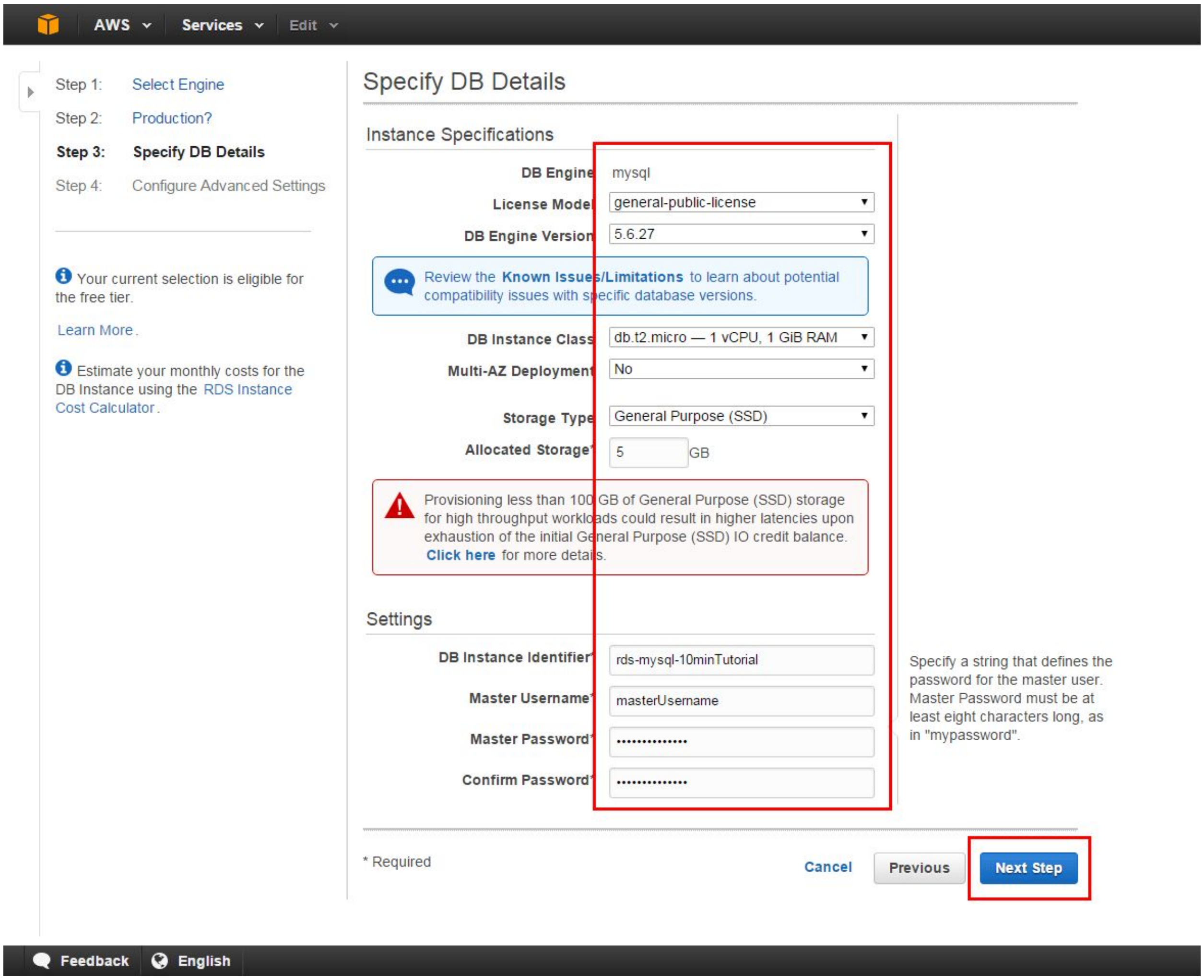
- **License Model: Select the default *general-public-license* to use the general license agreement for MySQL. MySQL has only one license model.**

- DB Engine Version: Select the default version of MySQL. Note that Amazon RDS supports multiple versions of MySQL in some Regions.
- DB Instance Class: Select *db.t2.micro --- 1vCPU, 1 GIB RAM*. This equates to 1 GB memory and 1 vCPU. To see a list of supported instance classes, see Amazon RDS Product Details.
- Multi-AZ Deployment: Select *No* to create your DB instance in a single Availability Zone for this tutorial. Using a Multi-AZ Deployment will automatically provision and maintain a synchronous standby replica in a different Availability Zone. Note that you will have to pay for Multi-AZ Deployment. For more information, see High Availability Deployment.
- Storage Type: Select *General Purpose (SSD)*. For more information about storage, see Storage for Amazon RDS.
- Allocated Storage: Type 5 to allocate 5 GB of storage for your database. You can scale up to a maximum of 6 TB with Amazon RDS for MySQL.
-

Settings:

- DB Instance Modifier: Type a name for the DB instance that is unique for your account in the Region you selected. For this tutorial, we will name it *rds-mysql-10minTutorial*.
- Master Username: Type a username that you will use to log in to your DB instance. We will use *masterUsername* in this example.
- Master Password: Type a password that contains from 8 to 41 printable ASCII characters (excluding /,", and @) for your master user password.
- Confirm Password: Retype your password
- Allocated Storage: Type 5 to allocate 5 GB of storage for your database. For more information about storage allocation, see Amazon Relational Database Service Features. (switch ordering, its after storage type)

Click Next Step



Your current selection is eligible for the free tier.

Learn More.

Estimate your monthly costs for the DB Instance using the RDS Instance Cost Calculator.

Specify DB Details

Instance Specifications

DB Engine	mysql
License Model	general-public-license
DB Engine Version	5.6.27

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

Provisioning less than 100 GB of General Purpose (SSD) storage for high throughput workloads could result in higher latencies upon exhaustion of the initial General Purpose (SSD) IO credit balance. **Click here** for more details.

Settings

DB Instance Identifier*	rds-mysql-10minTutorial
Master Username*	masterUsername
Master Password*	••••••••••••••
Confirm Password*	••••••••••••••

Specify a string that defines the password for the master user. Master Password must be at least eight characters long, as in "mypassword".

* Required