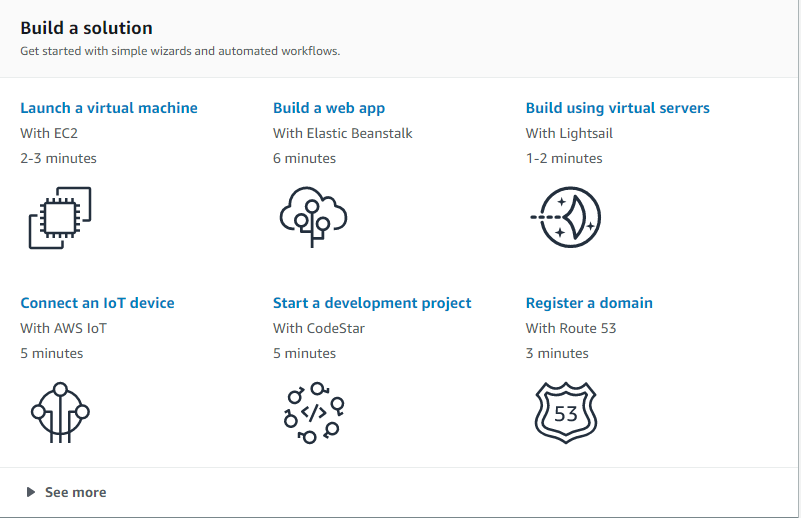
What Is Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Step1:

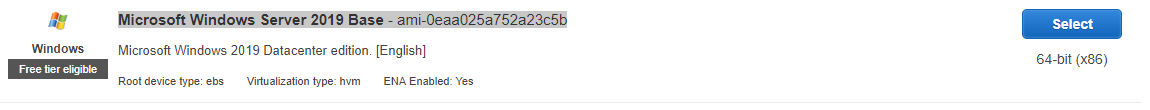
Click on Launch a virtual machine.



Step2:

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.

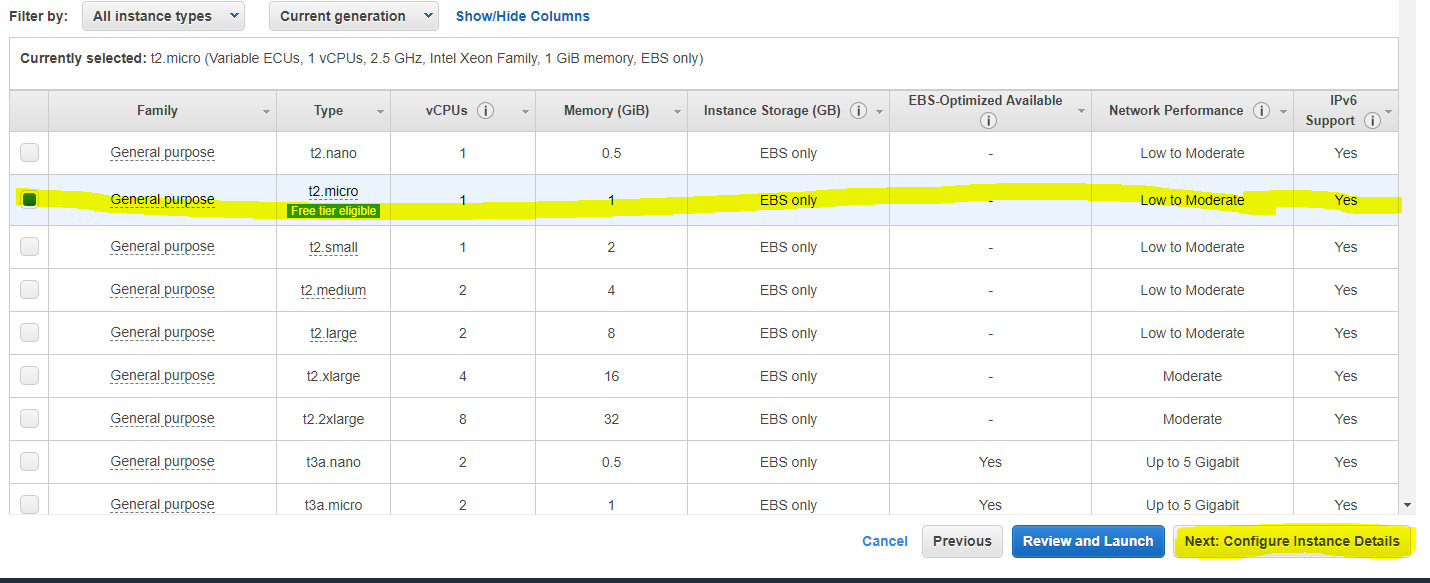
Select the **Microsoft Windows Server 2019 Base** - ami-0eaa025a752a23c5b



Step3:

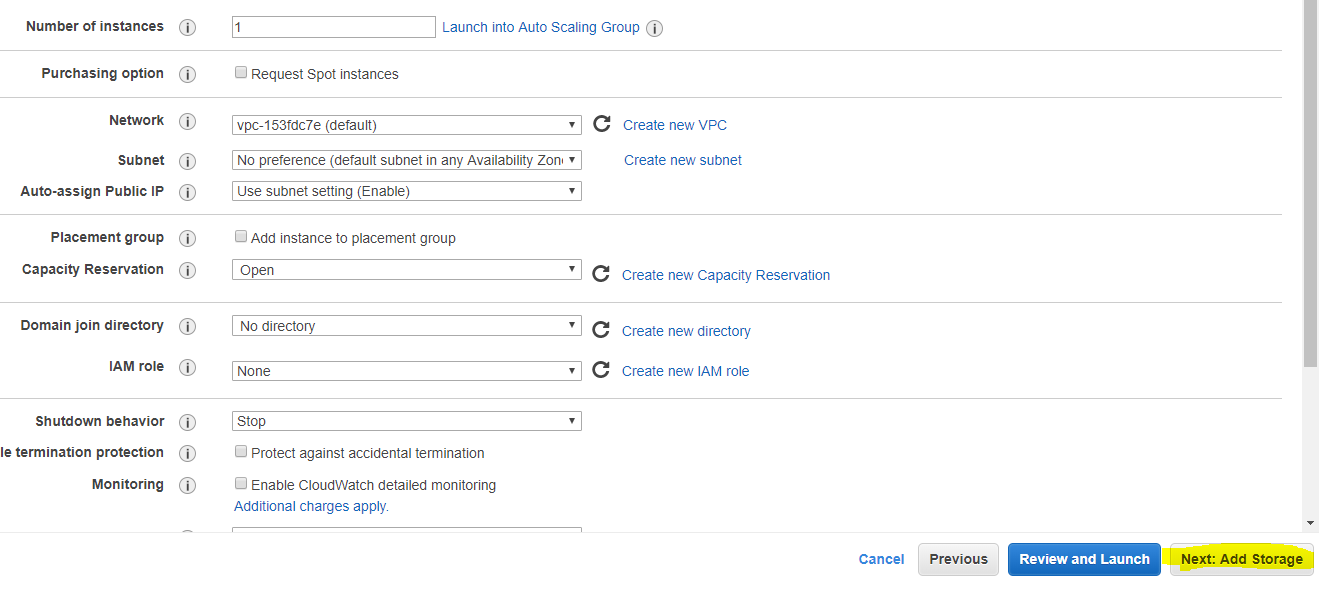
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](https://aws.amazon.com/ec2/instance-types/) about instance types and how they can meet your computing needs.

Chose an instance type and click next



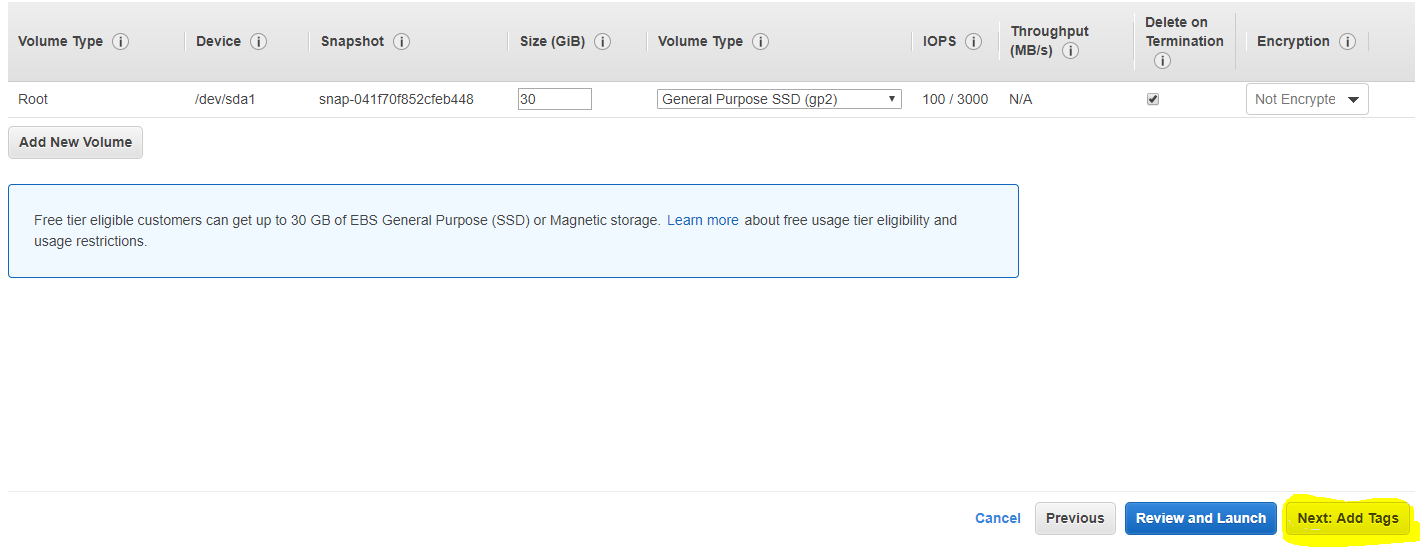
Step4:

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.



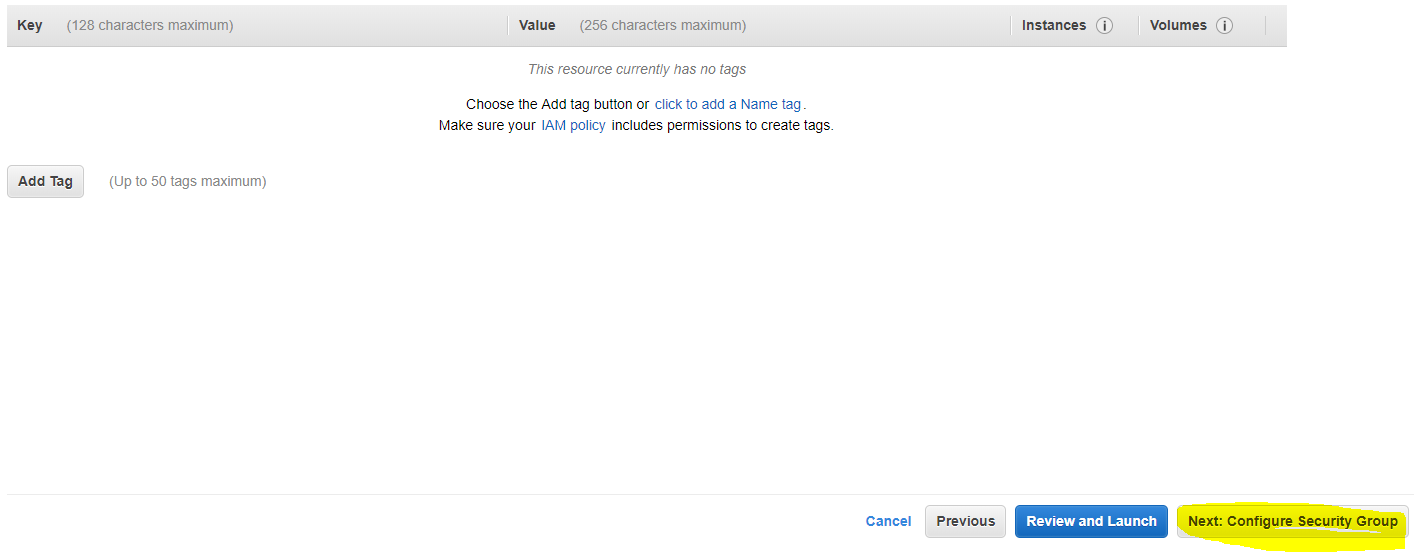
Step 5:

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](https://docs.aws.amazon.com/console/ec2/launchinstance/storage) about storage options in Amazon EC2.



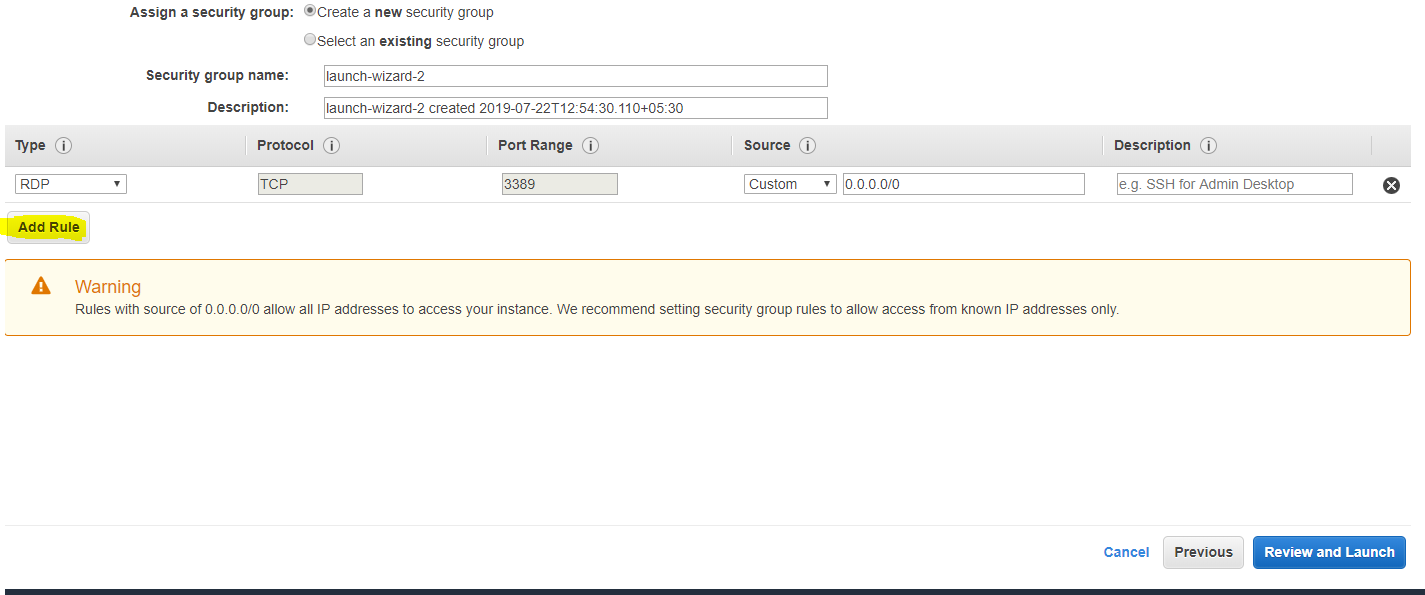
Step 6:

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.  
A copy of a tag can be applied to volumes, instances or both.  
Tags will be applied to all instances and volumes. [Learn more](https://docs.aws.amazon.com/console/ec2/tags) about tagging your Amazon EC2 resources.

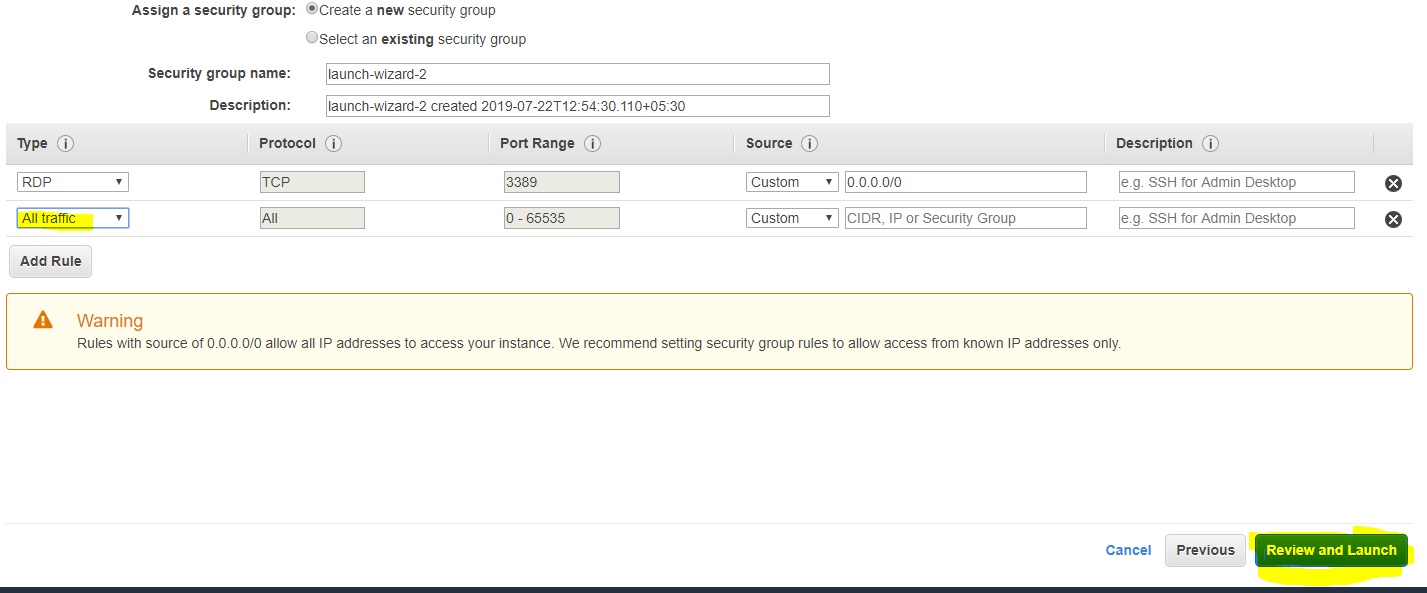


Step 7:

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](https://docs.aws.amazon.com/console/ec2/security-groups) about Amazon EC2 security groups.

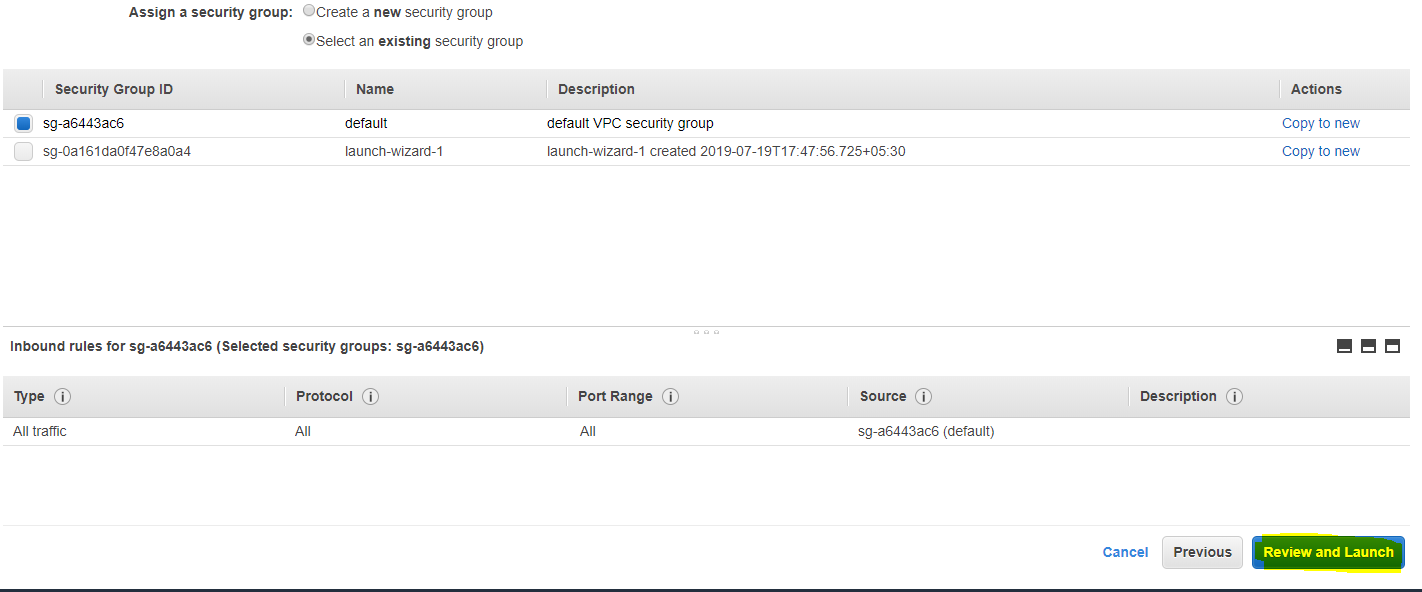


Add Rule



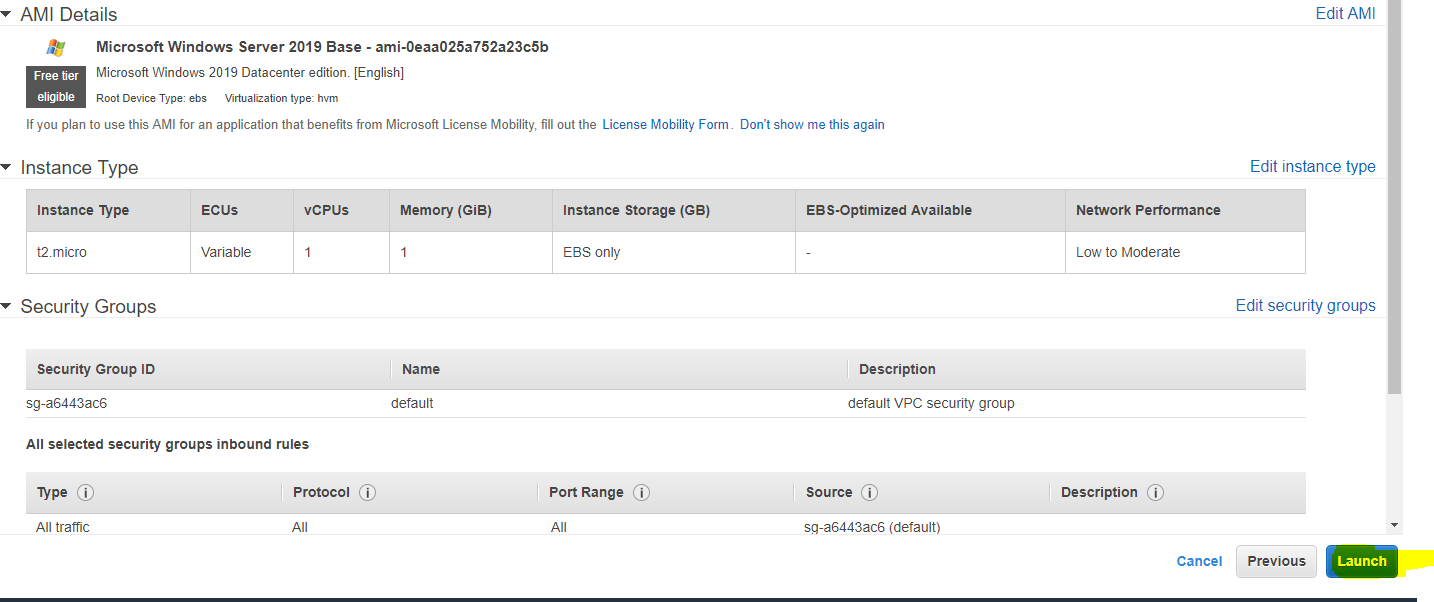
Step 8:

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](https://docs.aws.amazon.com/console/ec2/security-groups) about Amazon EC2 security groups.

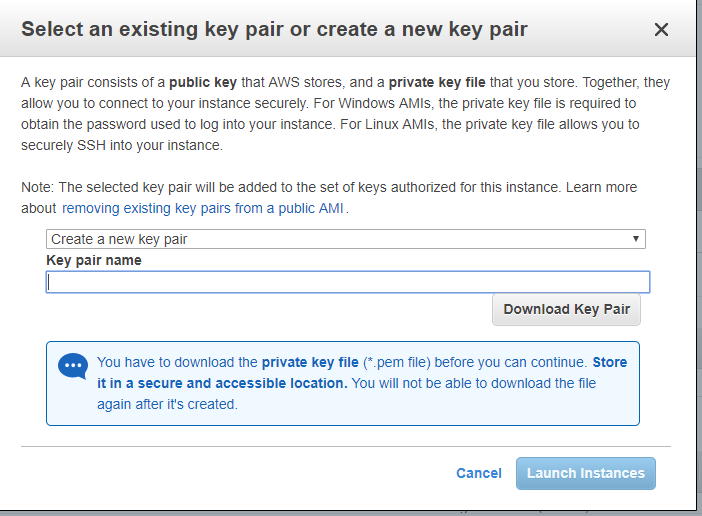


Step 9:

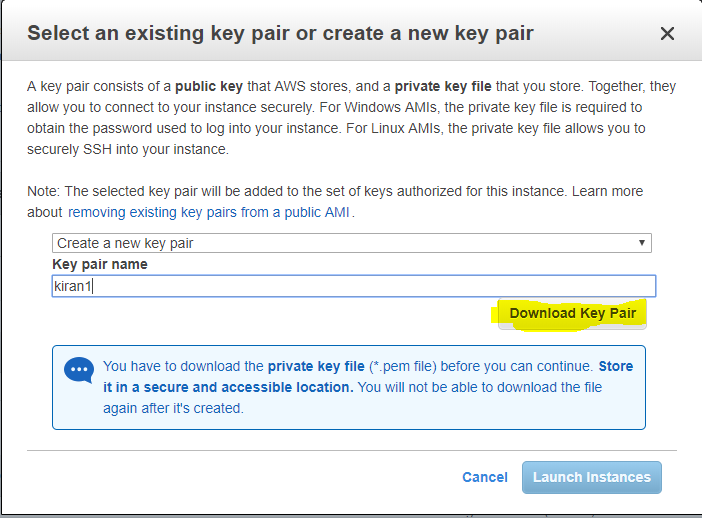
Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

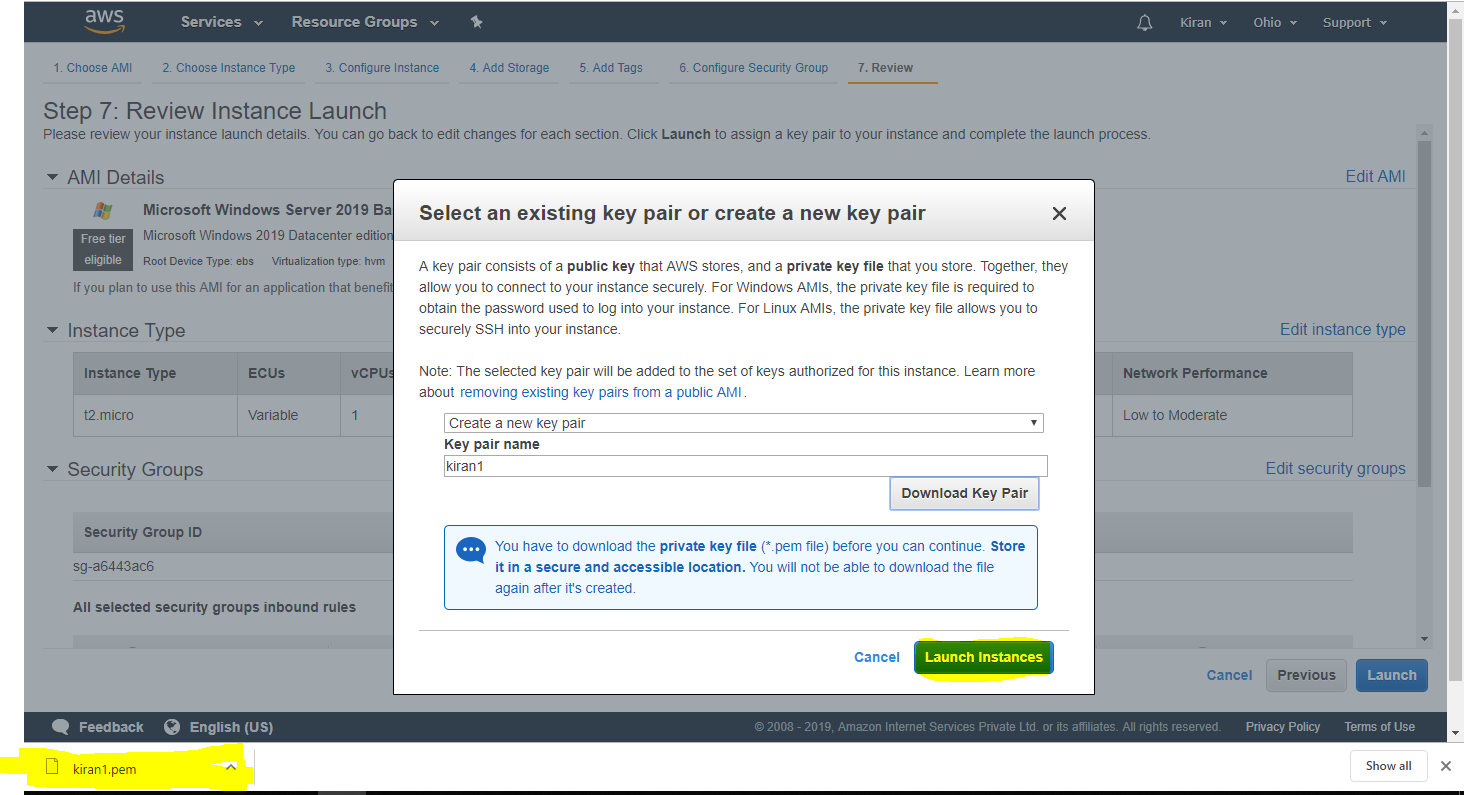


To crate key pair

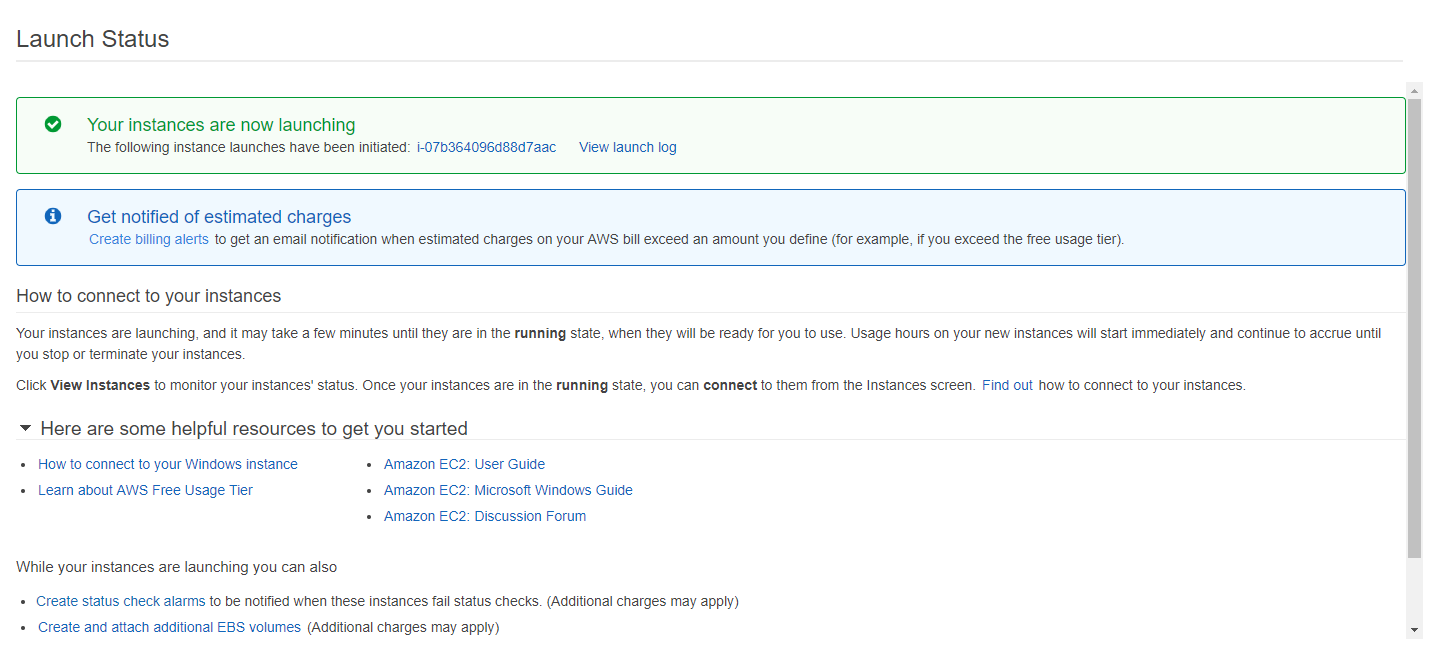


Click on Launch instances





Click on View Instances.





[User Guide for Windows Instances](https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/index.html)-(Click Link)

# What Is AWS Elastic Beanstalk?

With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS Cloud without having to learn about the infrastructure that runs those applications. Elastic Beanstalk reduces management complexity without restricting choice or control. You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring.

Elastic Beanstalk supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby. When you deploy your application, Elastic Beanstalk builds the selected supported platform version and provisions one or more AWS resources, such as Amazon EC2 instances, to run your application.

What is the use of Elastic Beanstalk?

**AWS Elastic Beanstalk** makes it even easier for developers to quickly deploy and manage applications in the **AWS** Cloud. Developers simply upload their **application**, and **Elastic Beanstalk** automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and **application** health monitoring.

What is an Elastic Beanstalk environment?

**AWS Elastic Beanstalk** makes it easy to create new **environments** for your application. You can create and manage separate **environments** for development, testing, and production use, and you can deploy any version of your application to any **environment**. ... **Elastic Beanstalk** lets you configure how deployments are performed.

What is the difference between ec2 and Elastic Beanstalk?

**Elastic Beanstalk**. **Elastic Beanstalk** is one layer of abstraction away from the **EC2**layer. **Elastic Beanstalk** will setup an "environment" for you that can contain a number of **EC2** instances, an optional database, as well as a few other **AWS**components such as a **Elastic Load Balancer**, Auto-Scaling Group, Security Group.

What is the difference between Elastic Beanstalk & Cloudformation?

**Elastic Beanstalk** is intended to make developers' lives easier. **CloudFormation** is intended to make systems engineers' lives easier. **Elastic Beanstalk** is a PaaS-like layer ontop of **AWS's** IaaS services which abstracts away the underlying EC2 instances, **Elastic** Load Balancers, auto scaling groups, etc.

Is Elastic Beanstalk free?

**Elastic Beanstalk** is a great way to get started quickly with **AWS**. ... **Elastic Beanstalk** itself is **free**, but you will incur in charges associated with resources created by EB, such as EC2 instances, ELBs or any **AWS** resource that is not **free**. My recommendation is that you start with understanding EC2 pricing.

Is AWS Elastic Beanstalk PaaS or IaaS?

**AWS Elastic** Compute Service or EC2 is **IaaS**(Infrastructure as a Service). This is because **Amazon** takes the responsibility of networking, storage, server and virtualization and the user is responsible for managing the Operating System, middleware, runtime, data and application. ... **AWS Elastic BeanStalk** is **PaaS**.

What is AWS Elastic Load Balancer?

**Elastic Load Balancing** (ELB) is a **load**-**balancing** service for Amazon Web Services (**AWS**) deployments. ELB automatically distributes incoming application traffic and scales resources to meet traffic demands

What is AWS Cloudfront?

Amazon **CloudFront** is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files, to your users.**CloudFront** delivers your content through a worldwide network of data centers called edge locations.

What is Cloudformation?

AWS **CloudFormation** is a service that helps you model and set up your Amazon Web Services resources so that you can spend less time managing those resources and more time focusing on your applications that run in AWS.

is RDS A Paas?

**RDS** is **PaaS**. AWS itself is IaaS as it provides for automated deployment of servers, CPUs, storage, and networking (i.e. the hardware). **RDS**, Beanstalk, Aurora, etc. are all **PaaS**.

What is AWS Route 53 used for?

**Amazon Route 53** is a highly available and scalable cloud Domain Name System (DNS) web service. ... Using **Amazon Route 53** Traffic Flow's simple visual editor, you can easily manage how your end-users are routed to your application's endpoints—whether in a single **AWS** region or distributed around the globe.

What service is Amazon CloudFormation?

**AWS CloudFormation** (**Amazon** Web **Services CloudFormation**)

**Amazon** Web **Services** CloudFormationis a free **service** that provides **Amazon** Web**Service** (**AWS**) customers with the tools they need to create and manage the infrastructure a particular software application requires to run on **Amazon** Web**Services**.

What does s3 stand for?

**S3** is a storage service offered by Amazon. It **stands** for simple storage service and provides cloud storage for various types of web development applications. Amazon employs the same infrastructure used by its e-commerce arm.

What supports all the AWS resources?

Examples of regional **resources** are **EC2** instances and EBS volumes. ... Global**resources** are not tied to a specific region and can be used in **all** regions. The global**resource** types that **AWS** Config **supports** are IAM users, groups, roles, and customer managed policies.

How does Elastic Beanstalk work?

**AWS Elastic Beanstalk** provides tools to automate background tasks. **Elastic Beanstalk** employs Auto Scaling and **Elastic** Load Balancing to scale and balance workloads. ... **AWS Elastic Beanstalk** actively separates the cloud from local systems in order to provide added security.

What is difference between PaaS and IaaS?

The **difference between** Saas, **Paas and Iaas**. The lowest level is infrastructure-as-a-service (**IaaS**). ... Platform as a service (**PaaS**) goes a stage further and includes the operating environment included the operating system and application services.

What is PaaS with example?

**PaaS** (Platform as a Service), as the name suggests, provides you computing platforms which typically includes operating system, programming language execution environment, database, web server etc. **Examples**: AWS Elastic Beanstalk, Windows Azure, Heroku, Force.com, Google App Engine, Apache Stratos.

Why do we use load balancer?

**Load balancing** improves responsiveness and increases availability of applications. A **load balancer** sits between the client and the server farm accepting incoming network and application traffic and distributing the traffic across multiple backend servers **using** various methods

### What you do when you see a broken build for your project in Jenkins?

I will open the console output for the broken build and try to see if any file changes were missed. If I am unable to find the issue that way, then I will clean and update my local workspace to replicate the problem on my local and try to solve it.

What is the difference between freestyle and pipeline in Jenkins?

The main **difference between** any job and a **Pipeline** Job is that the **Pipeline**Scripted job runs on the **Jenkins** master, using a lightweight executor expected to use very few resources in order to translate the **pipeline** to atomic commands that execute or send to the agents.

What is Jenkins and how it works?

**Jenkins** is an open source automation tool written in Java with plugins built for Continuous Integration purpose. **Jenkins** is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.

What is the difference between Jenkins master and slave?

**Jenkins** uses a **Master**-**Slave** architecture to manage distributed builds. In this architecture, **Master and Slave** communicate through TCP/IP protocol. Your main**Jenkins** server is the **Master**. ... A **Master** instance **of Jenkins** can also execute build jobs directly.

What is a slave node in Jenkins?

A **slave** is a Java executable that runs on a remote machine. The characteristics of the **slave** are : It hears requests from the **Jenkins** Master instance. **Slaves** can run on a variety of operating systems. The job of a **Slave** is to do as they are told to, which involves executing build jobs dispatched by the Master.