# 1 NGINX AND EC2 LAUNCH TEMPLATE

# Today's Takeaways

▶   Launching EC2 Instance

▶   EC2 Launch Template

▶   NGINX Web Server

CLARUSWAY
WAY TO REINVENT YOURSELF

# Launching EC2

**EC2**

- EC2 stands for Elastic Compute Cloud in AWS.

- EC2 is a virtual server that is physically in the AWS Data Center.

# Launching EC2
## What is EC2?

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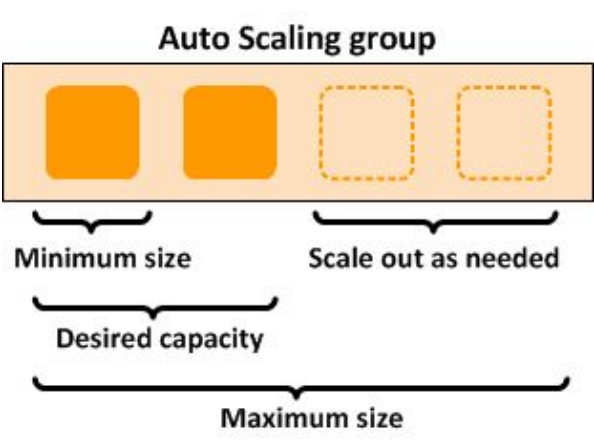


CLARUSWAY
WAY TO REINVENT YOURSELF
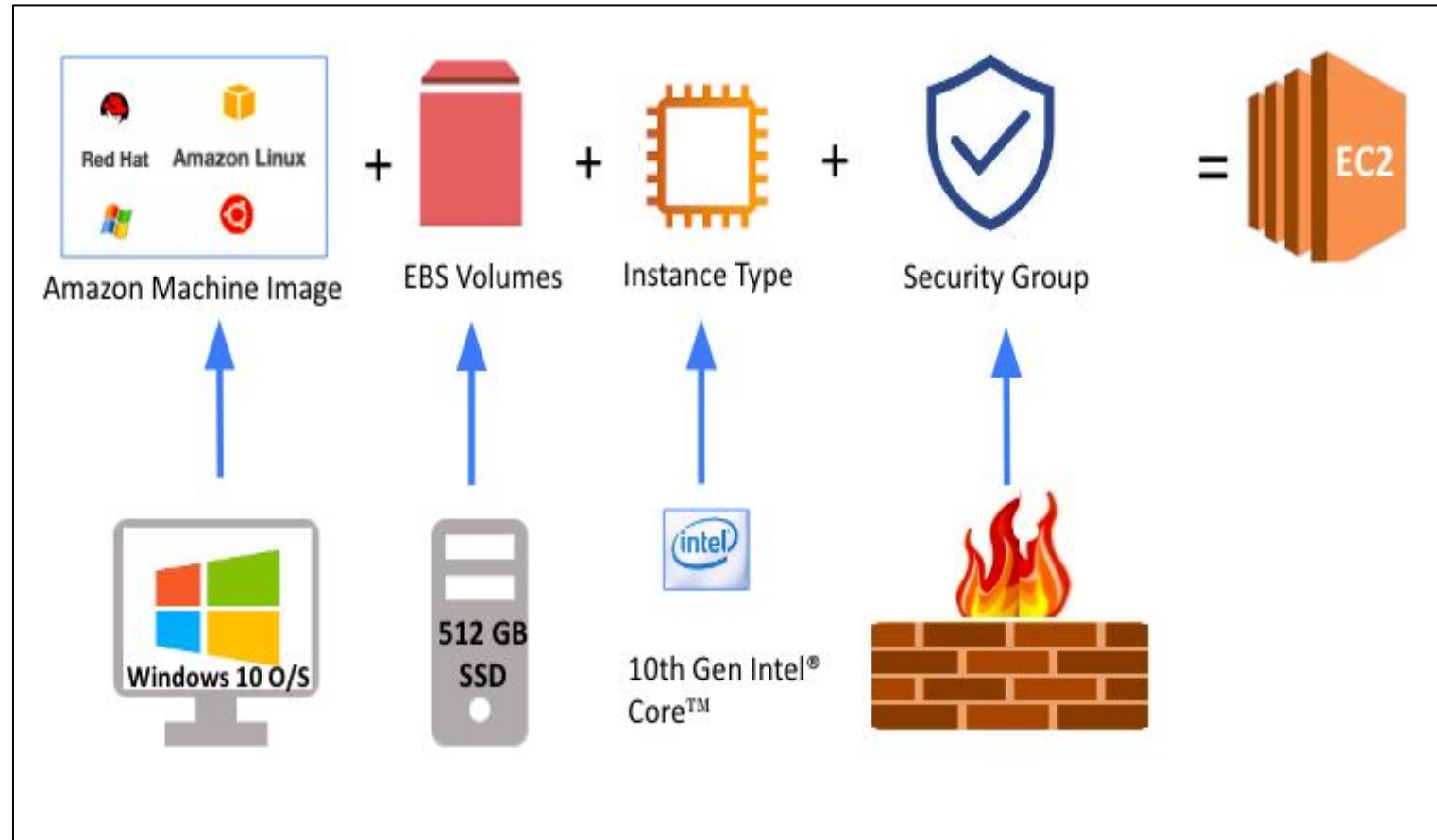
# Auto Scaling

## What is Amazon EC2 Auto Scaling?

Amazon EC2 Auto Scaling ensures that we have the correct number of Amazon EC2 instances available to handle the load for our application. We create collections of EC2 instances, called *Auto Scaling groups*. We can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that our group never goes below this size. We can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that our group never goes above this size. If we specify the desired capacity, either when we create the group or at any time thereafter, Amazon EC2 Auto Scaling ensures that our group has this many instances. If we specify scaling policies, then Amazon EC2 Auto Scaling can launch or terminate instances as demand on the application increases or decreases.

For example, the following Auto Scaling group has a minimum size of one instance, a desired capacity of two instances, and a maximum size of four instances. The scaling policies that we define adjust the number of instances, within our minimum and maximum number of instances, based on the criteria that we specify.



Auto Scaling group / Minimum size / Desired capacity / Scale out as needed / Maximum size

## Auto Scaling components

| Groups | EC2 instances are organized into *groups* so that they can be treated as a logical unit for the purposes of scaling and management. When we create a group, we can specify its minimum, maximum, and, desired number of EC2 instances. |
|---|---|
| Configuration Templates | ASG group uses a *launch template*, or a *launch configuration* (not recommended, offers fewer features), as a configuration template for its EC2 instances. We can specify information such as the AMI ID, instance type, key pair, security groups, and block device mapping for the instances. |
| Scaling Options | Amazon EC2 Auto Scaling provides several ways for we to scale wer Auto Scaling groups. For example, we can configure a group to scale based on the occurrence of specified conditions (dynamic scaling) or on a schedule. |

CLARUSWAY
WAY TO REINVENT YOURSELF

# Launch Configuration

## What is Amazon EC2 Launch Configuration?

- A launch configuration is a template that an EC2 Auto Scaling group uses to launch EC2 instances. When we create a launch configuration, we specify information for the instances such as the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups, and a block device mapping.

- If we've launched an EC2 instance before, we specify the same information in order to launch the instance. When we create an EC2 Auto Scaling group, we must specify a launch configuration.

- We can specify a launch configuration with multiple EC2 Auto Scaling groups. However, we can only specify one launch configuration for an EC2 Auto Scaling group at a time, and can't modify a launch configuration after its been created.

- Therefore, if we want to change the launch configuration for the EC2 Auto Scaling group, we must create a launch configuration and then update the EC2 Auto Scaling group with the new launch configuration.

CLARUSWAY
WAY TO REINVENT YOURSELF

# Launch Templates

## What is Amazon EC2 Launch Templates?

- A launch template is similar to a launch configuration, in that it specifies instance configuration information.

- It includes the ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and other parameters used to launch EC2 instances. However, defining a launch template instead of a launch configuration allows to have multiple versions of a launch template.

**What is EC2 Launch Template Versioning?**

- With versioning of launch templates, we can create a subset of the full set of parameters. Then, we can reuse it to create other versions of the same launch template. For example, we can create a launch template that defines a base configuration without an AMI or user data script.

- After we create the launch template, we can create a new version and add the AMI and user data that has the latest version of the application for testing. This results in two versions of the launch template.

- Storing a base configuration helps us to maintain the required general configuration parameters. We can create a new version of the launch template from the base configuration whenever we want. Can also delete the versions used for testing the application when we no longer need them.

CLARUSWAY
WAY TO REINVENT YOURSELF

# Launch Templates

## What is Amazon EC2 Launch Templates?

- Recommend approach is to use launch templates to ensure that we're accessing the latest features and improvements. Not all Amazon EC2 Auto Scaling features are available when we use launch configurations.

- For example, we cannot create an Auto Scaling group that launches both Spot and On-Demand Instances or that specifies multiple instance types. With launch templates, we can also use newer features of Amazon EC2. This includes the current generation of EBS Provisioned IOPS volumes (io2), EBS volume tagging, T2 Unlimited instances, Elastic Inference, and Dedicated Hosts, to name a few. Dedicated Hosts are physical servers with EC2 instance capacity that are dedicated to our use. While Amazon EC2 Dedicated Instances also run on dedicated hardware, the advantage of using Dedicated Hosts over Dedicated Instances is that we can bring eligible software licenses from external vendors and use them on EC2 instances.

- When we create a launch template, all parameters are optional. However, if a launch template does not specify an AMI, we cannot add the AMI when we create Auto Scaling group. If we specify an AMI but no instance type, we can add one or more instance types when we create Auto Scaling group.

CLARUSWAY
WAY TO REINVENT YOURSELF

# Launch Configuration

## Differences between Launch Template & Launch Configuration?

- Launch templates (LTs) are newer than launch configurations (LCs) and provide more options to work with. Thus, the AWS documentation recommends use of launch templates (LTs) over launch configuration (LCs):

- One of the practical key differences between LT and LC is the fact that **LC is immutable**. Once you define it, you can't edit it. Only a replacement is an option. However, a single LT can have multiple versions:

- Also LTs provide **more EC2 options** for you to configure, for example, dedicated hosting can be set only using a LT. Similarly, ability to use T2 unlimited burst credit option is only available in a LT.

CLARUSWAY
WAY TO REINVENT YOURSELF

# NGINX WEB SERVER

## What is a Web Server? What is NGINX?

The term *web server* can refer to hardware or software, or both of them working together.

1. On the hardware side, a web server is a computer that stores web server software and a website's component files. (for example, HTML documents, images, CSS stylesheets, and JavaScript files) A web server connects to the Internet and supports physical data interchange with other devices connected to the web.
2. On the software side, a web server includes several parts that control how web users access hosted files. At a minimum, this is an *HTTP server*. An HTTP server is software that understands URLs (web addresses) and HTTP (the protocol your browser uses to view webpages). An HTTP server can be accessed through the domain names of the websites it stores, and it delivers the content of these hosted websites to the end user's device.

At the most basic level, whenever a browser needs a file that is hosted on a web server, the browser requests the file via HTTP. When the request reaches the correct (hardware) web server, the (software) *HTTP server* accepts the request, finds the requested document, and sends it back to the browser, also through HTTP. (If the server doesn't find the requested document, it returns a 404 response instead.)

**NGINX**, which is pronounced as **'engine ex,'** is a widely used open-source web server software. It was a project started by Igor Sysoev in 2002. It got officially released in October 2004. It was created to solve the problem of C10k, which is defined as a challenge to manage the ten thousand connections all at the same time.

CLARUSWAY
WAY TO REINVENT YOURSELF
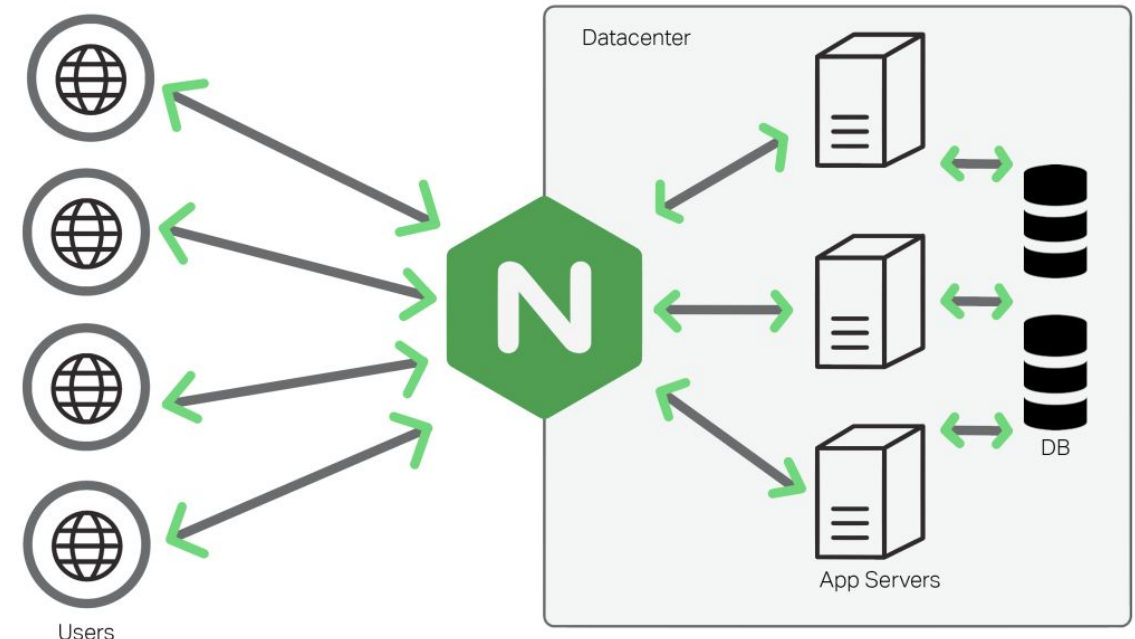
# EC2 Launch Template

# THANKS!

## Any questions?

You can find me at :

- ▸ @sumod
- ▸ sumod@clarusway.com

CLARUSWAY
WAY TO REINVENT YOURSELF