**Week-4 Day-1&2 Notes**

(High Availability and Scalability in Cloud - High Availability and Scalability - Elastic Load Balancer (ELB) and Auto Scaling Groups (ASG) - Classic Load Balancer (CLB) - Lab - CLB - Application Load Balancer (ALB) - Lab – ALB- Network Load Balancer (NLB) - Lab - NLB - Gateway Load Balancer (GWLB) - Elastic Load Balancer - Sticky Sessions - Elastic Load Balancer - Cross Zone Load Balancing- Elastic Load Balancer - SSL Certificates - Elastic Load Balancer - Connection Draining)

**High Availability and Scalability**

**High Availability**

* High availability (HA) means the application remains available with no interruption.
* We achieve high availability when an application continues to operate when one or more underlying components fail. For example, a router, switch, firewall, or server that fails.

**Scalability**

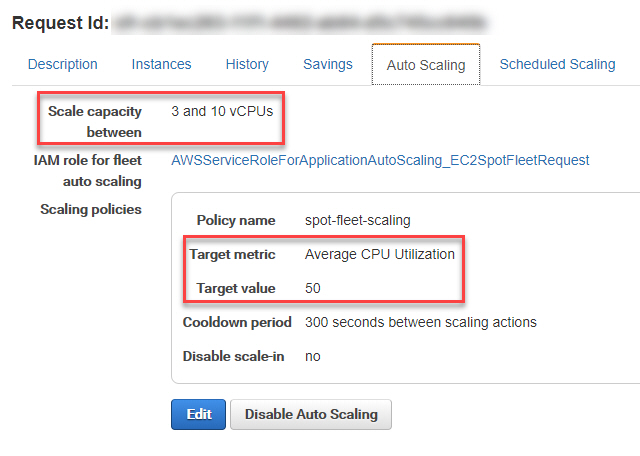
Scalability is the ability of a system to handle the increase in demand without impacting the application’s performance or availability.

When the demand is too high and there are not enough resources, then it impacts performance. There are two types of scalability:

* Vertical: scale up or down:
  + Add or remove resources:
    - CPU
    - Memory
    - Storage
* Horizontal: scale out or in:
  + Add or remove systems

For example, we can increase the number of CPU cores and memory in a web server (vertical) or we can increase the number of web servers (horizontal).

Here is a screenshot of the Amazon AWS auto scaling policy we use for networklessons.com web servers:



**Elastic Load Balancer (ELB) and Auto Scaling Groups (ASG)**

Elastic Load Balancing automatically distributes your incoming application traffic across all the EC2 instances that you are running. Elastic Load Balancing helps to manage incoming requests by optimally routing traffic so that no one instance is overwhelmed.

To use Elastic Load Balancing with your Auto Scaling group, [attach the load balancer to your Auto Scaling group](https://docs.aws.amazon.com/autoscaling/ec2/userguide/attach-load-balancer-asg.html). This registers the group with the load balancer, which acts as a single point of contact for all incoming web traffic to your Auto Scaling group.

**Auto Scaling group(ASG)**

An Auto Scaling group contains a collection of EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management. An Auto Scaling group also lets you use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies.

The size of an Auto Scaling group depends on the number of instances that you set as the desired capacity. You can adjust its size to meet demand, either manually or by using automatic scaling.

An Auto Scaling group starts by launching enough instances to meet its desired capacity. It maintains this number of instances by performing periodic health checks on the instances in the group. The Auto Scaling group continues to maintain a fixed number of instances even if an instance becomes unhealthy. If an instance becomes unhealthy, the group terminates the unhealthy instance and launches another instance to replace it.

**To attach an existing load balancer as you are creating a new Auto Scaling group**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>, and choose **Auto Scaling Groups** from the navigation pane.
2. Choose **Create Auto Scaling group**.
3. In steps 1 and 2, choose the options as desired and proceed to **Step 3: Configure advanced options**.
4. For **Load balancing**, choose **Attach to an existing load balancer**.
5. Under **Attach to an existing load balancer**, do one of the following:
   1. For Application Load Balancers, Network Load Balancers, and Gateway Load Balancers:

Choose **Choose from your load balancer target groups**, and then choose a target group in the **Existing load balancer target groups** field.

* 1. For Classic Load Balancers:

Choose **Choose from Classic Load Balancers**, and then choose your load balancer in the **Classic Load Balancers** field.

1. Proceed to create the Auto Scaling group. Your instances will be automatically registered to the load balancer after the Auto Scaling group has been created.

**To detach a load balancer from a group**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>, and choose **Auto Scaling Groups** from the navigation pane.
2. Select the check box next to an existing group.

A split pane opens up in the bottom of the **Auto Scaling groups** page.

1. On the **Details** tab, choose **Load balancing**, **Edit**.
2. Under **Load balancing**, do one of the following:
   1. For **Application, Network or Gateway Load Balancer target groups**, choose the delete (X) icon next to the target group.
   2. For **Classic Load Balancers**, choose the delete (X) icon next to the load balancer.
3. Choose **Update**.

**Classic Load Balancer**

Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. Classic Load Balancer is intended for applications that are built within the EC2-Classic network.

**Before you begin**

* Complete the steps in [Prepare your VPC and EC2 instances](https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-backend-instances.html#set-up-ec2).
* Launch the EC2 instances that you plan to register with your load balancer. Ensure that the security groups for these instances allow HTTP access on port 80.
* Install a web server, such as Apache or Internet Information Services (IIS), on each instance, enter its DNS name into the address field of an internet-connected web browser, and verify that the browser displays the default page of the server.

**To create a Classic Load Balancer**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation bar, choose a Region for your load balancer. Be sure to select the same Region that you selected for your EC2 instances.
3. On the navigation pane, under **LOAD BALANCING**, choose **Load Balancers**.
4. Choose **Create Load Balancer**.
5. For **Classic Load Balancer**, choose **Create**.

**Application Load Balancer**

The Application Load Balancer is a feature of Elastic Load Balancing that allows a developer to configure and route incoming end-user traffic to applications based in the AWS public cloud.

Application Load Balancer can control which client requests are served by your applications.

An Application Load Balancer functions at the application layer, the seventh layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action. You can configure listener rules to route requests to different target groups based on the content of the application traffic. Routing is performed independently for each target group, even when a target is registered with multiple target groups. You can configure the routing algorithm used at the target group level. The default routing algorithm is round robin; alternatively, you can specify the least outstanding requests routing algorithm.

Application Load Balancer is particularly useful for websites and mobile apps running in containers or on AWS EC2 instances.

**Tasks**

* [Step 1: Configure a target group](https://docs.aws.amazon.com/elasticloadbalancing/latest/application/create-application-load-balancer.html#configure-target-group)
* [Step 2: Register targets](https://docs.aws.amazon.com/elasticloadbalancing/latest/application/create-application-load-balancer.html#select-targets)
* [Step 3: Configure a load balancer and a listener](https://docs.aws.amazon.com/elasticloadbalancing/latest/application/create-application-load-balancer.html#configure-load-balancer)
* [Step 4: Test the load balancer](https://docs.aws.amazon.com/elasticloadbalancing/latest/application/create-application-load-balancer.html#test-load-balancer)

**To configure your target group**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the left navigation pane, under **Load Balancing**, choose **Target Groups**.
3. Choose **Create target group**.
4. In the **Basic configuration** section, set the following parameters:
   1. For **Choose a target type**, select **Instance** to specify targets by instance ID or **IP addresses** to specify targets by IP address. If the target type is a **Lambda function**, you can enable health checks by selecting **Enable** in the **Health checks** section.
   2. For **Target group name**, enter a name for the target group.
   3. Modify the **Port** and **Protocol** as needed.
   4. If the target type is **IP addresses**, choose **IPv4** or **IPv6** as the **IP address type**, otherwise skip to the next step.
   5. For VPC, select a virtual private cloud (VPC) with the targets that you want to include in your target group.
   6. For **Protocol version**, select **HTTP1** when the request protocol is HTTP/1.1 or HTTP/2; select **HTTP2**, when the request protocol is HTTP/2 or gRPC; and select **gRPC**, when the request protocol is gRPC.
5. In the **Health checks** section, modify the default settings as needed. For **Advanced health check settings**, choose the health check port, count, timeout, interval, and specify success codes. If health checks consecutively exceed the **Unhealthy threshold** count, the load balancer takes the target out of service. If health checks consecutively exceed the **Healthy threshold** count, the load balancer puts the target back in service.
6. (Optional) Add one or more tags as follows:
   1. Expand the **Tags** section.
   2. Choose **Add tag**.
   3. Enter the tag **Key** and tag **Value**. Allowed characters are letters, spaces, numbers (in UTF-8), and the following special characters: + - = . \_ : / @. Do not use leading or trailing spaces. Tag values are case-sensitive.
7. Choose **Next**.

**To register targets:**

1. In the **Register targets** page, add one or more targets as follows:
   * If the target type is **Instances**, select one or more instances, enter one or more ports, and then choose **Include as pending below**.
   * If the target type is **IP addresses**, do the following:
     1. Select a network **VPC** from the list, or choose **Other private IP addresses**.
     2. Enter the IP address manually, or find the IP address using instance details. You can enter up to five IP addresses at a time.
     3. Enter the ports for routing traffic to the specified IP addresses.
     4. Choose **Include as pending below**.
   * If the target type is **Lambda**, select a Lambda function, or enter a Lambda function ARN, and then choose **Include as pending below**.
2. Choose **Create target group**.

**To configure your load balancer and listener**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Choose **Create Load Balancer**.
4. Under **Application Load Balancer**, choose **Create**.
5. **Basic configuration**
   1. For **Load balancer name**, enter a name for your load balancer. For example, **my-alb**. The name of your Application Load Balancer must be unique within your set of Application Load Balancers and Network Load Balancers for the Region.
   2. For **Scheme**, choose **Internet-facing** or **Internal**.
   3. For **IP address type**, choose **IPv4** or **Dualstack**. Use **IPv4** if your clients use IPv4 addresses to communicate with the load balancer. Choose **Dualstack** if your clients use both IPv4 and IPv6 addresses to communicate with the load balancer.
6. **Network mapping**
   1. For **VPC**, select the VPC that you used for your EC2 instances. If you selected **Internet-facing** for **Scheme**, only VPCs with an internet gateway are available for selection.
   2. For **Mappings**, select two or more Availability Zones and corresponding subnets. Enabling multiple Availability Zones increases the fault tolerance of your applications.

Select one subnet per zone to enable. If you enabled **Dualstack** mode for the load balancer, select subnets with associated IPv6 CIDR blocks. You can specify one of the following:

* + - Subnets from two or more Availability Zones
    - Subnets from one or more Local Zones
    - One Outpost subnet

1. For **Security groups**, select an existing security group, or create a new one.

(Optional) To create a new security group for your load balancer, choose **Create a new security group**.

1. For **Listeners and routing**, the default listener accepts HTTP traffic on port 80. You can optionally choose **Add listener** to add another listener (for example, an HTTPS listener).

If you create an HTTPS listener, configure the required **Secure listener settings**. Otherwise, go to the next step.

When you use HTTPS for your load balancer listener, you must deploy an SSL certificate on your load balancer. The load balancer uses this certificate to terminate the connection and decrypt requests from clients before sending them to the targets. Additionally, specify the security policy that the load balancer uses to negotiate SSL connections with the clients.

For **Default SSL certificate**, do one of the following:

* 1. If you created or imported a certificate using AWS Certificate Manager, select **From ACM**, and then select the certificate.
  2. If you uploaded a certificate using IAM, select **From IAM**, and then select the certificate.
  3. If you want to import a certificate to ACM or IAM , enter a certificate name. Then, paste the PEM-encoded private key and body.

1. (Optional) You can use **Add-on services**, such as the **AWS Global Accelerator** to create an accelerator and associate the load balancer with the accelerator.
2. **Tag and create**

(Optional) Add a tag to categorize your load balancer. Tag keys must be unique for each load balancer.

* 1. Review your configuration, and choose **Create load balancer**.

**To test the load balancer**

1. After the load balancer is created, choose **Close**.
2. In the navigation pane, under **Load Balancing**, choose **Target Groups**.
3. Select the newly created target group.
4. Choose **Targets** and verify that your instances are ready. If the status of an instance is initial, it's typically because the instance is still in the process of being registered.
5. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
6. Select the newly created load balancer.
7. Choose **Description** and copy the DNS name of the load balancer (for example, my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com). Paste the DNS name into the address field of an internet-connected web browser. If everything is working, the browser displays the default page of your server.

**Network Load Balancer (NLB):**

AWS Network Load Balancer (NLB) is an Amazon Web Services ([AWS](https://www.techtarget.com/searchaws/definition/Amazon-Web-Services)) tool that distributes end user traffic across multiple cloud resources to ensure low latency and high throughput for applications. Amazon NLB manages Transmission Control Protocol ([TCP](https://www.techtarget.com/searchnetworking/definition/TCP)) traffic at [Layer 4](https://www.techtarget.com/searchnetworking/definition/Transport-layer) of the Open Systems Interconnection ([OSI](https://www.techtarget.com/searchnetworking/definition/OSI)) reference model. AWS designed the Network Load Balancer to handle millions of end user requests per second and unpredictable spikes in end user traffic to ensure high availability for cloud applications.

Network Load Balancer performs health checks on what it calls "targets" -- which can be an [EC2 instance](https://www.techtarget.com/searchaws/definition/Amazon-EC2-instances), IP address, microservice or container -- to ensure that it routes traffic to high-performing resources. When a target becomes slow or unavailable, the Network Load Balancer routes traffic to another target.

**To configure your target group**

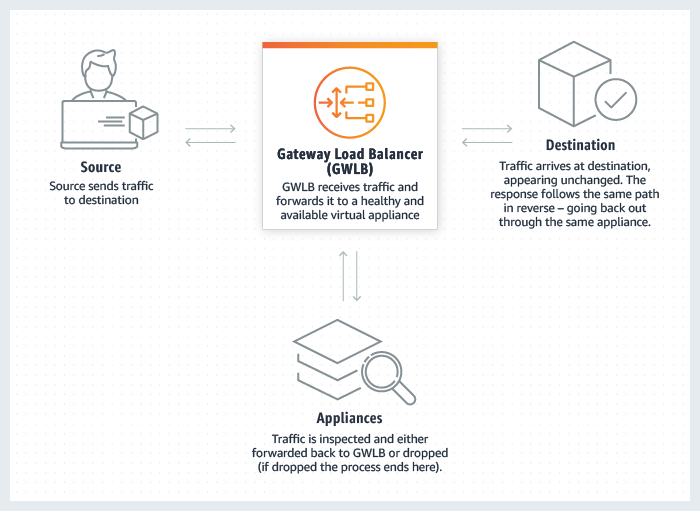
1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Load Balancing**, choose **Target Groups**.
3. Choose **Create target group**.
4. Keep **Target type** as instance.
5. For **Target group name**, enter a name for the new target group.
6. Keep **Protocol** as TCP, and **Port** as 80.
7. Select the **VPC** containing your instances. Keep the protocol version as **HTTP1**.
8. For **Health checks**, keep the default settings.
9. Choose **Next**.
10. On the **Register targets** page, complete the following steps. This is an optional step to create a target group. However, you must register your targets if you want to test your load balancer and ensure that it is routing traffic to your targets.
    1. For **Available instances**, select one or more instances.
    2. Keep the default port 80, and choose **Include as pending below**.
11. Choose **Create target group**.

**To create a Network Load Balancer**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation bar, choose a Region for your load balancer. Be sure to choose the same Region that you used for your EC2 instances.
3. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
4. Choose **Create Load Balancer**.
5. For **Network Load Balancer**, choose **Create**.

**Gateway Load Balancer(GWLB)**

Gateway Load Balancer helps you easily deploy, scale, and manage your third-party virtual appliances. It gives you one gateway for distributing traffic across multiple virtual appliances while scaling them up or down, based on demand. This decreases potential points of failure in your network and increases availability.  
  
You can find, test, and buy virtual appliances from third-party vendors directly in AWS Marketplace. A Gateway Load Balancer operates at the third layer of the Open Systems Interconnection (OSI) model, the network layer. It listens for all IP packets across all ports and forwards traffic to the target group that's specified in the listener rule.



**Sticky sessions**

The key to managing sticky sessions is determining how long your load balancer should consistently route the user's request to the same target. If your application has its own session cookie, then you can use application-based stickiness and the load balancer session cookie follows the duration specified by the application's session cookie.

**To enable duration-based sticky sessions for a load balancer using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select your load balancer.
4. On the **Description** tab, choose **Edit stickiness**.
5. On the **Edit stickiness**page, select **Enable load balancer generated cookie stickiness**.
6. (Optional) For **Expiration Period**, type the cookie expiration period, in seconds. If you do not specify an expiration period, the sticky session lasts for the duration of the browser session.
7. Choose **Save**.

If an instance fails or becomes unhealthy, the load balancer stops routing requests to that instance, and chooses a new healthy instance based on the existing load balancing algorithm. The load balancer treats the session as now "stuck" to the new healthy instance, and continues routing requests to that instance even if the failed instance comes back.

**To enable application-controlled session stickiness using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select your load balancer.
4. On the **Description** tab, choose **Edit stickiness**.
5. On the **Edit stickiness** page, select **Enable application generated cookie stickiness**.
6. For **Cookie Name**, type the name of your application cookie.
7. Choose **Save**.

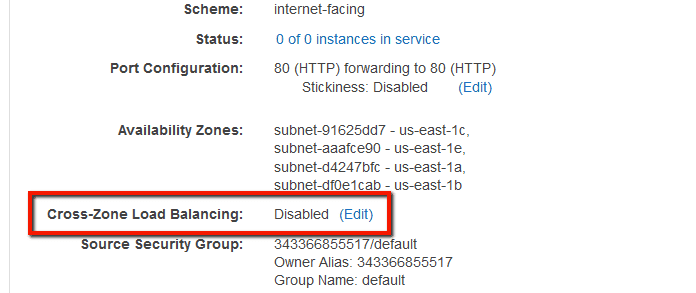
**Elastic Load Balancer - Cross Zone Load Balancing**

With cross-zone load balancing, each load balancer node for your Classic Load Balancer distributes requests evenly across the registered instances in all enabled Availability Zones. If cross-zone load balancing is disabled, each load balancer node distributes requests evenly across the registered instances in its Availability Zone only.

Cross-zone load balancing reduces the need to maintain equivalent numbers of instances in each enabled Availability Zone, and improves your application's ability to handle the loss of one or more instances.

**To enable cross-zone load balancing using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select your load balancer.
4. On the **Description** tab, choose **Change cross-zone load balancing setting**.
5. On the **Configure Cross-Zone Load Balancing** page, select **Enable**.
6. Choose **Save**.



**To disable cross-zone load balancing using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select your load balancer.
4. On the **Description** tab, choose **Change cross-zone load balancing**.
5. On the **Configure Cross-Zone Load Balancing** page, select **Disable**.
6. Choose **Save**.

To disable cross-zone load balancing, set the CrossZoneLoadBalancing attribute of your load balancer to false.

**Elastic Load Balancer - SSL Certificates**

The load balancer uses the certificate to terminate the connection and then decrypt requests from clients before sending them to the instances.

The SSL and TLS protocols use an X.509 certificate (SSL/TLS server certificate) to authenticate both the client and the back-end application. An X.509 certificate is a digital form of identification issued by a certificate authority (CA) and contains identification information, a validity period, a public key, a serial number, and the digital signature of the issuer.

You can create a certificate using AWS Certificate Manager or a tool that supports the SSL and TLS protocols, such as OpenSSL. You will specify this certificate when you create or update an HTTPS listener for your load balancer. When you create a certificate for use with your load balancer, you must specify a domain name.

Follow these steps to associate an ACM SSL certificate with your load balancer.

Note: ACM certificates must be requested or imported in the same AWS Region as your Load Balancer.

### **Associate an ACM SSL certificate with a Classic Load Balancer**

1. Open the [Amazon EC2 console](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose Load Balancers. Then, choose your Classic Load Balancer.
3. Choose the Listeners tab, and then choose Edit.
4. For Load Balancer Protocol, choose HTTPS.
5. For SSL Certificate, choose Change.
6. Select Choose a certificate from ACM.
7. Select the certificate from Certificates dropdown list, and then choose Save.

### **Associate an ACM SSL certificate with an Application Load Balancer**

1. Open the [Amazon EC2 console](https://console.aws.amazon.com/ec2/).
2. In the navigation pane, choose Load Balancers, and then choose your Application Load Balancer.
3. Choose Add listener.
4. For Protocol, choose HTTPS.
5. For port, choose 443.
6. For Default action(s), choose Forward to, and then select your ALB target group from the dropdown list.
7. For Default SSL certificate, choose From ACM (recommended) and then choose the ACM certificate.
8. Choose Save.

**Note:** Application Load Balancers support multiple SSL/TLS certificates using Server Name Identification (SNI). If you request a public certificate from ACM, you can't export private keys for ACM issued public certificates. You can't directly install Amazon-issued certificates on Amazon Elastic Compute Cloud (EC2) instances. Instead, use the certificate with a load balancer, and then register the EC2 instance behind the load balancer.

**Elastic Load Balancer - Connection Draining**

To ensure that a Classic Load Balancer stops sending requests to instances that are de-registering or unhealthy, while keeping the existing connections open, use connection draining. This enables the load balancer to complete in-flight requests made to instances that are de-registering or unhealthy.

When you enable connection draining, you can specify a maximum time for the load balancer to keep connections alive before reporting the instance as de-registered. The maximum timeout value can be set between 1 and 3,600 seconds (the default is 300 seconds). When the maximum time limit is reached, the load balancer forcibly closes connections to the de-registering instance.

**To enable connection draining using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select your load balancer.
4. On the **Instances** tab, for **Connection Draining**, choose **(Edit)**.
5. On the **Configure Connection Draining** page, select **Enable Connection Draining**.
6. (Optional) For **Timeout**, type a value between 1 and 3,600 seconds.
7. Choose **Save**.

**To disable connection draining using the console**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select your load balancer.
4. On the **Instances** tab, for **Connection Draining**, choose **(Edit)**.
5. On the **Configure Connection Draining** page, clear **Enable Connection Draining**.
6. Choose **Save**.