**Automatic scaling and load balancing with EC2 and ALB**

Created by[Neal Davis | AWS Certified Solutions Architect & Developer](https://deloittedevelopment.udemy.com/user/63f4a578-c67a-456b-916c-ddadf73e9a26/)

**AWS Workspace**

**60-120 minutes**

In this lab, you will be a Cloud Practitioner at a company that is considering moving applications to the cloud. Your boss is interested in learning about the value proposition of the cloud, especially in relation to

The executive team at our company is concerned about our ability to scale applications during peak periods of demand. Id like you to demonstrate the features of Amazon Web Services (AWS) that can be used to easily enable elasticity for applications.

Your task is to demonstrate how an Amazon EC2 Auto Scaling group with an Application Load Balancer (ALB) can automatically scale based on the number of connections made through the ALB. The EC2 instances should display a web page that is generated based on content stored in an Amazon S3 bucket.

**How you'll work**

Your project has been broken into a set of tasks. To complete these tasks, use the provided workspace. You can launch your workspace by clicking below or using the button in the top right of the screen.

Launch workspace

Task

1-Create S3 bucket and upload artifacts

2-Create Target Group and Application Load balancer

3-create Launch Template and Auto scaling group

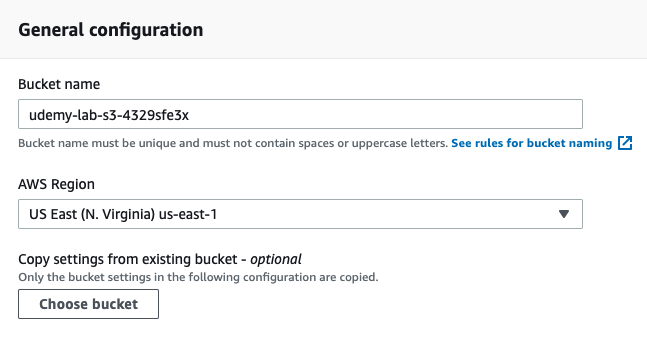
4-Test with multiple connections from browser

5-Clean up

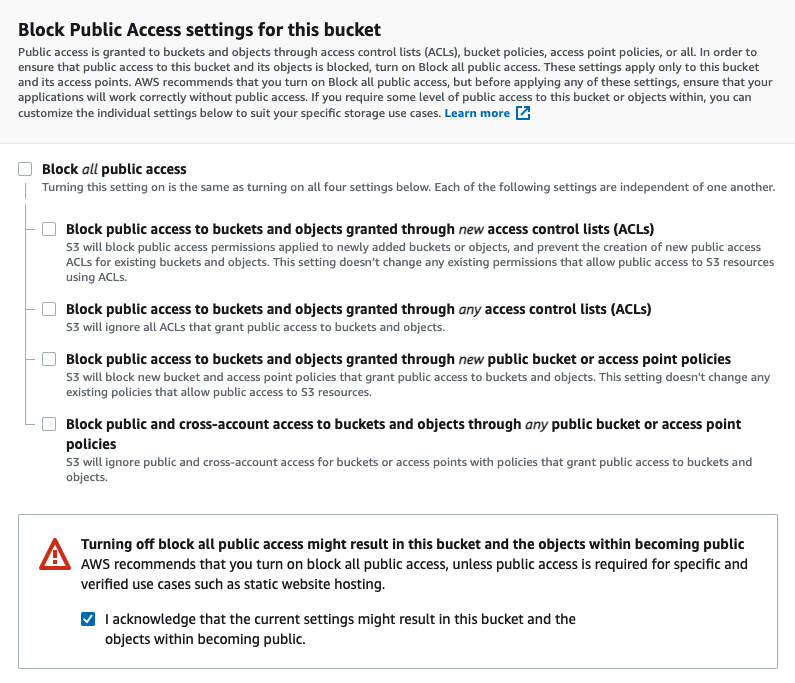
Create S3 bucket and upload artifacts

You will now create an Amazon S3 bucket in which to store the file that will be used to generate the HTML web page. The bucket will need to be configured for public access. Once the bucket has been created upload the file and ensure it is publicly accessible.

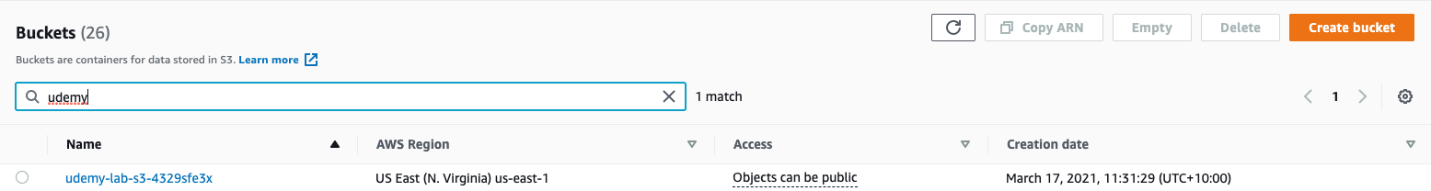
1. In the **AWS Management Console**, go to **Services** and click **S3**
2. Click **Create Bucket** and then set the bucket name to "udemy-lab-s3-" and add a random string of letters and numbers on the end to ensure uniqueness. E.g. "udemy-lab-s3-4329sfe3x9"



1. Change the **AWS Region** to "US-East (N. Virginia) us-east-1"
2. Deselect the option to "Block all public access" and select the checkbox to acknowledge the change



1. Click **Create Bucket**
2. You should now see the new bucket in the S3 Management Console



1. Click the name of the bucket and then select the **Permissions** tab. Under **Bucket Policy** click **Edit** and copy and paste the code below into the policy editor

{

*"Version"*: "2012-10-17",

*"Statement"*: [

        {

*"Effect"*: "Allow",

*"Principal"*: "\*",

*"Action"*: "s3:GetObject",

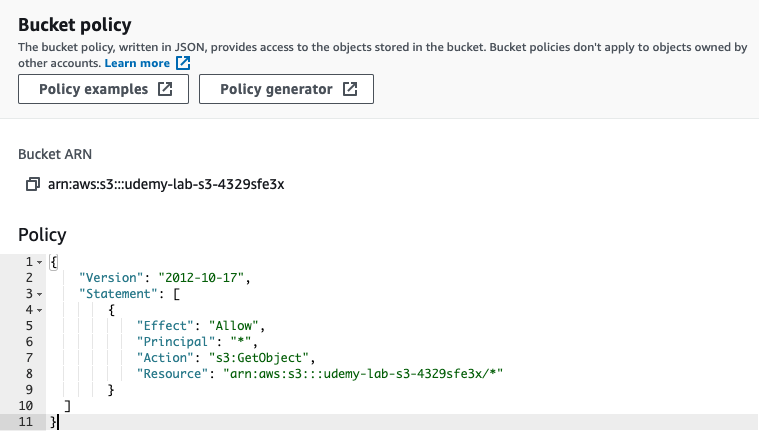
*"Resource"*: "arn:aws:s3:::YOURBUCKETNAMEHERE/\*"

        }

    ]

}

1. Copy the **Bucket ARN** and paste it over the ARN next to **Resource** (make sure you keep the /\* at the end). It should now look like this:



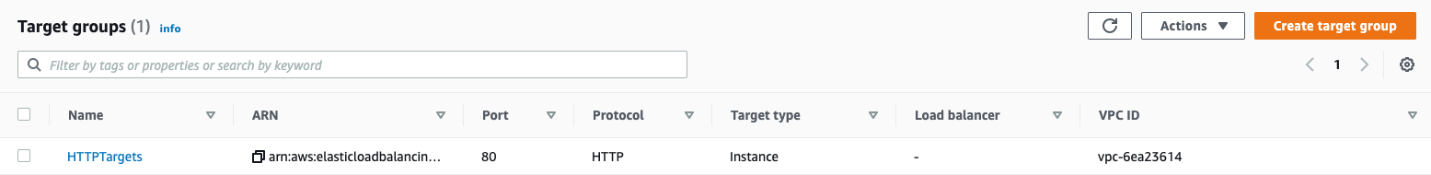
1. Click **Save changes**
2. On the **Objects** tab, click **Upload**, select the index.txt file from the resources, and click **Upload**

Ref s3-bucket-policy.json . index.txt

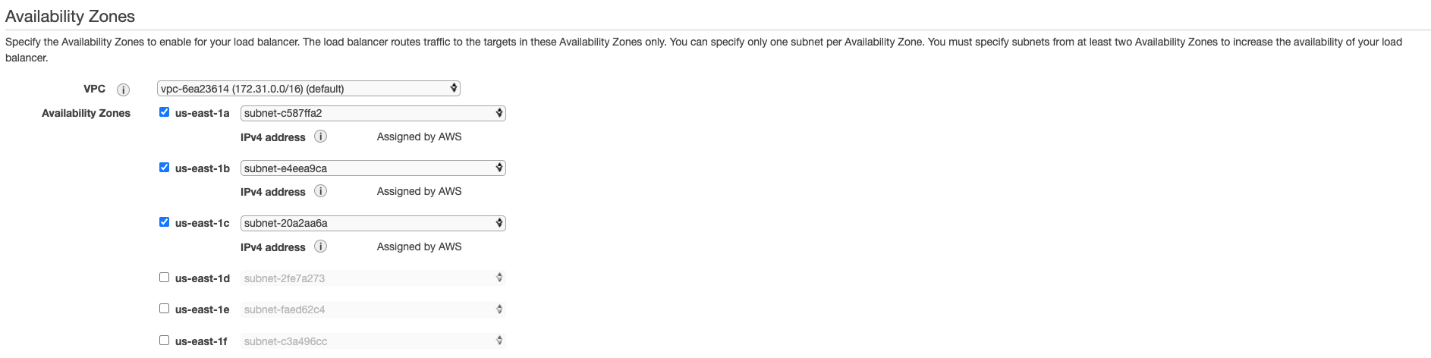
Create Target Group and Application Load balancer

You will now create a Target Group and Application Load Balancer. The ALB will be configured to distribute connections to the Target Group which will contain the instances launched by the Auto Scaling group.

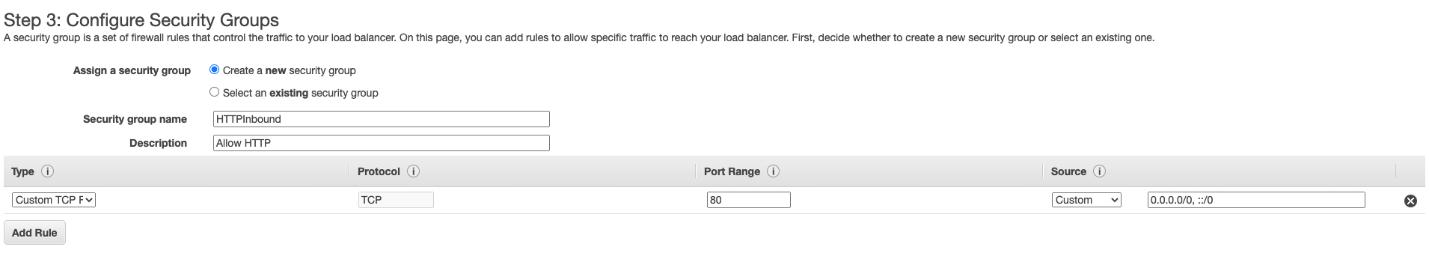
1. In the **AWS Management Console**, go to **Services** and click **EC2**
2. In the left-hand menu, choose **Target Groups** and then **Create target group**
3. On the **Basic configuration** page, leave the default values selected, enter the **Target group name** as "HTTPTargets" and click **Next**, then click **Create target group**
4. Your target group should look like this:



1. In the left-hand menu, choose **Load Balancers** and then **Create Load Balancer**. Click **Create** under **Application Load Balancer**
2. On the **Basic Configuration** page, enter the **Name** as "HTTPLoadBalancing"
3. Under **Availability Zones** enable at least three Availability Zones and subnets as in the image below:



1. Click **Next: Configure Security Settings**, and then **Next: Configure Security Groups**
2. Choose the option to **Create a new security group**, enter the **Security group name** as "HTTPInbound" and the **Description** as "Allow HTTP". The rule should appear as in the image below:

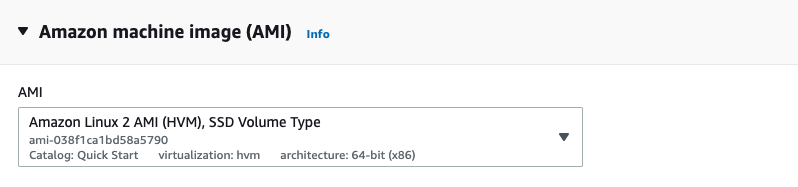


1. Click **Next: Configure Routing** and under **Target group** select **Existing target group** and ensure the "HTTPTargets" target group is selected
2. Click **Next: Register Targets**, and then **Next: Review**, and finally **Create** to create the load balancer

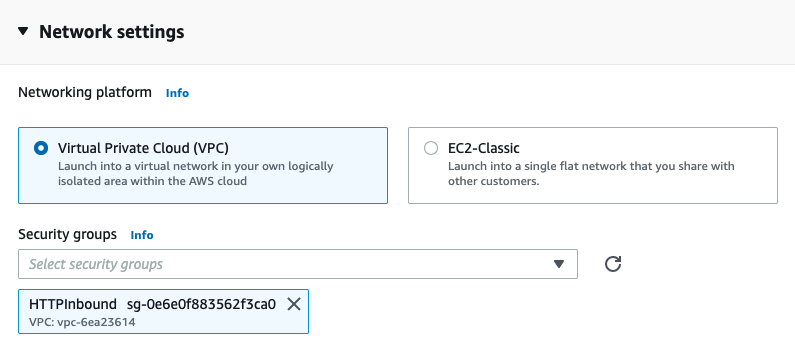
create Launch Template and Auto scaling group

You will now create a launch template with user data that picks up the index.txt file from the Amazon S3 bucket and configures a web page on EC2 instances that describes which Availability Zone the EC2 instance is located in. You will also create an Amazon EC2 Auto Scaling group that uses the launch configuration.

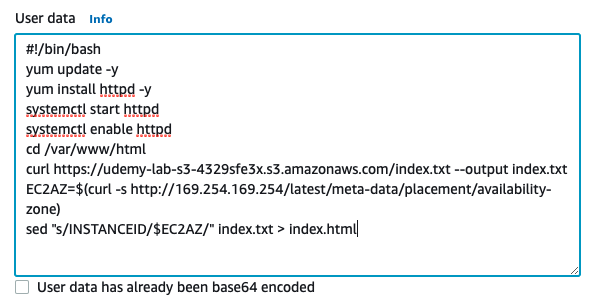
1. In the **AWS Management Console**, go to **Services** and click **EC2**
2. In the left-hand menu, choose **Launch Templates** and click **Create launch template**
3. For **Launch template name** enter "HTTPWebServer". Under **Amazon machine image** select the **Amazon Linux 2 AMI (HVM)** using the **64-bit (x86)** architecture



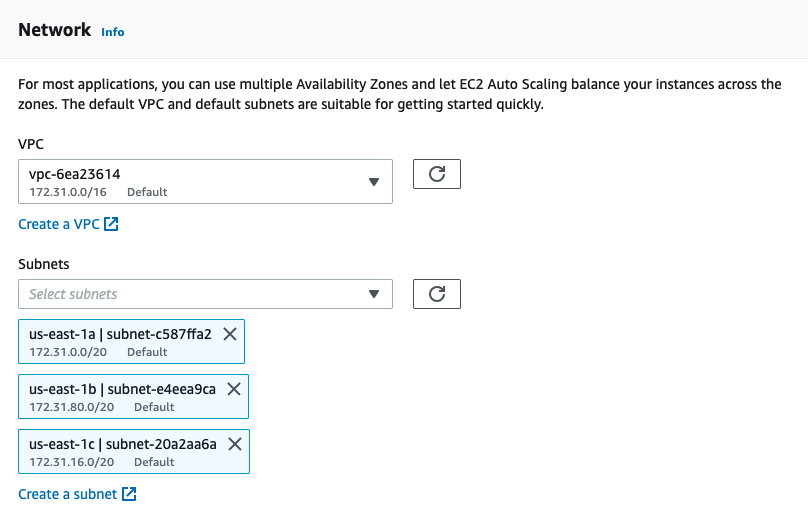
1. Under **Instance type** select **t2.micro**
2. Under **Network Settings** select the **HTTPInbound** security group



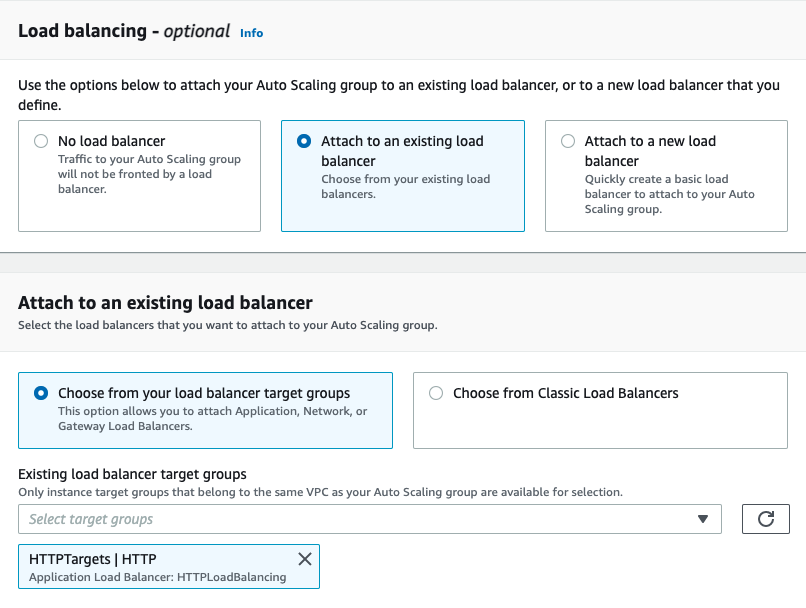
1. Under **Advanced details** copy and paste the code in the **user-data-httpd.sh** file into the **User data** area
2. You must then edit the bucket name on line 7 by entering the name of the bucket created earlier. It should now look like this (but with YOUR bucket name):



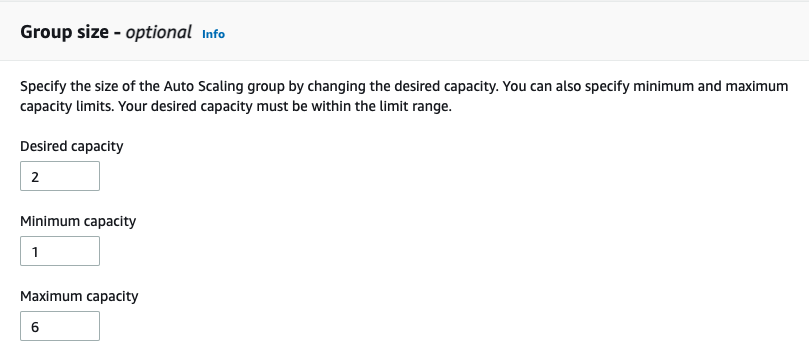
1. Click **Create launch template**
2. Back in the EC2 Management Console, in the left-hand menu, select **Auto Scaling Groups** and then **Create Auto Scaling group**
3. For **Auto Scaling group name** enter "MyWebServers" and under **Launch template** select the **HTTPWebServer** launch template and then click **Next**
4. Under **Network** add the same subnets (at least three) that you selected for the load balancer and then click **Next**



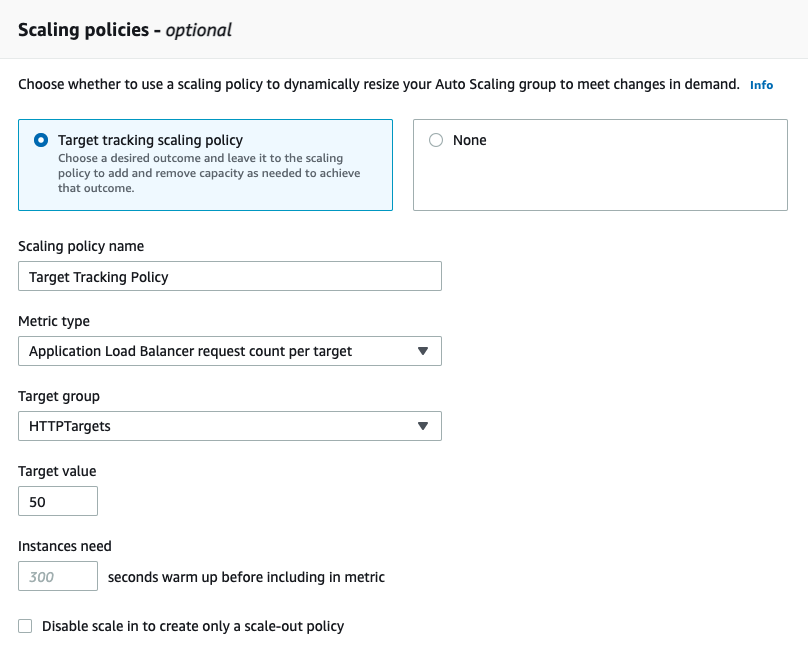
1. Under **Load balancing** select **Attach to an existing load balancer** and the select the **HTTPTargets** target group and then click **Next**



1. Set the **Group size** to the following values:
   1. Desired capacity = 2
   2. Minimum capacity = 1
   3. Maximum capacity = 6



1. Under **Scaling policies** select **Target tracking scaling policy** and for **Metric type** select **Application Load Balancer request count per target**. Select the **HTTPTargets** target group and enter the **Target value** as "5"



1. Then click **Next** three times and then finally **Create Auto Scaling group**

Ref user-data-httpd.sh

Test with multiple connections from browser

You will now test causing a scale-out event by using your web browser to create multiple connections to the Application Load Balancer.

1. In the **AWS Management Console**, go to **Services** and click **EC2**
2. In the left-hand menu, choose **Load Balancers** and under the Basic Configuration for the **HTTPLoadBalancing** load balancer copy the **DNS name** to your clipboard
3. In a browser window paste the DNS name of the load balancer and refresh a few times to confirm you are cycling between instances in different Availability Zones
4. You should currently have two instances running. To cause a scaling event keep refreshing your browser connection multiple times per minute over a three-minute period
5. Within a few minutes you should find that a scale-out event occurs and multiple EC2 instances are automatically launched and viewable in the EC2 Dashboard under **Instances**
6. Next, close the browser / tab that you used to connect to the Elastic Load Balancer. After around 15 minutes a scale-in event should occur and automatically terminate EC2 instances that are no longer required

Clean up

Now that you have created an auto scaling website, lets clean up the lab environment.

1. Go to the **Auto Scaling Groups** screen and select the MyWebServers group. Click on the **Delete** button.
2. Then go the **Load Balancer** screen, select the HTTPLoadBalancing load balancer, click on the **Actions** menu and select **Delete**. Confirm the delete when prompted.
3. Next go the **Target Groups** screen, select the HTTPTargets target group, click on the **Actions** menu and select **Delete**. Confirm the delete when prompted.
4. After that, go the **Launch Templates** screen, select the HTTPWebServer launch template, click on the **Actions** menu and select **Delete template**. Confirm the delete when prompted.
5. Lastly, go to the **S3** screen, select the udemy-lab-s3- bucket you created earlier, and click on the **Delete** button. Confirm the delete when prompted.