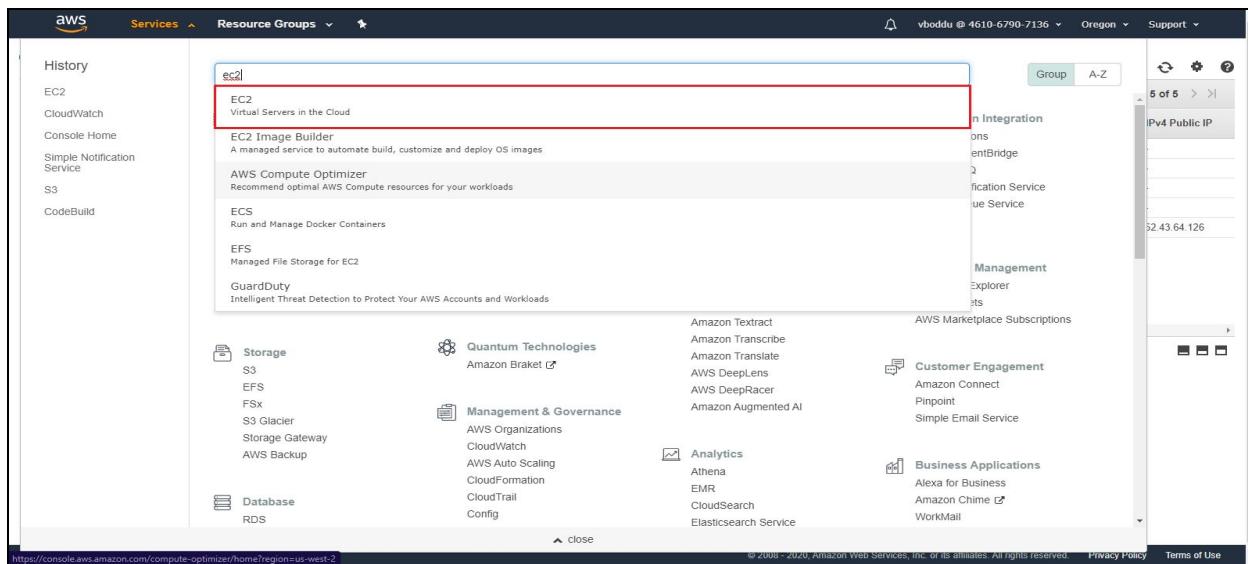


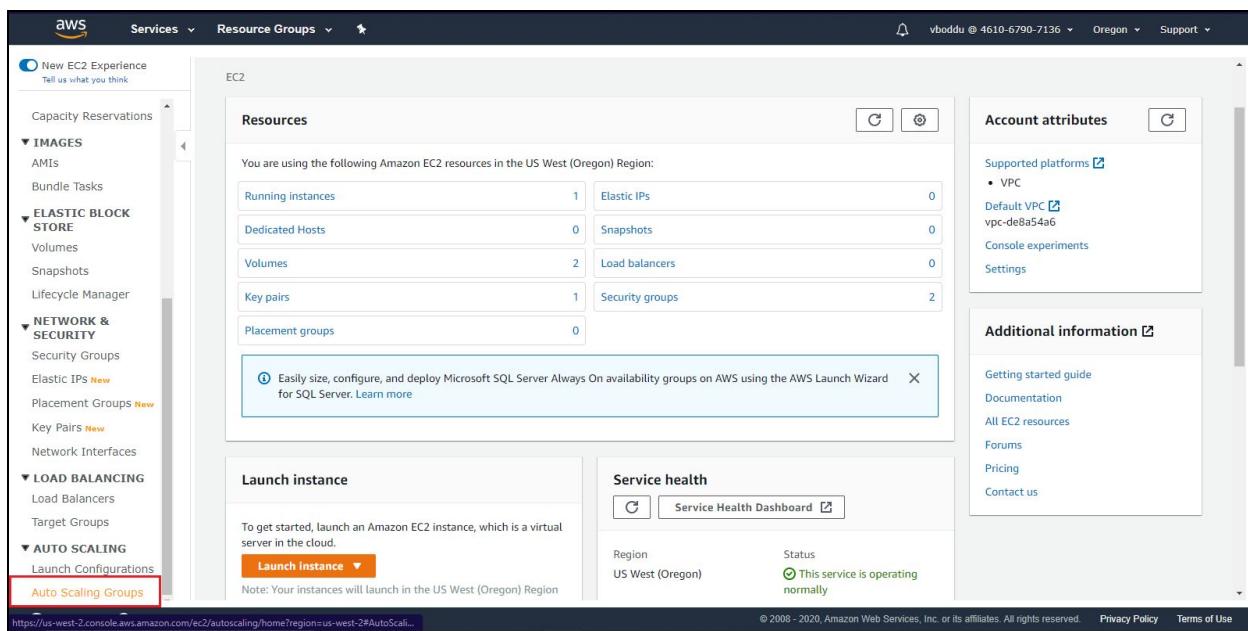
Autoscaling and load balancing

By the end of the document, we will be able to configure Auto scaling and load balancing for the EC2 instance.

First, we need to login to the AWS console and search for EC2 service.



Following is the dashboard for EC2 service and click on Auto Scaling Groups as shown in the dashboard.



Now click on Create auto-scaling group.

The screenshot shows the AWS EC2 Auto Scaling console. On the left, there's a navigation sidebar with various links like EC2 Dashboard, Instances, Launch Templates, and Security Groups. The main content area has a header 'Welcome to Auto Scaling' and a note about the new design. It features three sections: 'Benefits of Auto Scaling' with icons for 'Automated Provisioning', 'Adjustable Capacity', and 'Launch Template Support'. Below these are three 'Learn more' links. To the right, there's an 'Additional Information' section with links to Getting Started Guide, Documentation, All EC2 Resources, Forums, Pricing, and Contact Us. At the bottom, there are 'Feedback', language selection ('English (US)'), and standard AWS footer links.

Click on create new launch configuration to configure the Auto Scaling EC2 instance.

This screenshot shows the 'Create Auto Scaling Group' wizard. It has two main options: 'Launch Configuration' (selected) and 'Launch Template' (New). Under 'Launch Configuration', there's a note about using existing configurations if they support EC2 features. A red box highlights the 'Create a new launch configuration' button. The right side of the screen shows the 'Launch Template' section with a note about its benefits and a 'Create new launch template' button. The bottom of the screen includes standard AWS footer links.

The following are the operating systems that are provided by AWS and Here we need to select one of them as shown below.

1. Choose AMI 2. Choose Instance Type 3. Configure details 4. Add Storage 5. Configure Security Group 6. Review

Create Launch Configuration

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-01460aa81365561fe 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	Select 64-bit
Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-087c2c50437d0b80d Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type Free tier eligible	Select 64-bit
SUSE Linux Enterprise Server 15 SP1 (HVM), SSD Volume Type - ami-0e6612c7b620ecf3a SUSE Linux Enterprise Server 15 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled. Free tier eligible	Select 64-bit
Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-06d51e91cea0dac8d Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services). Free tier eligible	Select 64-bit
Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-0994c095691a46fb5 Ubuntu Server 16.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services). Free tier eligible	Select 64-bit
Microsoft Windows Server 2019 Base - ami-0a1d405ce719bebfd Microsoft Windows 2019 Datacenter edition: [English]	Select 64-bit

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Then Select General purpose t2.micro tier instance for our requirement. It provides 1 vCPU and 1 GB memory for the instance and the performance of the instance is low to moderate. Now click on next to configure the instance.

1. Choose AMI 2. Choose Instance Type 3. Configure details 4. Add Storage 5. Configure Security Group 6. Review

Create Launch Configuration

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

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	t2.xlarge	4	16	EBS only	-	Moderate
General purpose	t2.2xlarge	8	32	EBS only	-	Moderate
General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit
General purpose	t3a.micro	2	1	EBS only	Yes	Up to 5 Gigabit

Cancel Previous Next: Configure details © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Once click on configure details we need to provide inputs like the name of the auto scaling, User data and make remaining as default. Click on add storage.

Services ▾ Resource Groups ▾

1. Choose AMI 2. Choose Instance Type 3. Configure details 4. Add Storage 5. Configure Security Group 6. Review

Create Launch Configuration

Name (Required)

Purchasing option Request Spot Instances

IAM role

Monitoring Enable CloudWatch detailed monitoring [Learn more](#)

▶ Advanced Details

Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.

Cancel Previous Skip to review Next: Add Storage

Services ▾ Resource Groups ▾

1. Choose AMI 2. Choose Instance Type 3. Configure details 4. Add Storage 5. Configure Security Group 6. Review

Create Launch Configuration

Name (Required)

Purchasing option Request Spot Instances

IAM role

Monitoring Enable CloudWatch detailed monitoring [Learn more](#)

▼ Advanced Details

Kernel ID

RAM Disk ID

User data As text As file Input is already base64 encoded

```
sudo git clone https://github.com/vboddu001/devopsBackend.git
cd devopsBackend
sudo npm install
sudo npm install express
sudo npm i -g pm2
pm2 start index.js
```

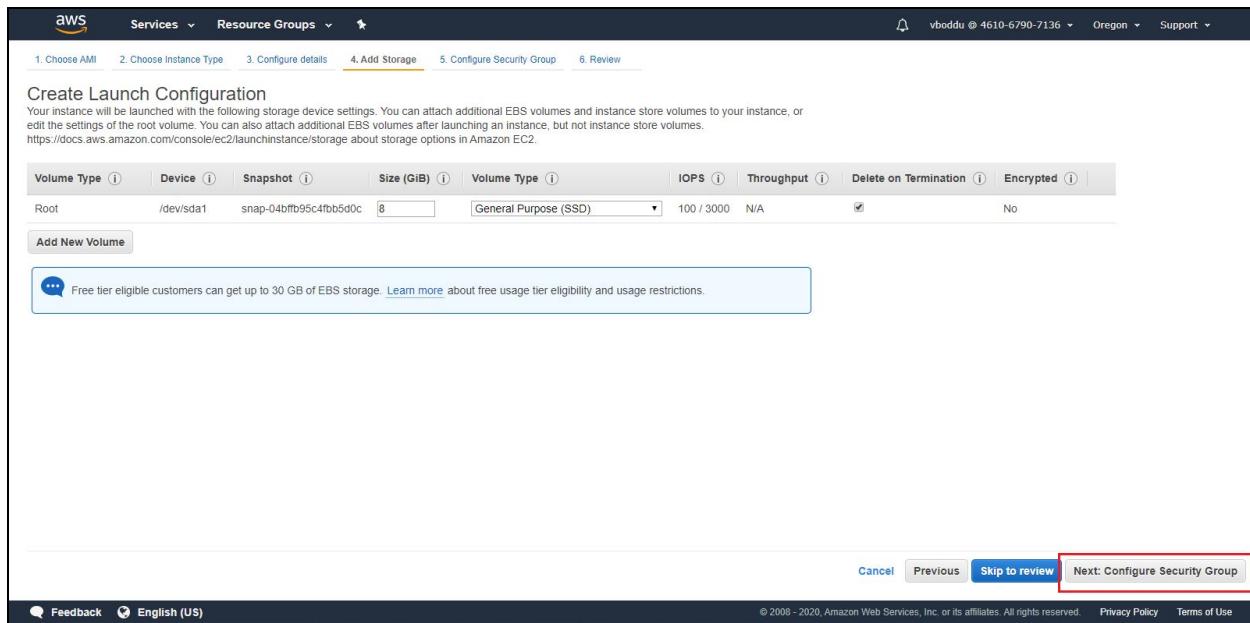
IP Address Type Only assign a public IP address to instances launched in the default VPC and subnet. (default) Assign a public IP address to every instance. Do not assign a public IP address to any instances. Note: this option only affects instances launched into an Amazon VPC

Cancel Previous Skip to review Next: Add Storage

These are the commands that are used in the User data.

```
#!/bin/bash
sudo apt-get update
curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -
sudo apt-get install nodejs -y
cd /var/www/
sudo git clone https://github.com/vboddu1/devopsBackend.git
cd devopsBackend
sudo npm install
sudo npm install express
sudo npm i -g pm2
pm2 start index.js
```

By default, AWS provides 8 GB storage for the t2.micro instance. Now click on next to configure security groups.



Create a new security group to instance. A security group is a set of firewall rules that control the traffic for our instance. Here we are adding all traffic ports. Now click on review and launch instance.

Step 6: Configure Security Group

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 about Amazon EC2 security groups.](#)

Assign a security group: Create a new security group Select an existing security group

Security group name: AutoScaling-Security-Group

Description: launch-wizard-1 created 2020-01-22T01:35:16.495+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
All traffic	All	0 - 65535	My IP 182.72.208.42/32	e.g. SSH for Admin Desktop

Add Rule

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous Review and Launch

Now review the instance configuration and click on Launch.

Create Launch Configuration

Review the details of your launch configuration. You can go back to edit the details of each section before you finish.

⚠ Improve security of instances launched using your launch configuration, Autoscaling. Your security group, AutoScaling-Security-Group, 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

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-06d51e91cea0dac8d

Instance Type

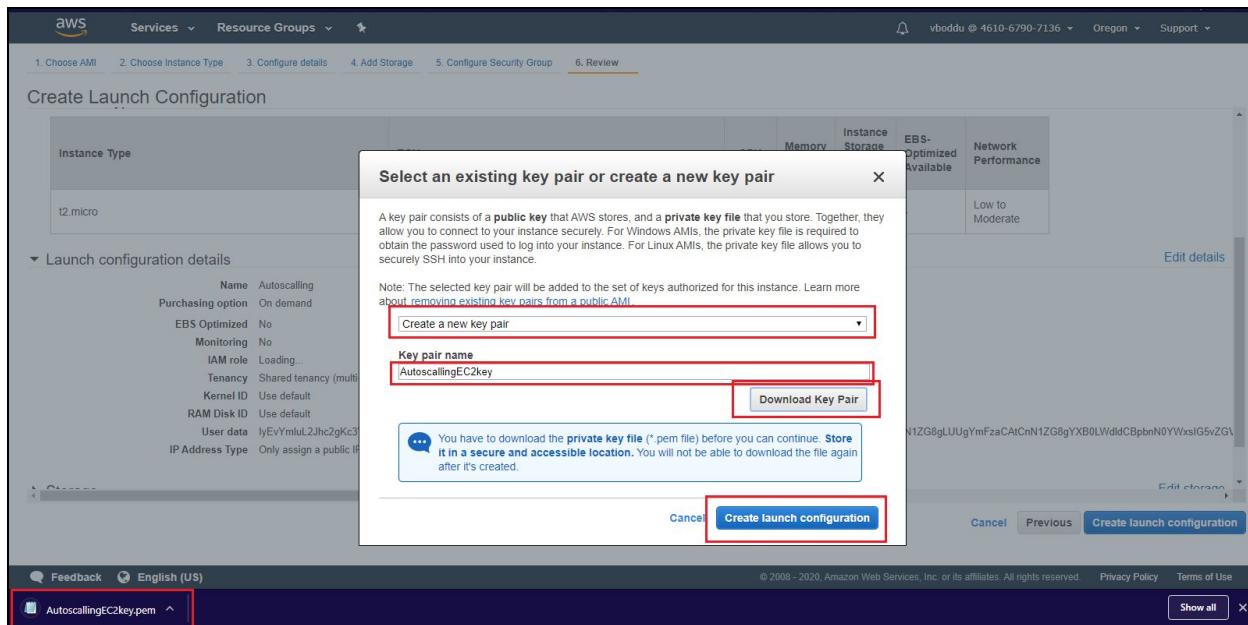
Instance Type	ECUs	vCPUs	Memory GiB	Instance Storage (GiB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Launch configuration details

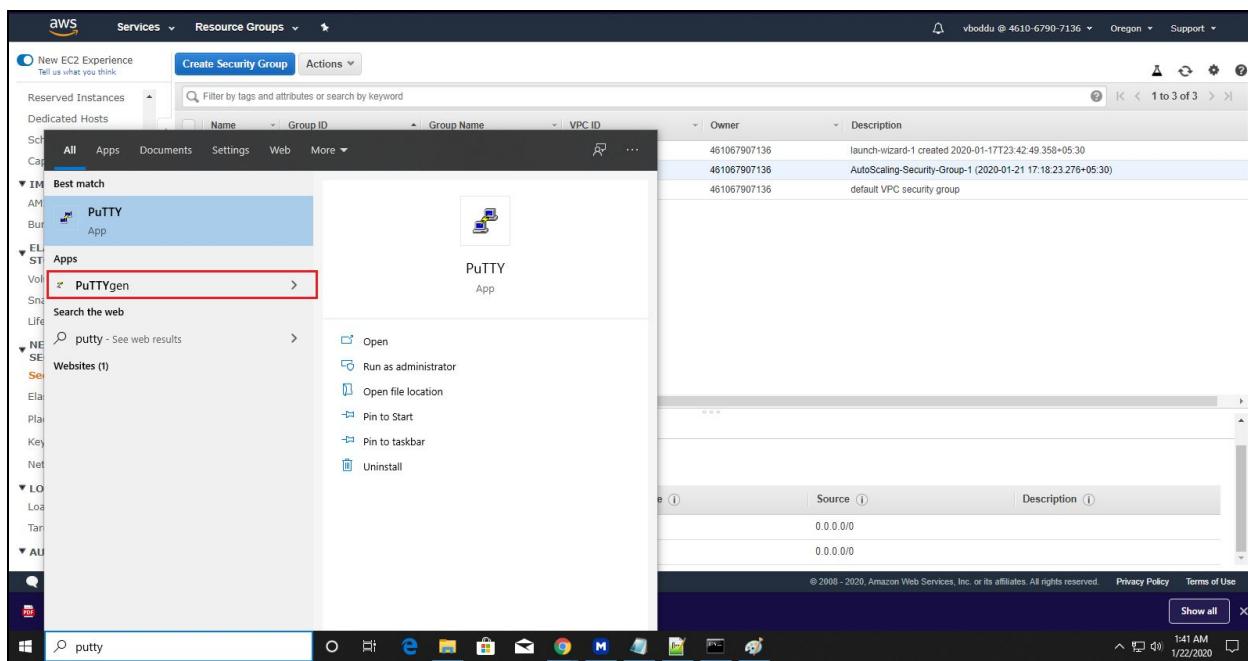
Name: Autoscaling
Purchasing option: On demand
FRS Optimized: No

Cancel Previous Create launch configuration

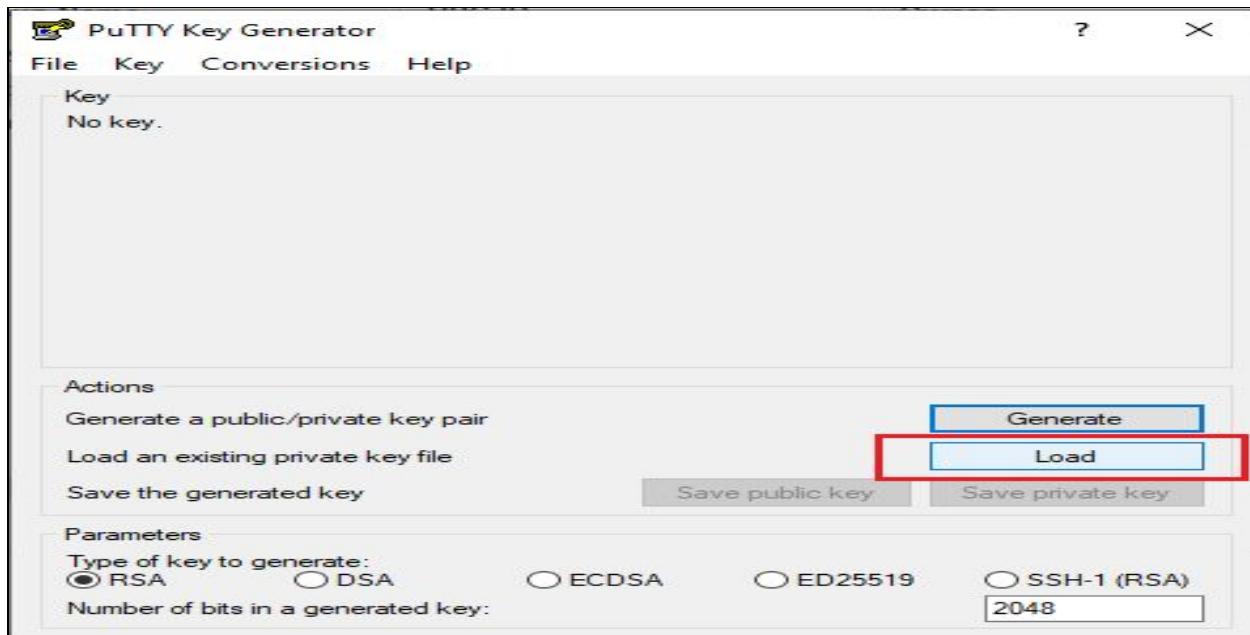
Next, we need to create a key. Using the key we can be able to access the instance. Choose to create a new key and click on the download key pair. A **pem** file will be downloaded.



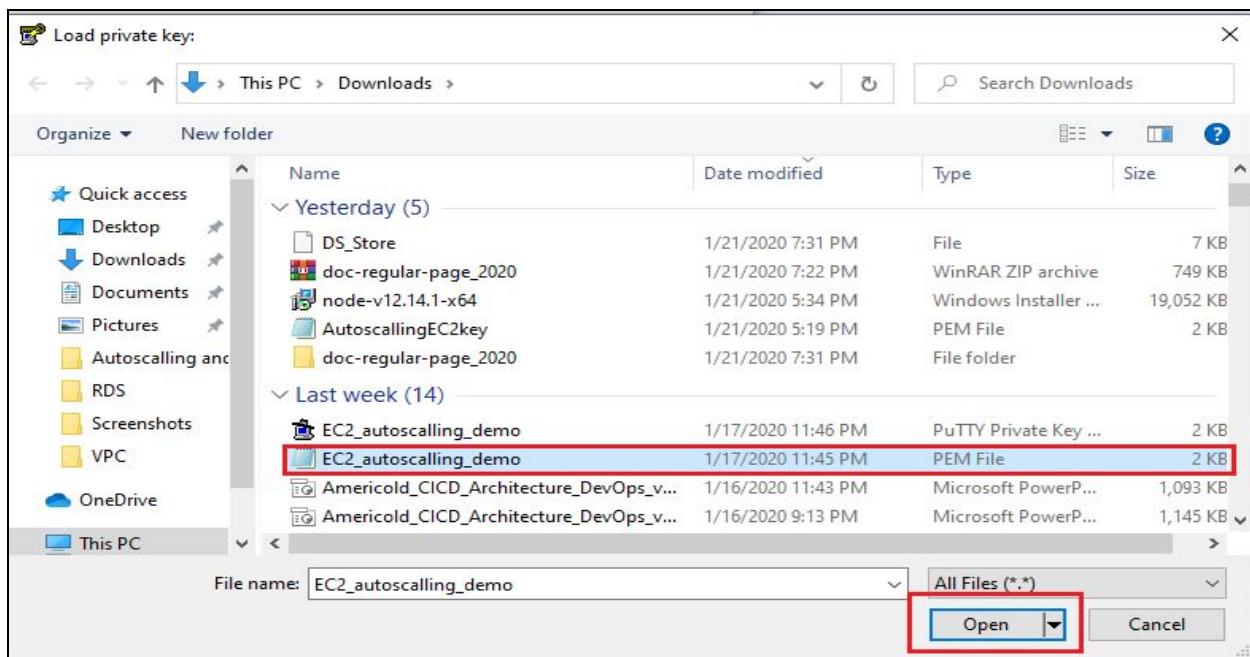
After downloading the key pair we need to generate a public key using the PuTTYgen as follows.



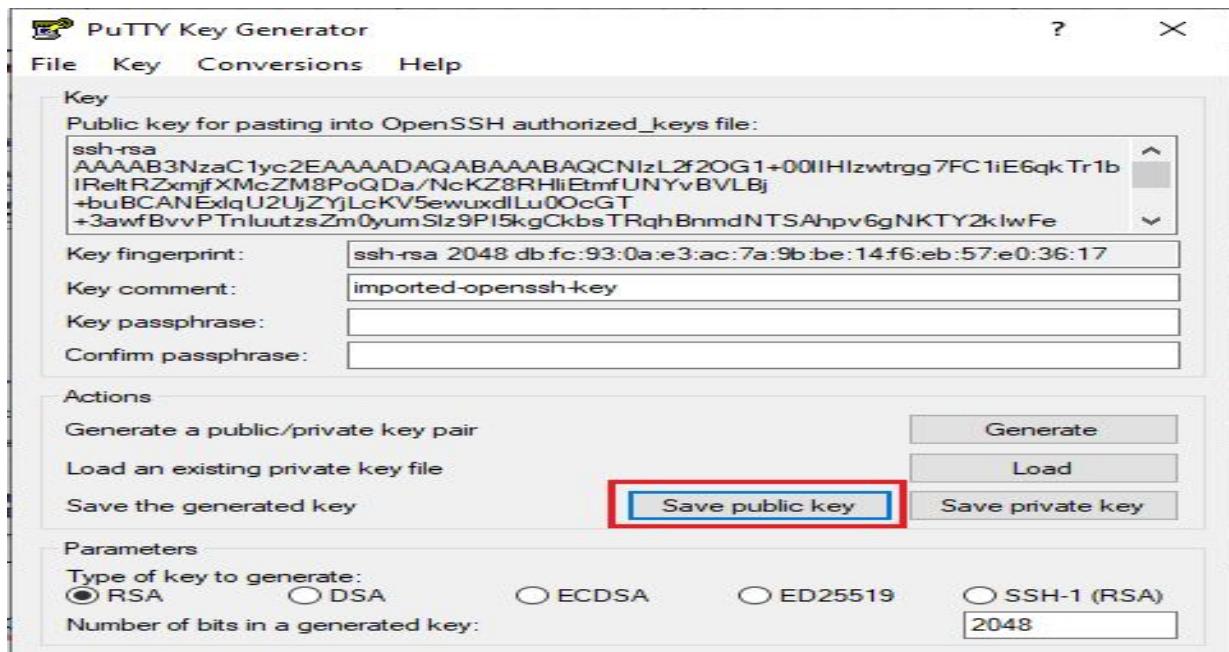
Now, Load the downloaded pem file into the PuTTY Key Generator. Click on load to select the key pair.



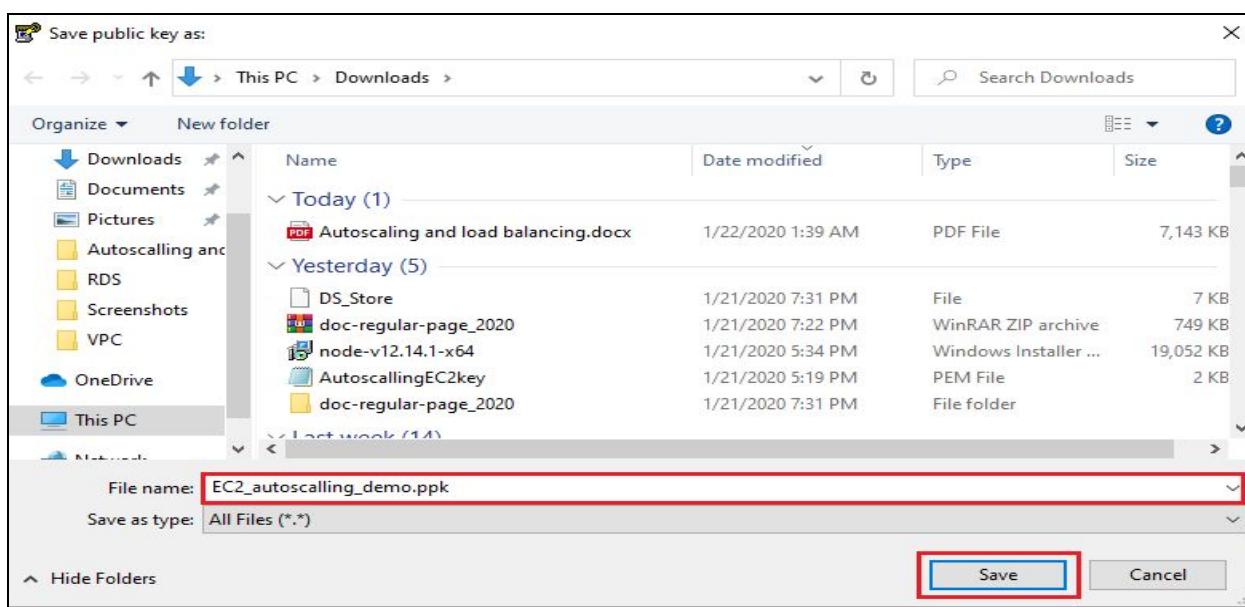
Select the key pair file and click on open



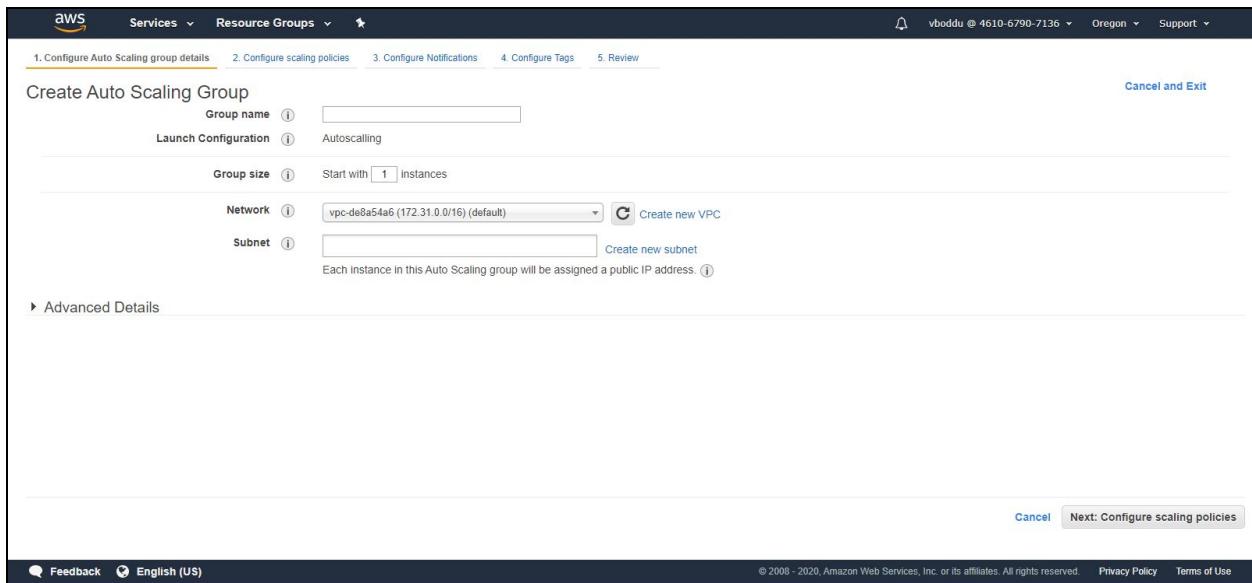
After loading the key pair into Putty key generator click on save public key.



Save the file name with ppk format and click on save.



Once we configure the Auto Scaling launch configuration then we need to create the Autoscaling group below is the dashboard for creation of Autoscaling group



1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

Create Auto Scaling Group

Group name:

Launch Configuration: Autoscaling

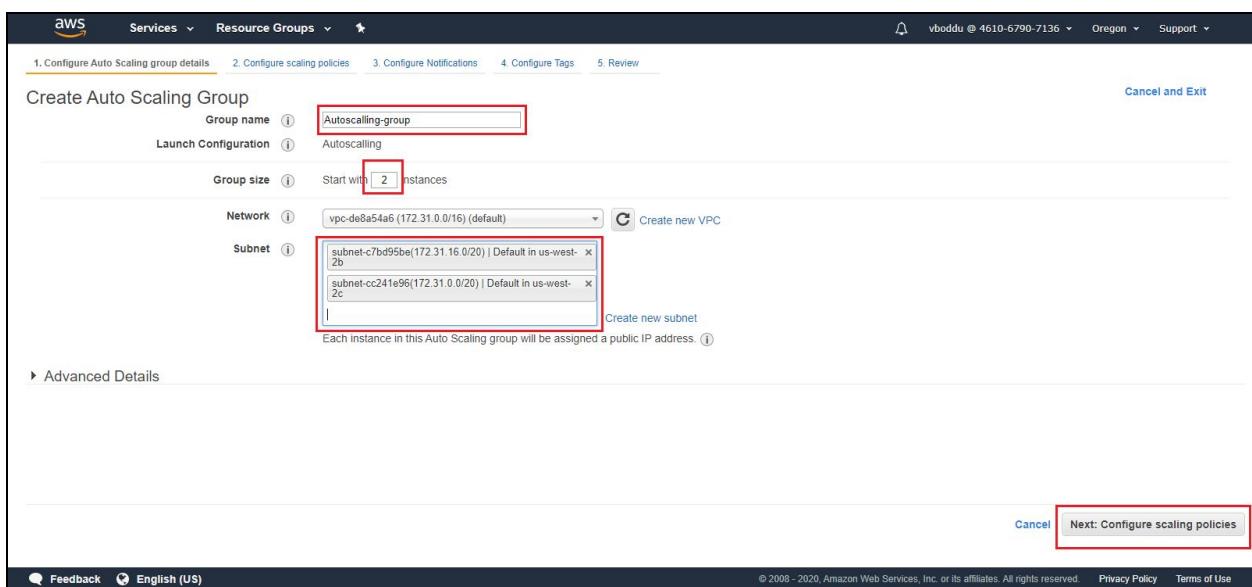
Group size: Start with instances

Network: vpc-de8a54a6 (172.31.0.0/16) (default)

Subnet:
Each instance in this Auto Scaling group will be assigned a public IP address.

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Here We need to provide various inputs like group name, group size and select at least two default subnets and click on create scaling policies.



1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

Create Auto Scaling Group

Group name:

Launch Configuration: Autoscaling

Group size: Start with instances

Network: vpc-de8a54a6 (172.31.0.0/16) (default)

Subnet:

Each instance in this Auto Scaling group will be assigned a public IP address.

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

In this section keep everything as default and click on Next.

The screenshot shows the AWS Auto Scaling 'Create Auto Scaling Group' wizard at step 2. The 'Keep this group at its initial size' radio button is selected. At the bottom right, the 'Next: Configure Notifications' button is highlighted with a red box.

Click on configure tags

The screenshot shows the AWS Auto Scaling 'Create Auto Scaling Group' wizard at step 3. The 'Add notification' button is visible. At the bottom right, the 'Next: Configure Tags' button is highlighted with a red box.

Tags are case-sensitive key-value pairs that define the group. Configure tags and click on the review as shown below.

Services ▾ Resource Groups ▾

1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

Create Auto Scaling Group

A tag consists of a case sensitive key-value pair that you can use to identify your group. For example, you could define a tag with Key = Environment and Value = Production. You can optionally choose to apply these tags to instances in the group when they launch. [Learn more](#).

Key	Value
Name	Autoscaling_demo

Add tag 49 remaining

Tag New Instances ⓘ

Cancel Previous Review

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Now review the instance configuration and click on create Autoscaling group.

Services ▾ Resource Groups ▾

1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

Create Auto Scaling Group

Please review your Auto Scaling group details. You can go back to edit changes for each section. Click **Create Auto Scaling group** to complete the creation of an Auto Scaling group.

Auto Scaling Group Details

Group name	Autoscalling-group	Edit details
Group size	2	
Minimum Group Size	2	
Maximum Group Size	2	
Subnet(s)	subnet-c7bd95be,subnet-cc241e96	
Health Check Grace Period	300	
Detailed Monitoring	No	
Instance Protection	None	
Service Linked Role	AWSServiceRoleForAutoScaling	

Scaling Policies

Notifications

Tags

Name	Autoscaling_demo	tag new instances	Edit tags
------	------------------	-------------------	-----------

Cancel Previous Create Auto Scaling group

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Auto Scaling group creation status

Successfully created Auto Scaling group

[View creation log](#)

View

- [View your Auto Scaling groups](#)
- [View your launch configurations](#)

Here are some helpful resources to get you started

[Close](#)

Try the new design for Amazon EC2 Auto Scaling

This older console is being replaced with the new EC2 Auto Scaling console. No new features or improvements will be made in this older console. Go to the new console.

Create Auto Scaling group Actions

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace Period
Autoscaling-gr...	Autoscaling	2	2	2	2	us-west-2b, us-west-2c	300	300

Auto Scaling Group: Autoscaling-group

Details Activity History Scaling Policies Instances Monitoring Notifications Tags Scheduled Actions Lifecycle Hooks

Launch Configuration Autoscaling Availability Zone(s) us-west-2b, us-west-2c

Desired Capacity 2 Subnet(s) subnet-c7bd95be.subnet-cc241e96

Min 2 Classic Load Balancers

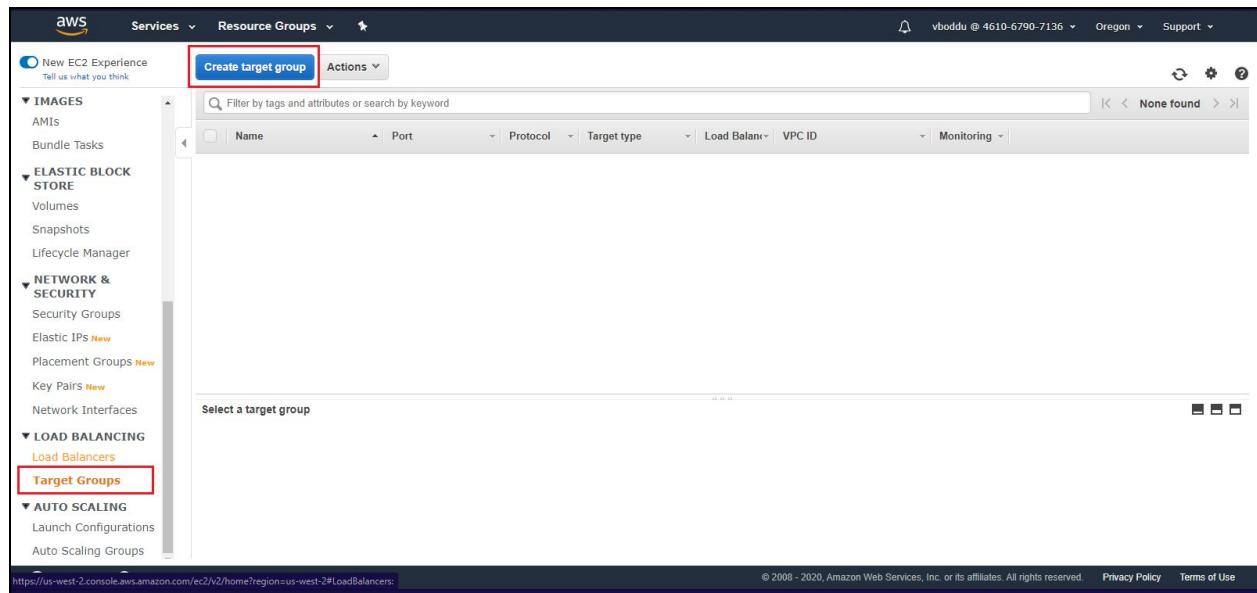
Max 2 Target Group

Health Check Type EC2

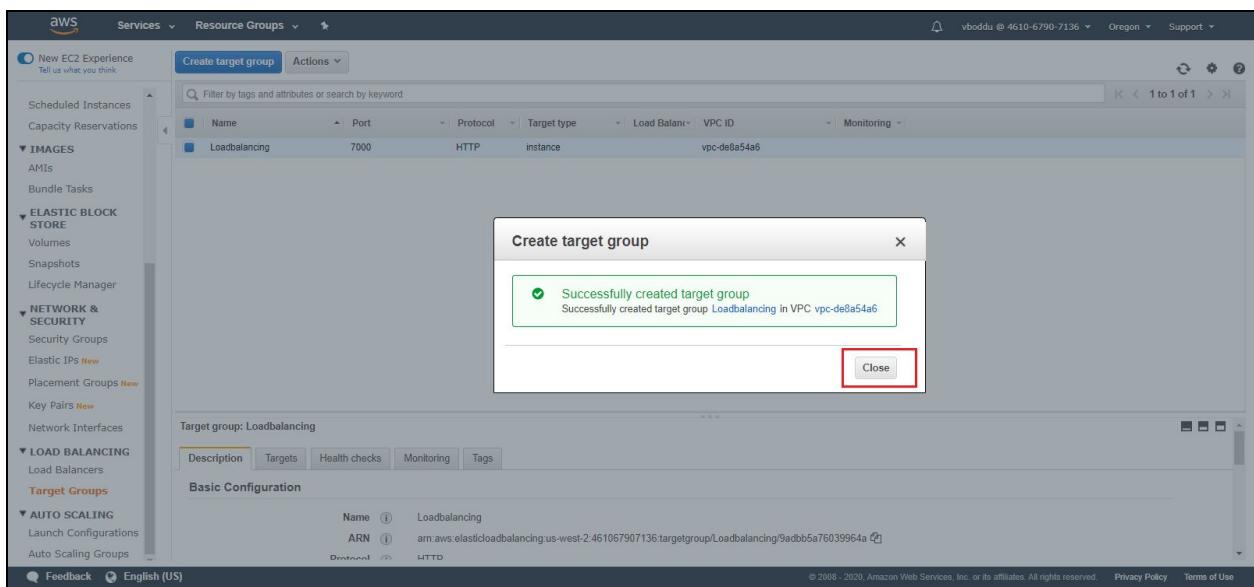
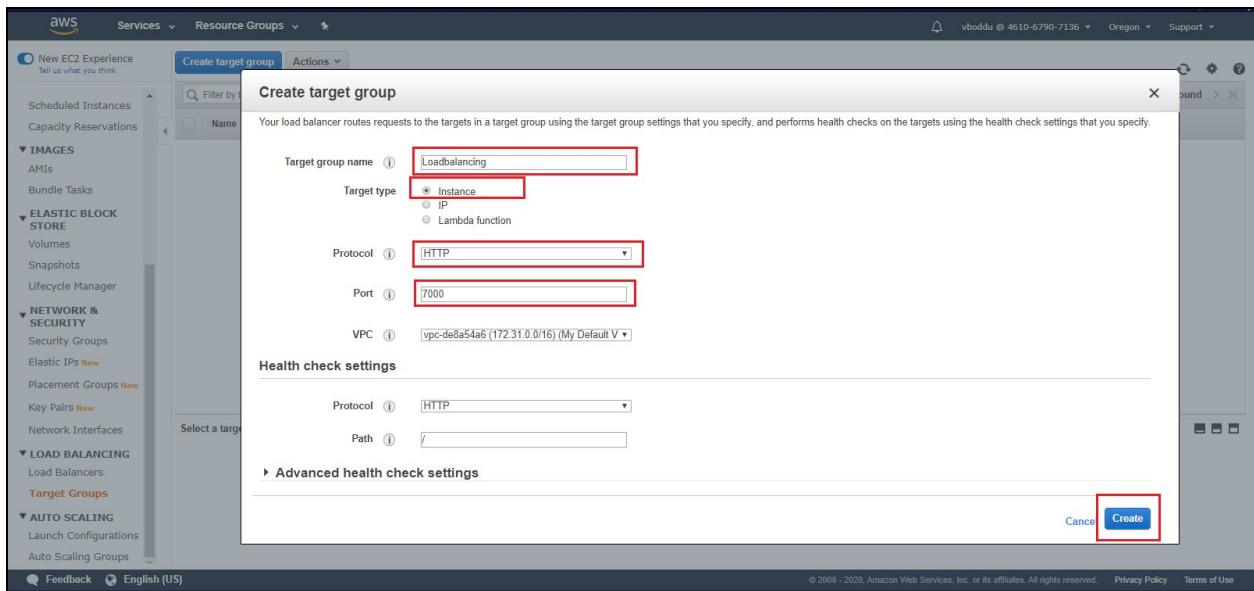
Health Check Grace Period 300

Instance Protection

For creation of load balancer first we need to create target groups.



Once we select the create target group, it will list the option where we need to provide various inputs like name of the target group, target type, protocol, port, make remaining as default and create.



Next we need to create the load balancer, select load balancers and create load balancer.

The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, Services dropdown, Resource Groups dropdown, and user information (vboddu @ 4610-6790-7136, Oregon, Support). The main menu on the left has sections like New EC2 Experience, Scheduled Instances, Capacity Reservations, IMAGES, AMIs, Bundle Tasks, ELASTIC BLOCK STORE, Volumes, Snapshots, Lifecycle Manager, NETWORK & SECURITY, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, LOAD BALANCING (with Load Balancers selected), Target Groups, AUTO SCALING, Launch Configurations, and Auto Scaling Groups. A red box highlights the 'Create Load Balancer' button at the top of the main content area. The content area displays a message: 'You do not have any load balancers in this region.' Below this is a 'Select a load balancer' section with three tabs: Application Load Balancer, Network Load Balancer, and Classic Load Balancer. The Application Load Balancer tab is active, showing icons for HTTP and HTTPS and a 'Create' button.

Once we select the create load balancer, it will list the types of load balancer select the application load balancer.

This screenshot shows the 'Select load balancer type' page. The top navigation bar is identical to the previous one. The main content area displays three options: Application Load Balancer, Network Load Balancer, and Classic Load Balancer. The Application Load Balancer section is highlighted with a red box around its title and 'Create' button. It contains a circular icon with 'HTTP HTTPS' and a 'Create' button. Below it is a brief description: 'Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.' There is also a 'Learn more >' link. The Network Load Balancer and Classic Load Balancer sections are also shown but are not highlighted.

when we select the application load balancer, it will provide various inputs like name of the load balancer, listeners(connection requests) and choose availability zones(subnets). Click on next to configure security settings.

This screenshot shows the 'Step 1: Configure Load Balancer' page in the AWS Management Console. The 'Basic Configuration' section includes fields for Name (LoadBalancer), Scheme (Internet-facing), and IP address type (IPv4). The 'Listeners' section shows a single listener for HTTP port 80. The 'Availability Zones' section lists one VPC subnet. Navigation links at the top include '1. Configure Load Balancer', '2. Configure Security Settings', '3. Configure Security Groups', '4. Configure Routing', '5. Register Targets', and '6. Review'. A 'Next: Configure Security Settings' button is visible at the bottom right.

This screenshot shows the same 'Step 1: Configure Load Balancer' page, but with specific fields highlighted with red boxes: the 'Name' field containing 'LoadBalancer', the 'Load Balancer Protocol' dropdown set to 'HTTP', and the 'Load Balancer Port' input field set to '7000'. The rest of the interface is identical to the first screenshot, including the navigation links and the 'Next: Configure Security Settings' button.

Step 1: Configure Load Balancer

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol: HTTP | Load Balancer Port: 7000

Add listener

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC	vpc-de8a54a6 (172.31.0.0/16) (default)
Availability Zones	<input checked="" type="checkbox"/> us-west-2a subnet-52441119 IPv4 address Assigned by AWS <input checked="" type="checkbox"/> us-west-2b subnet-c7bd95be IPv4 address Assigned by AWS <input type="checkbox"/> us-west-2c subnet-cc241e96 <input type="checkbox"/> us-west-2d subnet-41fed569

Cancel | Next: Configure Security Settings

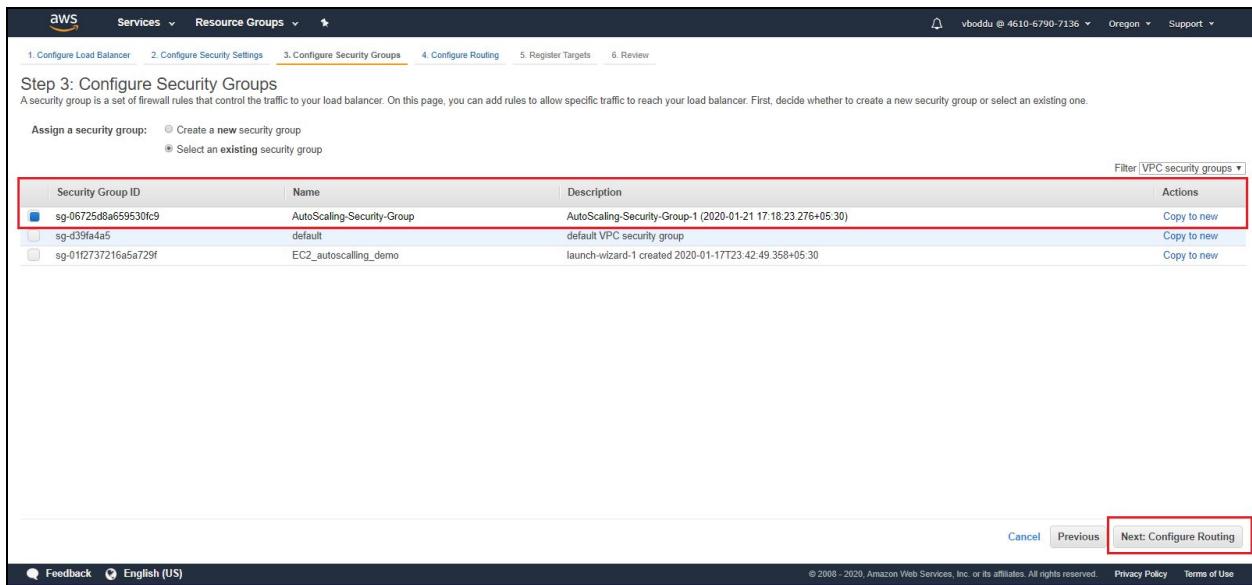
Click on next to configure security groups.

Step 2: Configure Security Settings

⚠ Improve your load balancer's security. Your load balancer is not using any secure listener.
If your traffic to the load balancer needs to be secure, use the HTTPS protocol for your front-end connection. You can go back to the first step to add/configure secure listeners under Basic Configuration section. You can also continue with current settings.

Cancel | Previous | Next: Configure Security Groups

Here choose the security groups and click on configure routing.



Step 3: Configure Security Groups

A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group:

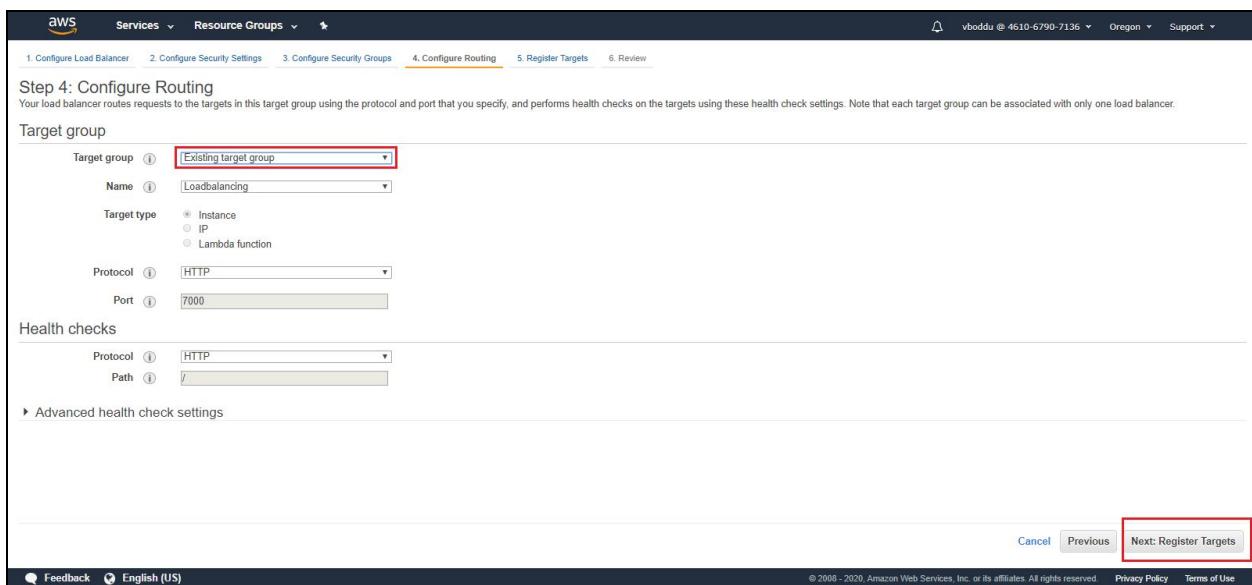
- Create a new security group
- Select an existing security group

Security Group ID	Name	Description	Actions
<input checked="" type="checkbox"/> sg-06725d8a659530c9	AutoScaling-Security-Group	AutoScaling-Security-Group-1 (2020-01-21 17:18:23.276+05:30)	Copy to new
<input type="checkbox"/> sg-d39fa4a5	default	default VPC security group	Copy to new
<input type="checkbox"/> sg-01f2737216a5a729f	EC2_autoscalling_demo	launch-wizard-1 created 2020-01-17T23:42:49.358+05:30	Copy to new

Filter: VPC security groups ▾

Cancel Previous Next: Configure Routing

In the configure routing section, select the existing target group that you created before, and click on register targets



Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and performs health checks on the targets using these health check settings. Note that each target group can be associated with only one load balancer.

Target group

Target group	<input type="text" value="Existing target group"/>
Name	<input type="text" value="Loadbalancing"/>
Target type	<input checked="" type="radio"/> Instance <input type="radio"/> IP <input type="radio"/> Lambda function
Protocol	<input type="text" value="HTTP"/>
Port	<input type="text" value="7000"/>

Health checks

Protocol	<input type="text" value="HTTP"/>
Path	<input type="text" value="/"/>

Advanced health check settings

Cancel Previous Next: Register Targets

Here click on next

Now review the configuration and click on Create.

AWS Services Resource Groups ★

Load Balancer Creation Status

✓ Successfully created load balancer
Load balancer LoadBalancer was successfully created.
Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic, and for the targets to complete the registration process and pass the initial health checks.

Suggested next steps

- Discover other services that you can integrate with your load balancer. Visit the [Integrated services](#) tab within [LoadBalancer](#)
- Consider using AWS Global Accelerator to further improve the availability and performance of your applications. [AWS Global Accelerator console](#)

Close

Feedback English (US) © 2006 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

New EC2 Experience Tell us what you think.

Services Resource Groups ★

Create Load Balancer Actions ▾

Filter by tags and attributes or search by keyword

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
LoadBalancer	LoadBalancer-2103386796...	provisioning	vpc-de8a54a6	us-west-2a, us-west-2b	application	January 21, 2020 at 5:26:22 ...	Edit

Load balancer: LoadBalancer

Description Listeners Monitoring Integrated services Tags

Basic Configuration

Name	LoadBalancer
ARN	arn:aws:elasticloadbalancing:us-west-2:461067907136:loadbalancer/app/LoadBalancer/14439b2430d3442e
DNS name	LoadBalancer-2103386796.us-west-2.elb.amazonaws.com
State	provisioning
Type	application
Schema	internet-facing
IP address type	ipv4

Edit IP address type

Feedback English (US) © 2006 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Click on target groups to register the instances as shown below.

The screenshot shows the AWS EC2 Target Groups interface. On the left, there's a navigation sidebar with various services like Scheduled Instances, Capacity Reservations, IMAGES, ELASTIC BLOCK STORE, NETWORK & SECURITY, LOAD BALANCING, AUTO SCALING, and more. Under LOAD BALANCING, 'Target Groups' is selected and highlighted with a red box. In the main content area, there's a table with one row for a target group named 'Loadbalancing'. The table columns include Name, Port, Protocol, Target type, Load Balancer, VPC ID, and Monitoring. Below the table, a section titled 'Target group: Loadbalancing' contains tabs for Description, Targets, Health checks, Monitoring, and Tags. The 'Targets' tab is selected. Under 'Basic Configuration', there are fields for Name (Loadbalancing), ARN (arn:aws:elasticloadbalancing:us-west-2:461067907136:targetgroup/Loadbalancing9adbb5a76039964a), and Port (HTTP). A status message indicates '1 to 1 of 1' targets registered.

Choose actions and select Register ip targets

This screenshot is similar to the previous one but focuses on the 'Actions' dropdown menu. The 'Actions' button is highlighted with a red box. A context menu is open, showing options: 'Edit health check', 'Register and deregister instance / IP targets' (which is highlighted with a red box), 'Edit attributes', and 'Delete'. The rest of the interface is identical to the first screenshot, showing the registered target group 'Loadbalancing'.

Here select the Autoscaling instances and click on save.

Register and deregister targets

Registered targets
To deregister instances, select one or more registered instances and then click Remove.

Instance	Name	Port	State	Security groups	Zone
No instances available.					

Instances
To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

Add to registered	on port 7000						
Search Instances	X						
Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR	
<input type="checkbox"/>	i-0261d0f69365847	my_server	running	default	us-west-2b	subnet-c7bd95be	172.31.16.0/20
<input checked="" type="checkbox"/>	i-0093d3cebdeb73dbb	Autoscalling_demo	running	AutoScaling-Security...	us-west-2b	subnet-c7bd95be	172.31.16.0/20
<input checked="" type="checkbox"/>	i-02d88b00814b1966	Autoscalling_demo	running	AutoScaling-Security...	us-west-2c	subnet-cc241e96	172.31.0.0/20

Cancel **Save**

To attach load balancer to auto scaling group click on auto scaling groups and then select actions dropdown

Create Auto Scaling group Actions

Try the new design for Amazon EC2 Auto Scaling
This older console is being replaced with the new EC2 Auto Scaling console. No new features or improvements will be made in this older console. Go to the new console.

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace Period
Autoscaling-gr...	Autoscalling	2	2	2	2	us-west-2b, us-west-2c	300	300

Auto Scaling Group: Autoscaling-group

Details Activity History Scaling Policies Instances Monitoring Notifications Tags Scheduled Actions Lifecycle Hooks

Launch Configuration Autoscalling	Desired Capacity 2	Availability Zone(s) us-west-2b, us-west-2c
Desired Capacity 2	Subnet(s) subnet-c7bd95be, subnet-cc241e96	Classic Load Balancers 1
Min 2	Target Groups 1	Health Check Type EC2
Max 2	Health Check Grace Period 300	Instance Protection Default
	Termination Policies Default	Suspended Processes 1
	Max Instance Lifetime 1	Placement Groups 1

Edit

https://us-west-2.console.aws.amazon.com/ec2/autoscaling/home?region=us-west-2#LaunchCo...

Click on edit.

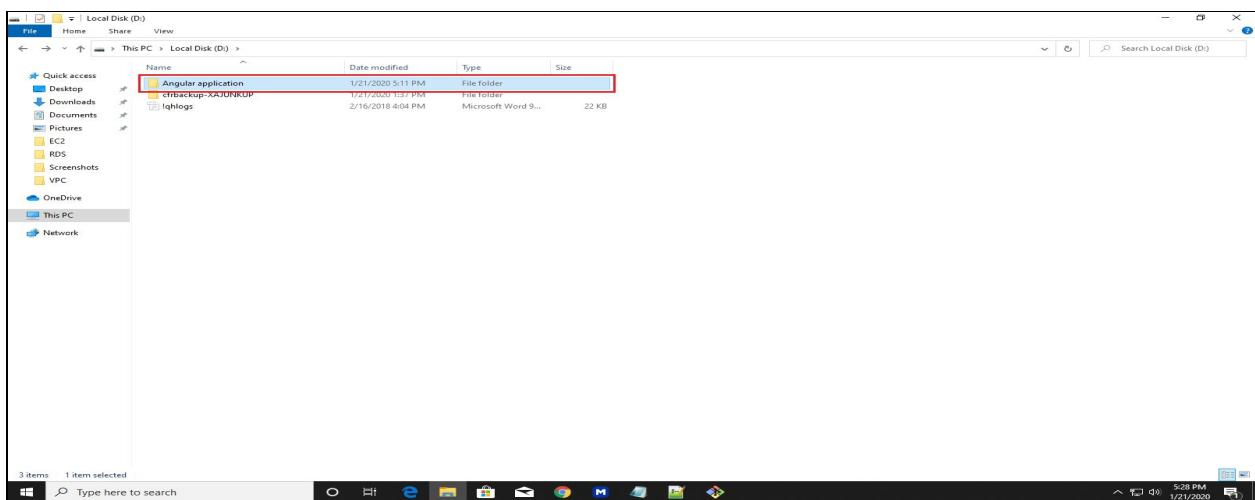
The screenshot shows the AWS Auto Scaling console. In the top navigation bar, there is a message: "Try the new design for Amazon EC2 Auto Scaling. This older console is being replaced with the new EC2 Auto Scaling console. No new features or improvements will be made in this older console. Go to the new console." Below this, there is a table titled "Create Auto Scaling group" with one row. The "Actions" column for this row contains three buttons: "Edit" (highlighted with a red box), "Delete", and a third button. The table has columns for Name, Launch Configuration, Instances, Desired, Min, Max, Availability Zones, Default Cooldown, and Health Check Grace Period. The row shows "Autoscalling-gr..." under Name, "Autoscalling" under Launch Configuration, and values 2, 2, 2, 2, "us-west-2b, us-west-2c", 300, and 300 respectively. Below the table, there is a section titled "Auto Scaling Group: Autoscalling-group" with tabs for Details, Activity History, Scaling Policies, Instances, Monitoring, Notifications, Tags, Scheduled Actions, and Lifecycle Hooks. The Details tab is selected. It shows the Launch Configuration set to "Autoscalling", Desired Capacity at 2, and Subnet(s) as "subnet-c7bd95be(172.31.16.0/20) | Default in us-west-2b" and "subnet-cc241e96(172.31.0.0/20) | Default in us-west-2c". Other settings include Availability Zone(s) "us-west-2b, us-west-2c", Classic Load Balancers (empty), Target Groups (empty), Health Check Type "EC2", Health Check Grace Period 300, Instance Protection (empty), Termination Policies "Default", Suspended Processes (empty), Max Instance Lifetime (empty), and Placement Groups (empty). At the bottom of the page, there is a footer with links for Feedback, English (US), Privacy Policy, and Terms of Use.

Here choose the load balancer target group and click on the save as shown below.

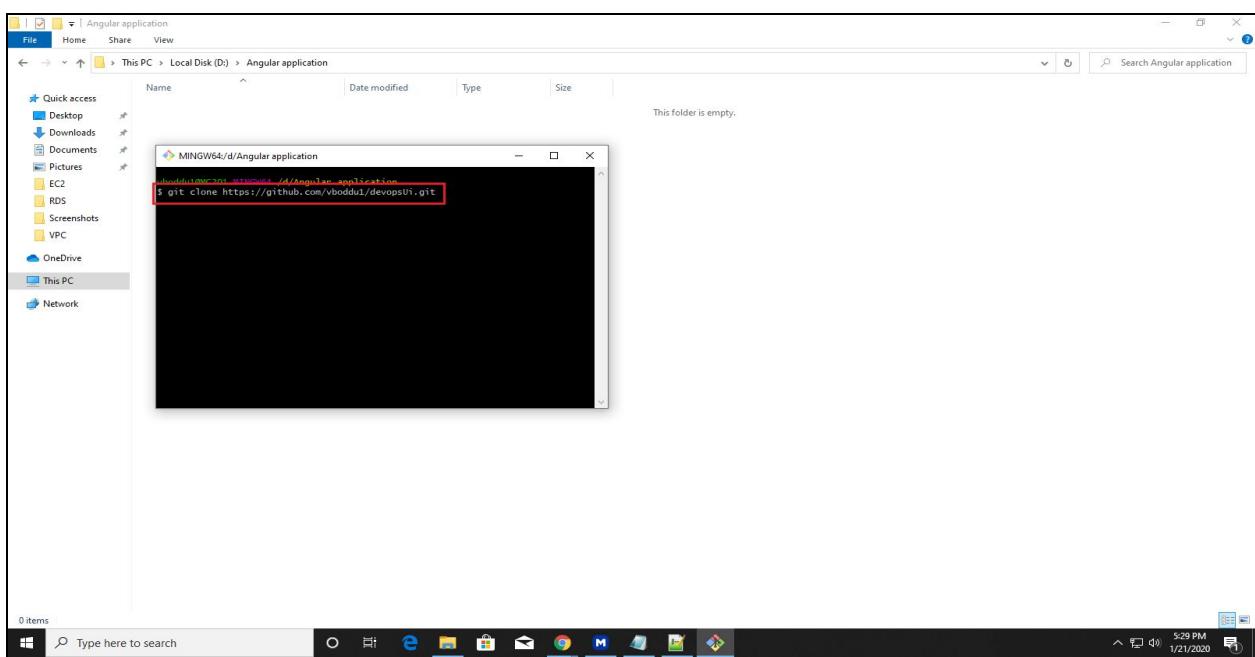
The screenshot shows the "Edit details - Autoscaling-group" dialog box. The "Launch Instances Using" section has "Launch Configuration" selected. The "Launch Configuration" dropdown is set to "Autoscalling". The "Desired Capacity" is set to 2. The "Availability Zone(s)" dropdown is set to "us-west-2b, us-west-2c". The "Subnet(s)" dropdown lists "subnet-c7bd95be(172.31.16.0/20) | Default in us-west-2b" and "subnet-cc241e96(172.31.0.0/20) | Default in us-west-2c". The "Classic Load Balancers" dropdown is empty. The "Target Groups" input field contains "Loadbalancing" and is highlighted with a red box. The "Health Check Type" is set to "EC2". The "Health Check Grace Period" is set to 300. The "Instance Protection" and "Termination Policies" dropdowns are empty. The "Suspended Processes" dropdown is empty. At the bottom right of the dialog box, there are "Cancel" and "Save" buttons, with "Save" also highlighted with a red box. The background shows the same AWS Auto Scaling console interface as the previous screenshot.

Static Web Hosting on S3

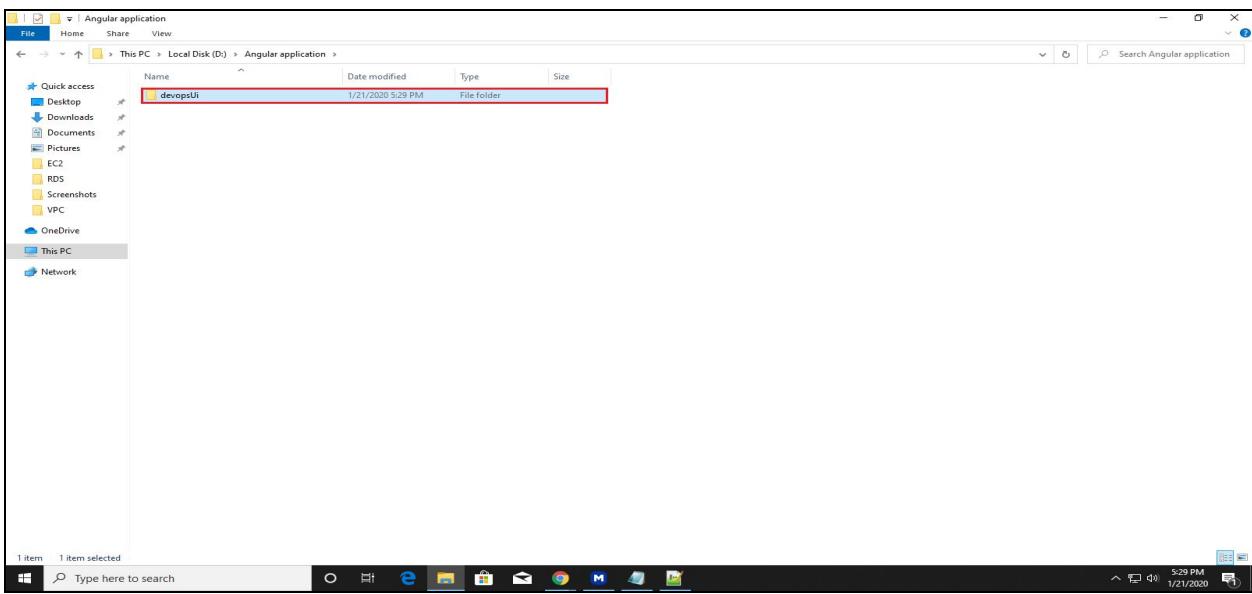
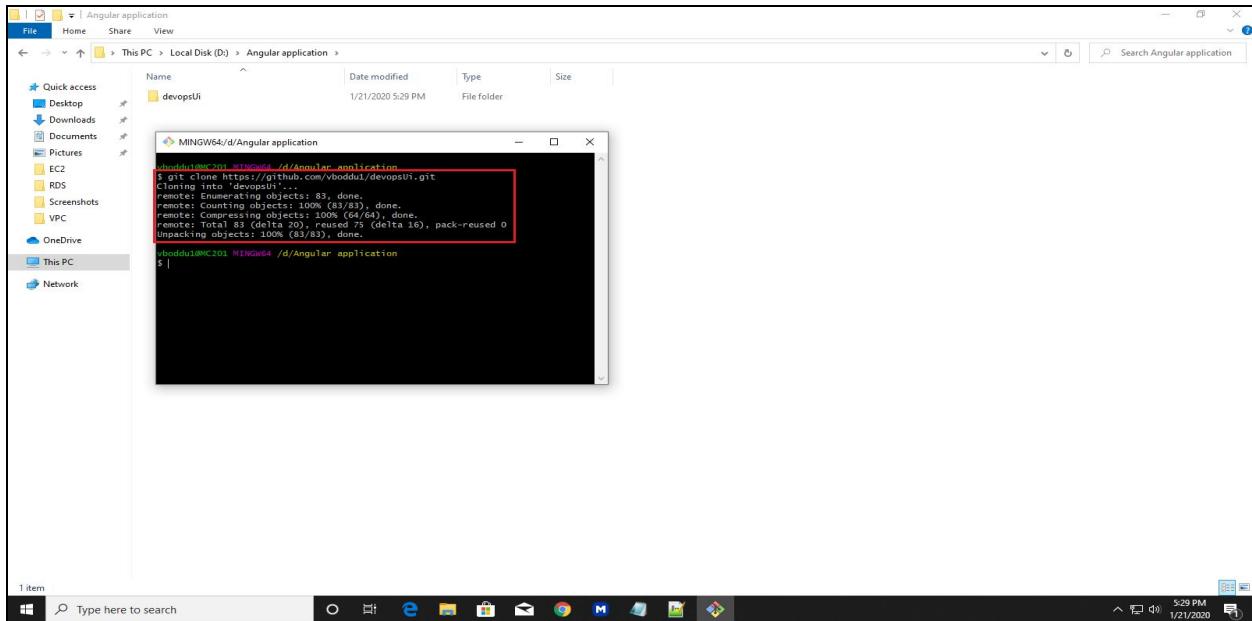
Create a folder



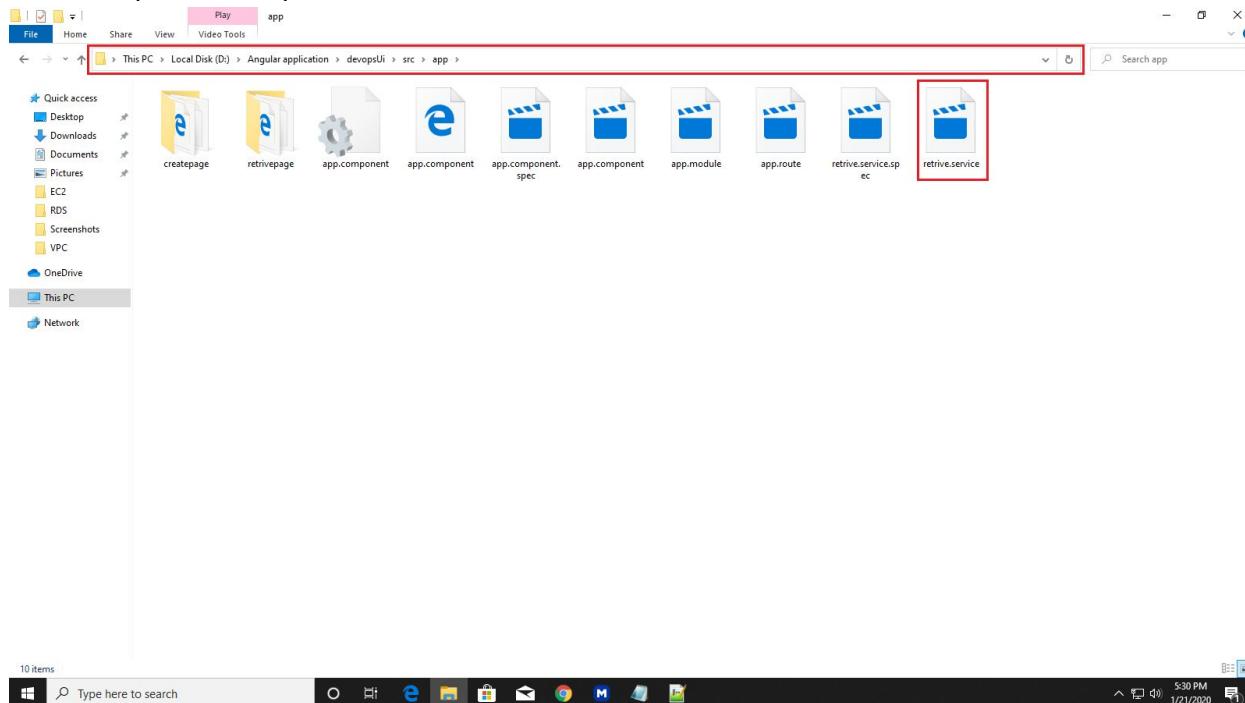
Clone your application using below git URL <https://github.com/vboddu1/devopsUi.git>



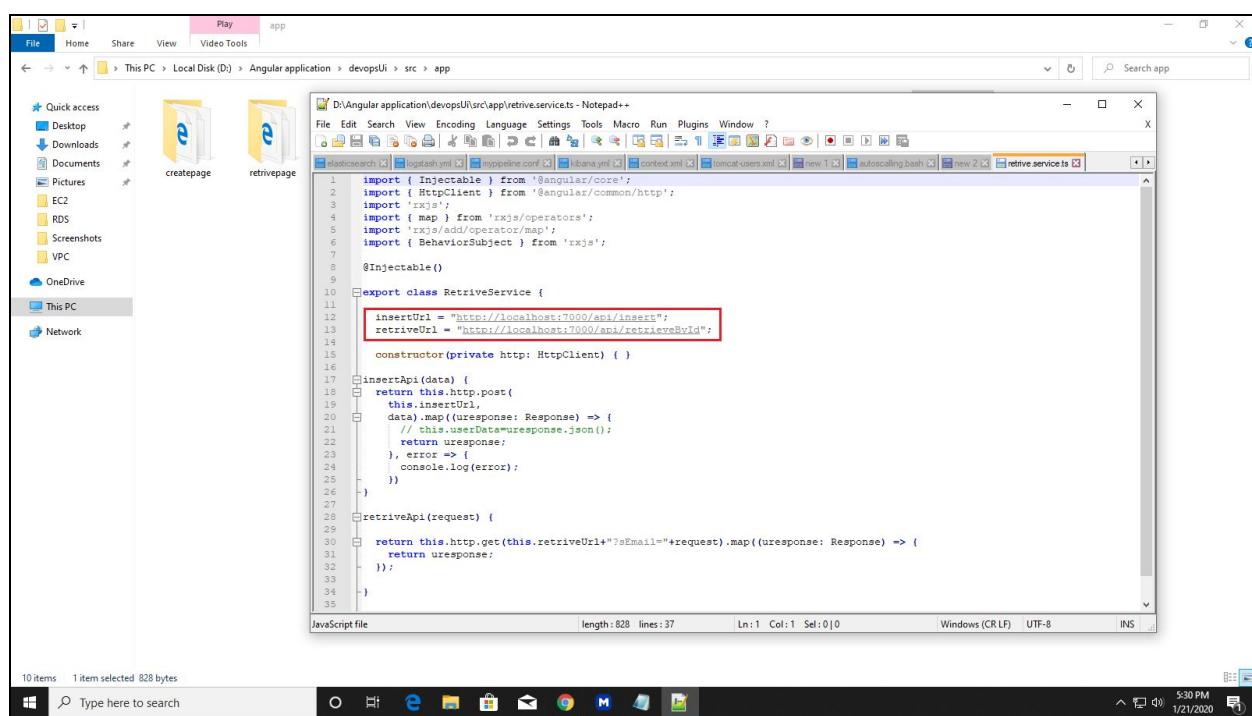
Your application was cloned successfully

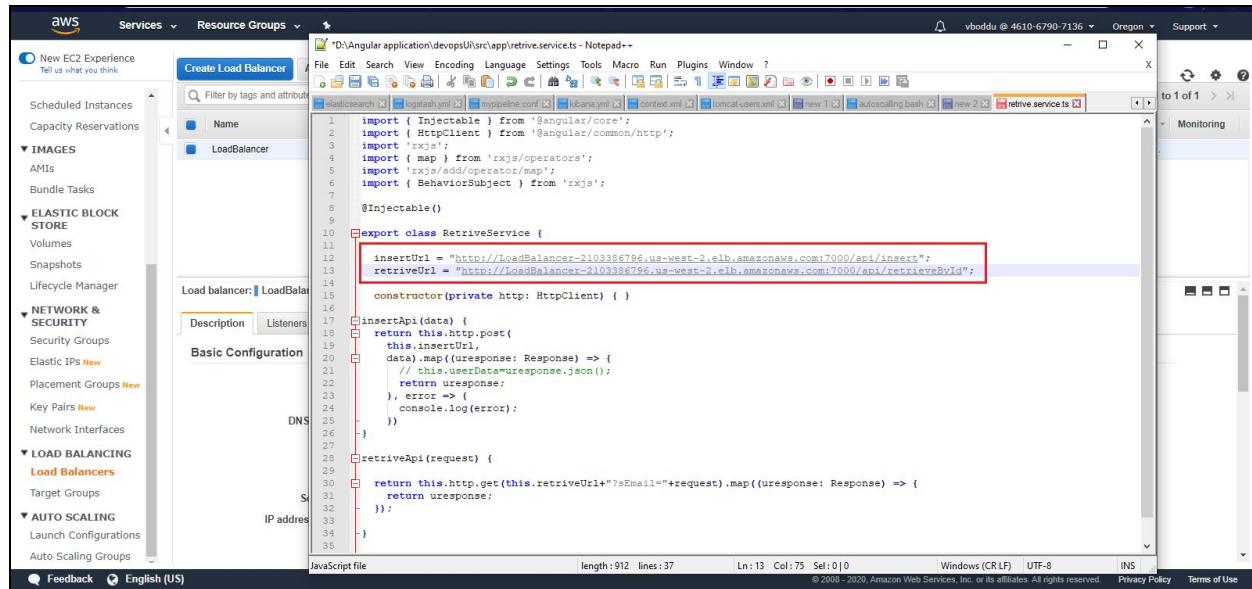


Go to the path and open retrieve.service

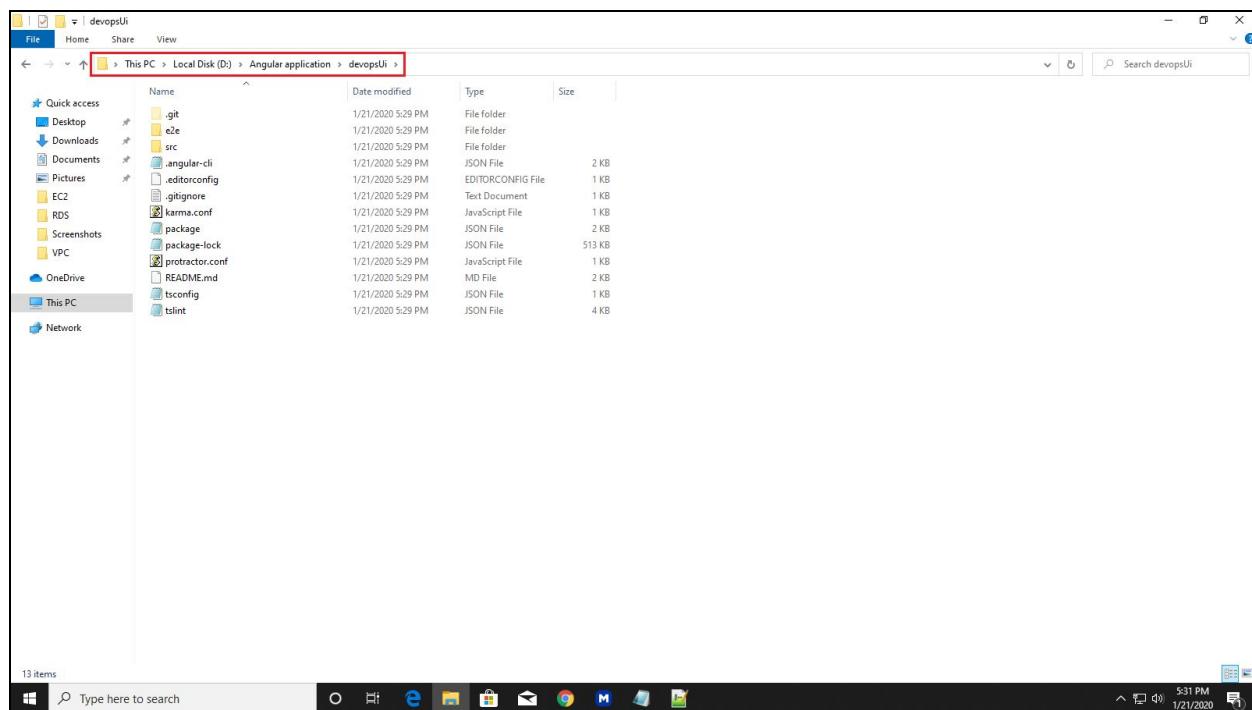


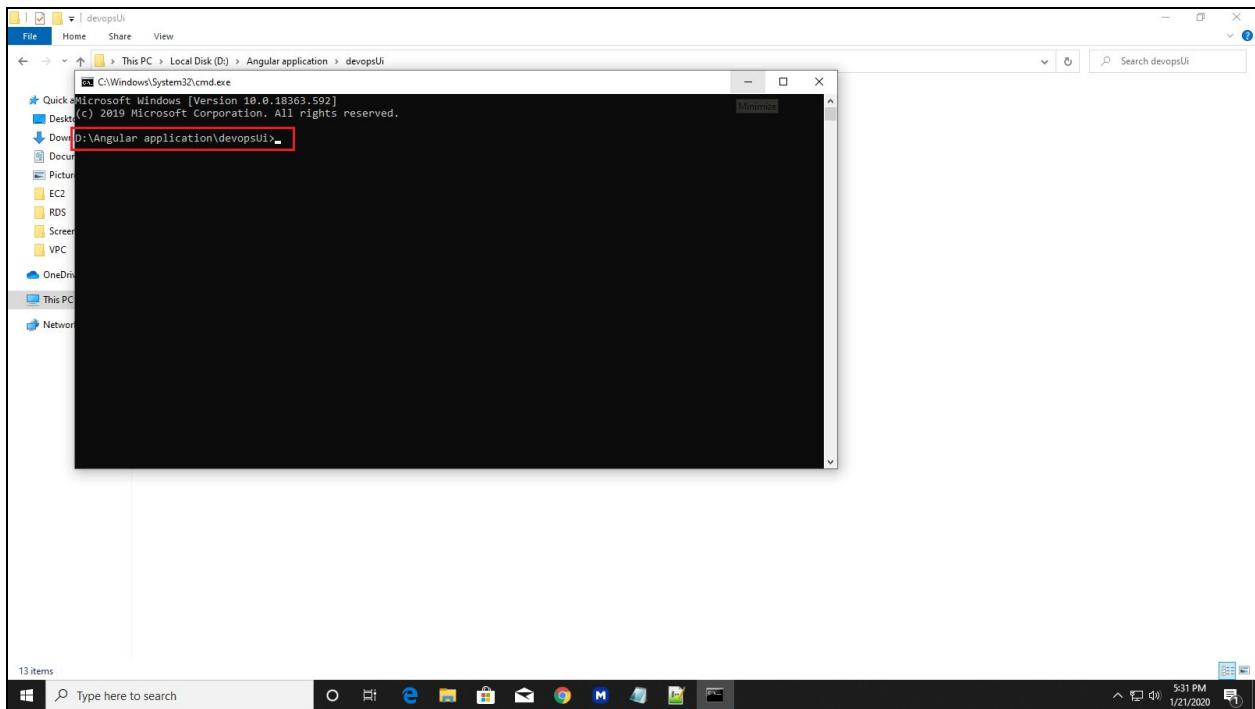
Replace localhost with load balancing URL





Once you are done with your modifications open cmd





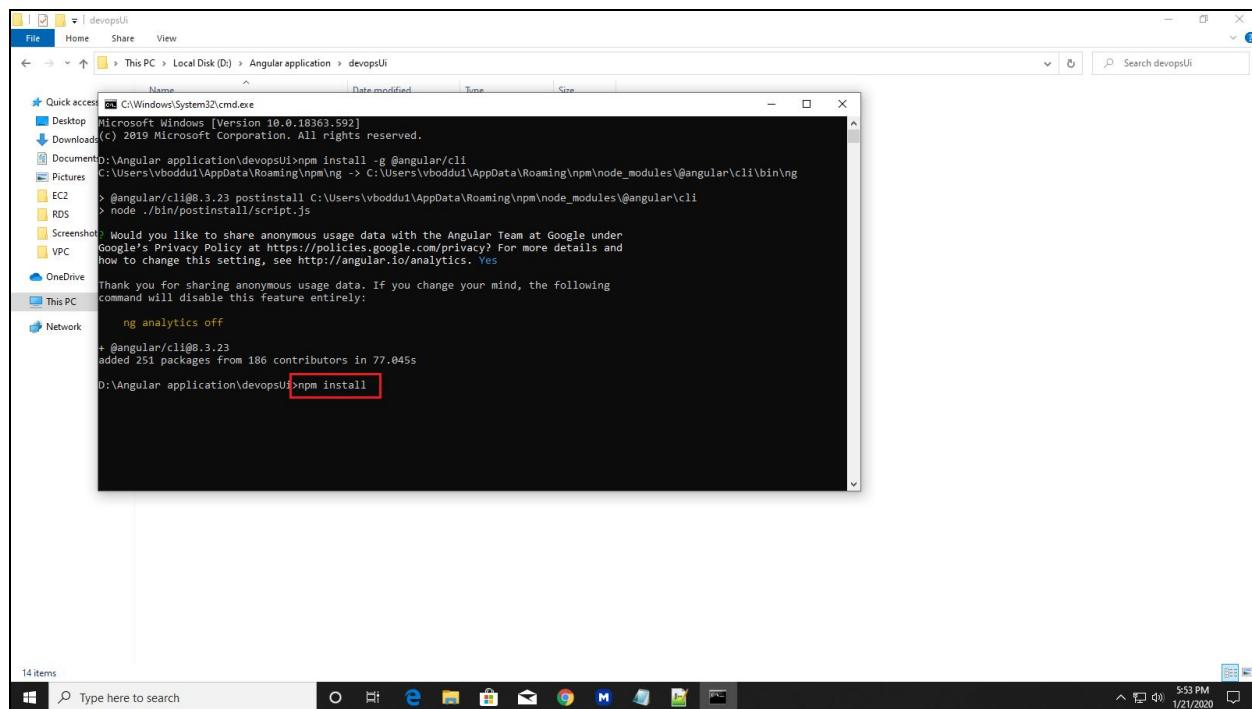
Run the below command to install angular/cli

The image consists of three vertically stacked screenshots of a Windows desktop environment. Each screenshot shows a file explorer window open to the path 'This PC > Local Disk (D:) > Angular application > devopsUi'. A command prompt window is overlaid on the file explorer. The first screenshot shows the command 'npm install -g @angular/cli' being typed. The second screenshot shows the command being executed, with the output including the creation of a symbolic link at 'C:\Users\vboddul\AppData\Roaming\npm\ng' pointing to 'C:\Users\vboddul\AppData\Roaming\npm\node_modules\@angular\cli\bin\ng'. The third screenshot shows the final output of the command, which includes a prompt asking if the user wants to share anonymous usage data with the Angular team. The user has selected 'y' to accept.

```
devopsUi
File Home Share View
← → ↑ This PC > Local Disk (D:) > Angular application > devopsUi
Name Date modified Time Size
Quick access C:\Windows\System32\cmd.exe
Desktop Microsoft Windows [Version 10.0.18363.592]
Download (c) 2019 Microsoft Corporation. All rights reserved.
Document D:\Angular application\devopsUi>npm install -g @angular/cli
Pictures EC2 RDS Screenshot VPC
OneDrive This PC Network

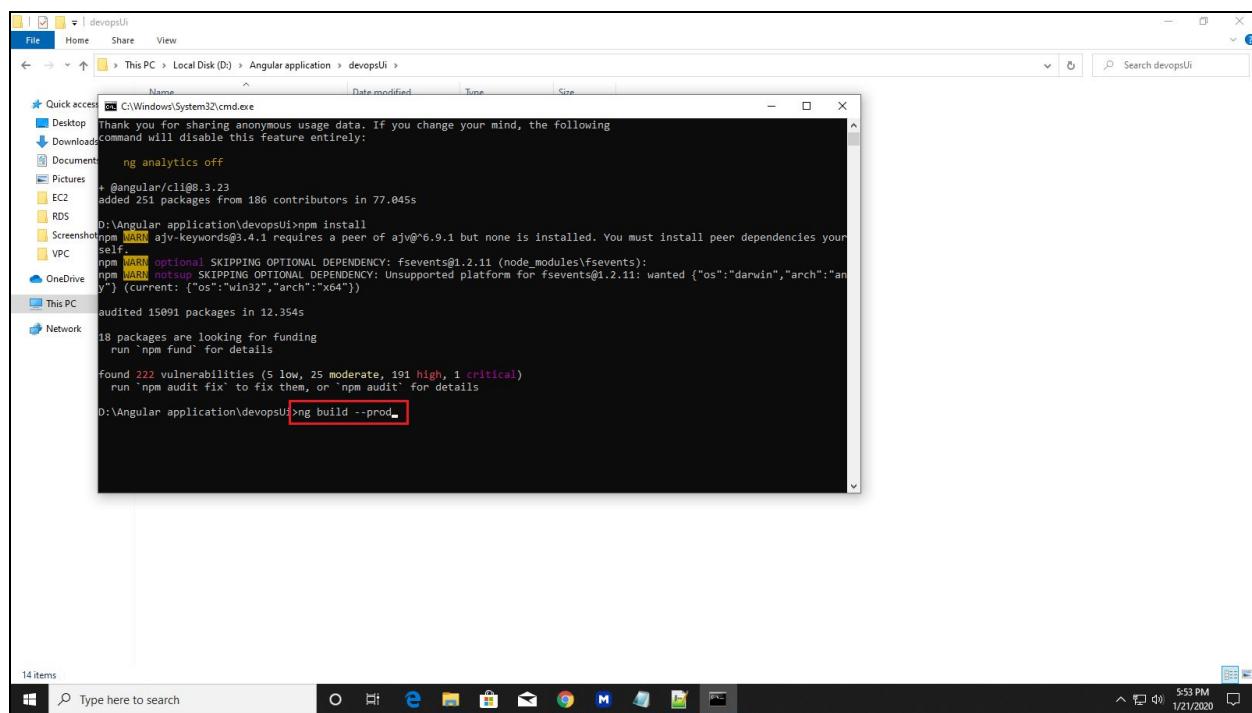
14 items
Type here to search 5:51 PM 1/21/2020
File Home Share View
← → ↑ This PC > Local Disk (D:) > Angular application > devopsUi
Name Date modified Time Size
Quick access npm
Desktop Microsoft Windows [Version 10.0.18363.592]
Download (c) 2019 Microsoft Corporation. All rights reserved.
Document D:\Angular application\devopsUi>npm install -g @angular/cli
C:\Users\vboddul\AppData\Roaming\npm\ng -> C:\Users\vboddul\AppData\Roaming\npm\node_modules\@angular\cli\bin\ng
> @angular/cli@8.3.23 postinstall C:\Users\vboddul\AppData\Roaming\npm\node_modules\@angular\cli
> node ./bin/postinstall/script.js
Would you like to share anonymous usage data with the Angular Team at Google under
Google's Privacy Policy at https://policies.google.com/privacy? For more details and
how to change this setting, see http://angular.io/analytics. (Y/N) y_
14 items
Type here to search 5:53 PM 1/21/2020
```

Run the below command to install node modules



D:\Angular application\devopsUi>npm install -g @angular/cli
 C:\Users\vboddu1\AppData\Roaming\npm\ng -> C:\Users\vboddu1\AppData\Roaming\npm\node_modules\@angular\cli\bin\ng
 > @angular/cli@8.3.23 postinstall C:\Users\vboddu1\AppData\Roaming\npm\node_modules\@angular\cli
 > node ./bin/postinstall/script.js
 Would you like to share anonymous usage data with the Angular Team at Google under
 Google's Privacy Policy at https://policies.google.com/privacy? For more details and
 how to change this setting, see http://angular.io/analytics. Yes
 Thank you for sharing anonymous usage data. If you change your mind, the following
 command will disable this feature entirely:
 ng analytics off
 + @angular/cli@8.3.23
 added 251 packages from 186 contributors in 77.045s
 D:\Angular application\devopsUi>npm install

Run the below command to build your application and to create dist folder



D:\Angular application\devopsUi>npm install
 Thank you for sharing anonymous usage data. If you change your mind, the following
 command will disable this feature entirely:
 ng analytics off
 + @angular/cli@8.3.23
 added 251 packages from 186 contributors in 77.045s
 D:\Angular application\devopsUi>ng build --prod
 ajv-keywords@3.4.1 requires a peer of ajv@^6.9.1 but none is installed. You must install peer dependencies yourself.
 npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.11 (node_modules\fsevents):
 npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.11: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
 audited 15091 packages in 12.354s
 18 packages are looking for funding
 run 'npm fund' for details
 found 222 vulnerabilities (5 low, 25 moderate, 191 high, 1 critical)
 run 'npm audit fix' to fix them, or 'npm audit' for details
 D:\Angular application\devopsUi>

```

File Home Share View
Name Date modified Type Size
C:\Windows\System32\cmd.exe

D:\Angular application\devopsUi>npm install
npm WARN ajv-keywords@3.4.1 requires a peer of ajv@^6.9.1 but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.11 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.11: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
audited 15091 packages in 12.354s
18 packages are looking for funding
  run 'npm fund' for details
Found 222 vulnerabilities (5 low, 25 moderate, 191 high, 1 critical)
  run 'npm audit fix' to fix them, or 'npm audit' for details

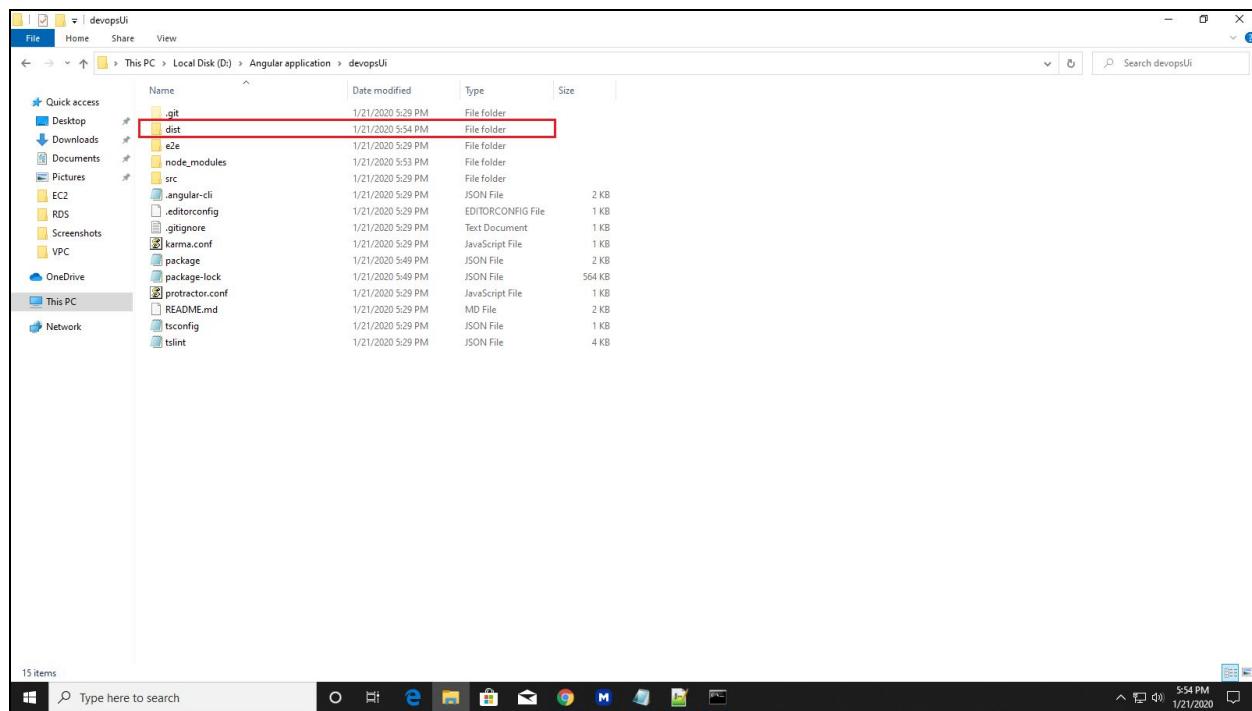
D:\Angular application\devopsUi>ng build --prod
Your global Angular CLI version (8.3.23) is greater than your local
version (1.6.6). The local Angular CLI version is used.

To disable this warning use "ng config -g cli.warnings.versionMismatch false".
Date: 2020-01-21T12:24:22.992Z
Hash: c6868be17d411df871c0
Time: 2000ms
chunk {0} main.5cd8d578be5114ceaa3c.bundle.js (main) 540 kB [initial] [rendered]
chunk {1} polyfills.3305061234c35bd995e8c.bundle.js (polyfills) 59.9 kB [initial] [rendered]
chunk {2} styles.9cad0738f18adc3d19ed.bundle.css (styles) 79 bytes [initial] [rendered]
chunk {3} inline.3bb2a3758f1fe0653182.bundle.js (inline) 1.45 kB [entry] [rendered]

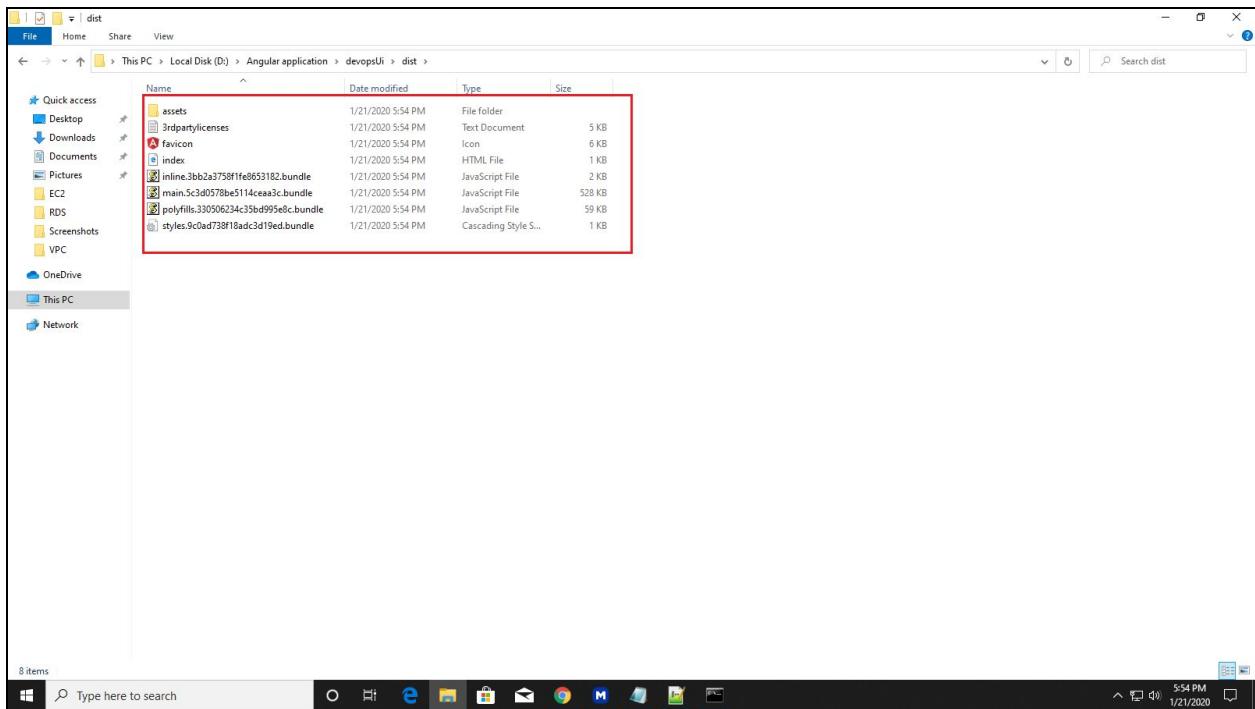
D:\Angular application\devopsUi>

```

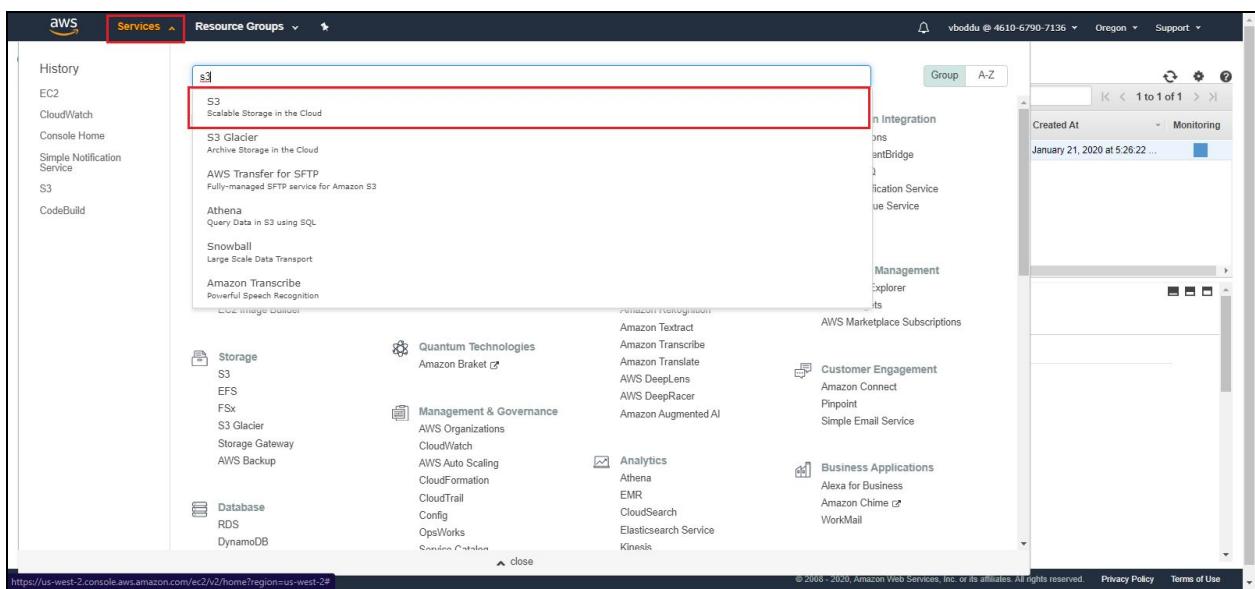
Once your application successfully built the dist folder will be created as shown below



Copy all the files that are present in your dist folder



Go to your AWS console and search for S3



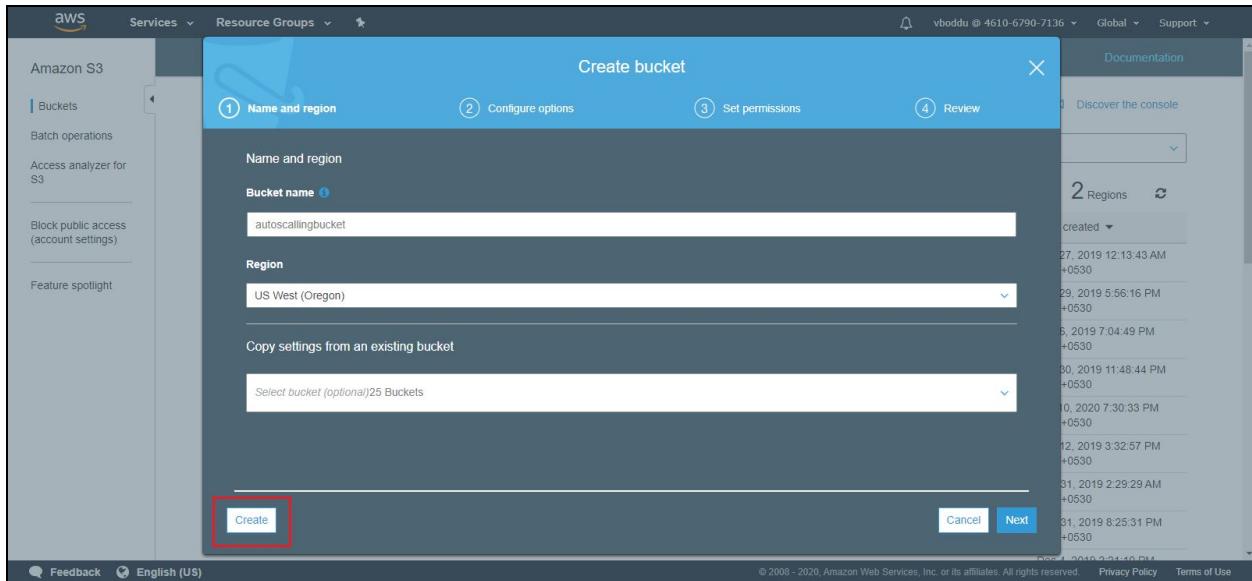
Click on create bucket

The screenshot shows the AWS S3 service dashboard. On the left, there's a sidebar with options like 'Buckets', 'Batch operations', 'Access analyzer for S3', 'Block public access (account settings)', and 'Feature spotlight'. The main area is titled 'S3 buckets' and shows a list of existing buckets. At the top of this list, there's a blue button labeled '+ Create bucket'. To the right of the list, it says '25 Buckets' and '2 Regions'. Below the list, there are filters for 'Bucket name', 'Access', 'Region', and 'Date created'. The list itself includes entries such as 'cf-templates-1cqmpt1snh9e-us-east-1', 'cf-templates-1cqmpt1snh9e-us-east-2', 'codepipeline-us-east-1-83728269471', 'codepipeline-us-east-2-608898523880', 'deeplens-sagemaker-signdemo', 'demoawsmeetup', 'dev-mss-contracts', and 'dev.msscontracts.tk'. Each entry has a checkbox, an icon, and columns for Region and Date created.

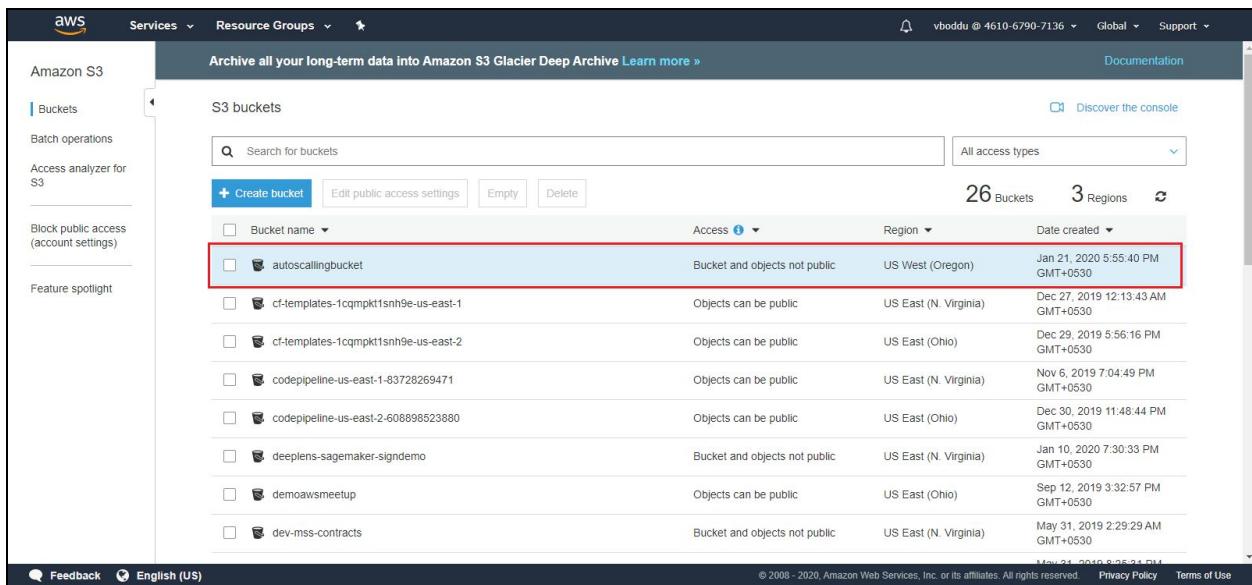
Provide S3 bucket name and region

This screenshot shows the 'Create bucket' wizard. It has four steps: 1. Name and region, 2. Configure options, 3. Set permissions, and 4. Review. Step 1 is active. It has fields for 'Bucket name' (containing 'autoscalingbucket') and 'Region' (set to 'US West (Oregon)'). There's also a section for 'Copy settings from an existing bucket' with a dropdown menu showing 'Select bucket (optional) 25 Buckets'. At the bottom are 'Cancel' and 'Next' buttons. The background shows a list of existing buckets with their creation dates and regions.

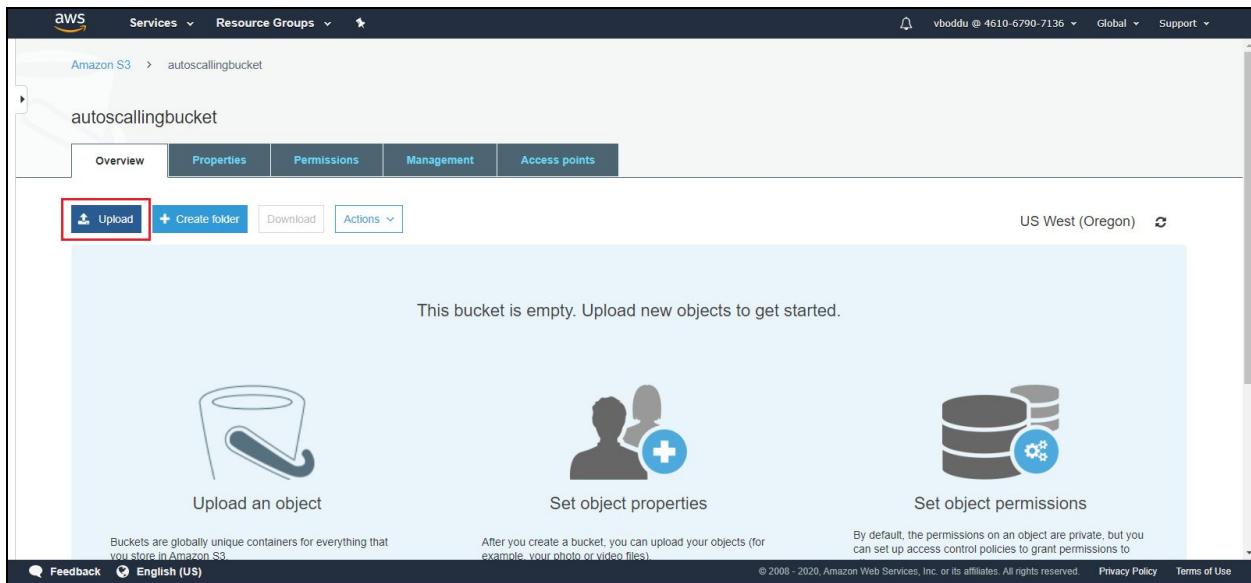
Click on create



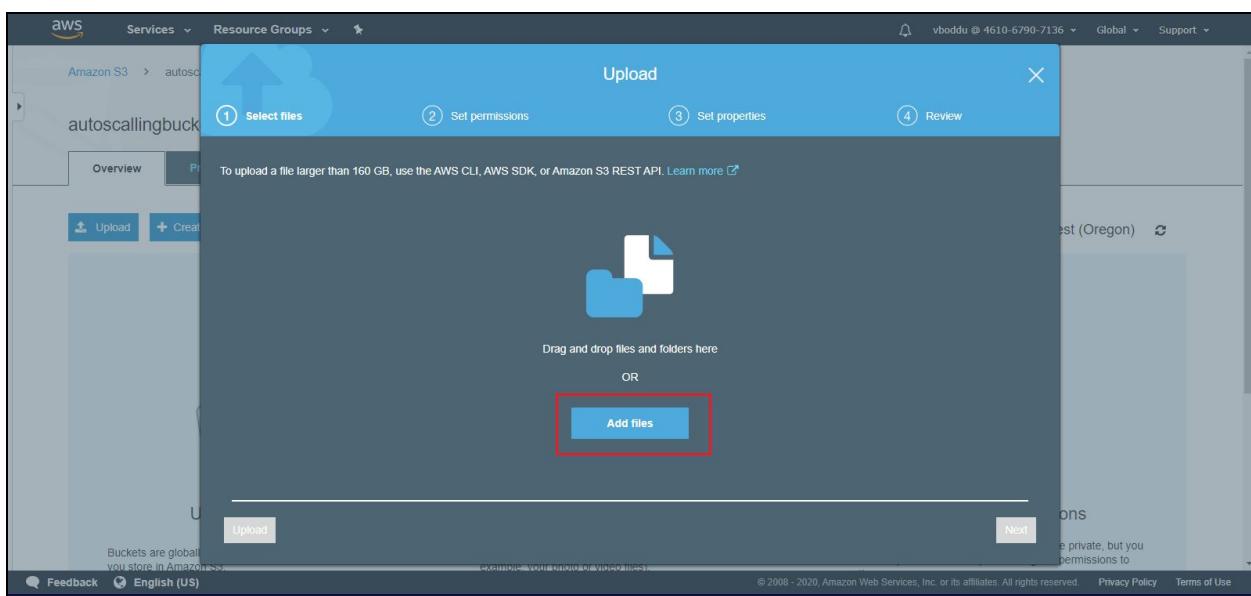
Your bucket will be created as shown below, click on your bucket



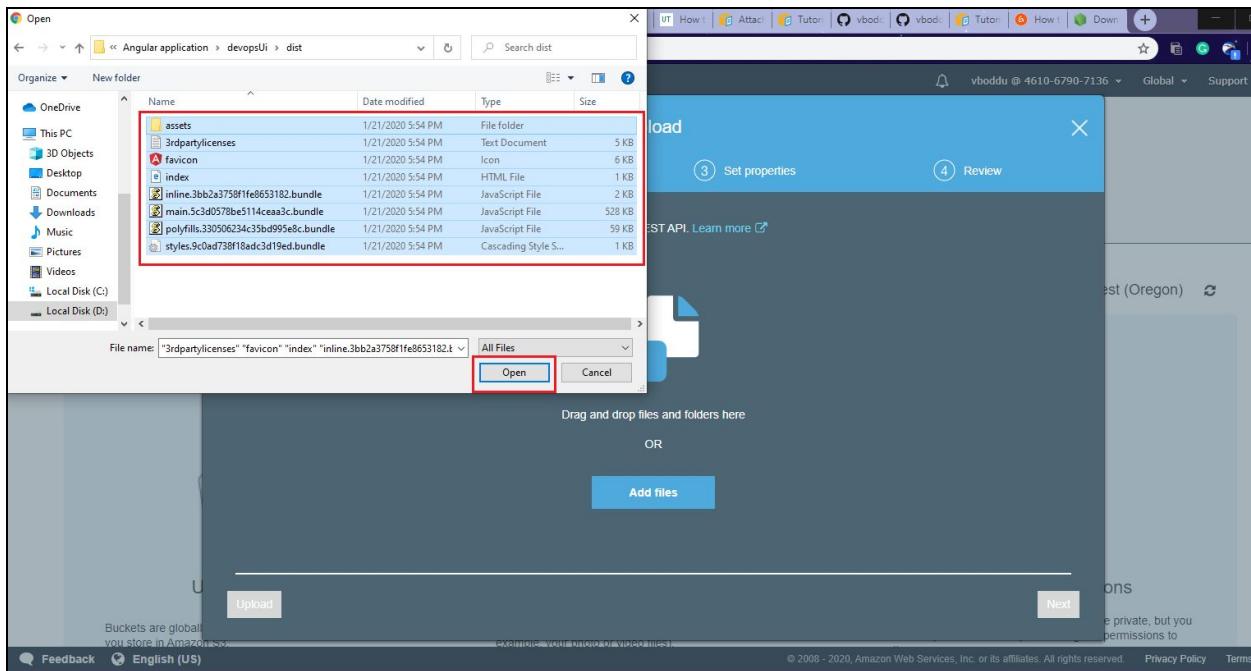
Click on upload



Click on the Add files



Go to your Angular application dist folder and then drag and drop all the files onto your S3 bucket



All the files in your dist folder will get uploaded onto your S3 bucket as shown below

The screenshot shows the AWS S3 console with the path 'Amazon S3 > autoscallingbucket'. The 'Properties' tab is selected. The file list shows the following files:

Name	Last modified	Size	Storage class
assets	Jan 21, 2020 5:56:28 PM GMT+0530	4.5 KB	Standard
3rdpartylicenses.txt	Jan 21, 2020 5:56:28 PM GMT+0530	5.3 KB	Standard
favicon.ico	Jan 21, 2020 5:56:28 PM GMT+0530	934.0 B	Standard
index.html	Jan 21, 2020 5:56:28 PM GMT+0530	1.4 KB	Standard
inline.3bb2a3758f1fe8653182.bundle.js	Jan 21, 2020 5:56:29 PM GMT+0530	527.4 KB	Standard
main.5c3d0578be5114ceaa3c.bundle.js	Jan 21, 2020 5:56:30 PM GMT+0530	58.5 KB	Standard
polyfills.330506234c35bd995e8c.bundle.js	Jan 21, 2020 5:56:30 PM GMT+0530	79.0 B	Standard
styles.9c8ad738f18adc3d19ed.bundle.css	Jan 21, 2020 5:56:29 PM GMT+0530		

Click on Permissions, select block public access and then click on edit

Amazon S3 > autoscallingbucket

autoscallingbucket

Overview Properties Permissions Management Access points

Block public access Access Control List Bucket Policy CORS configuration

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access

Off

- Block public access to buckets and objects granted through new access control lists (ACLs)
- Block public access to buckets and objects granted through any access control lists (ACLs)
- Block public access to buckets and objects granted through new public bucket or access point policies

Operations 0 In progress 4 Success 0 Error

https://s3.console.aws.amazon.com/s3/#

Disable block all public access and click on save

Amazon S3 > autoscallingbucket

autoscallingbucket

Overview Properties Permissions Management Access points

Block public access Access Control List Bucket Policy CORS configuration

Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another:

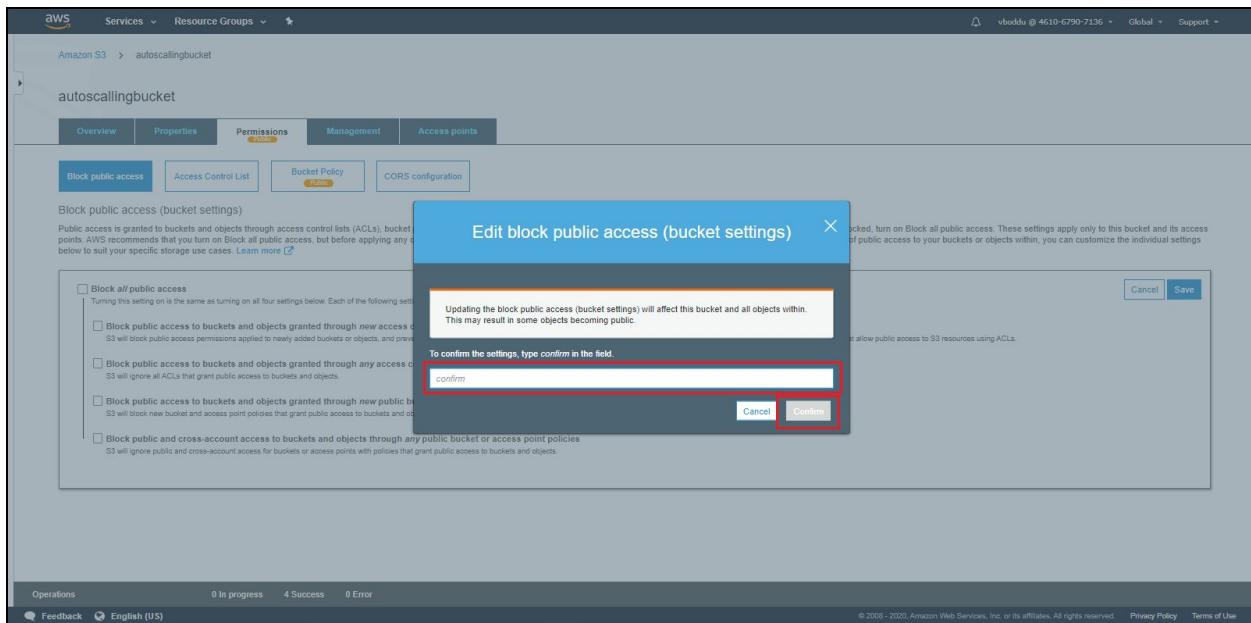
- Block public access to buckets and objects granted through new access control lists (ACLs)
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Cancel Save

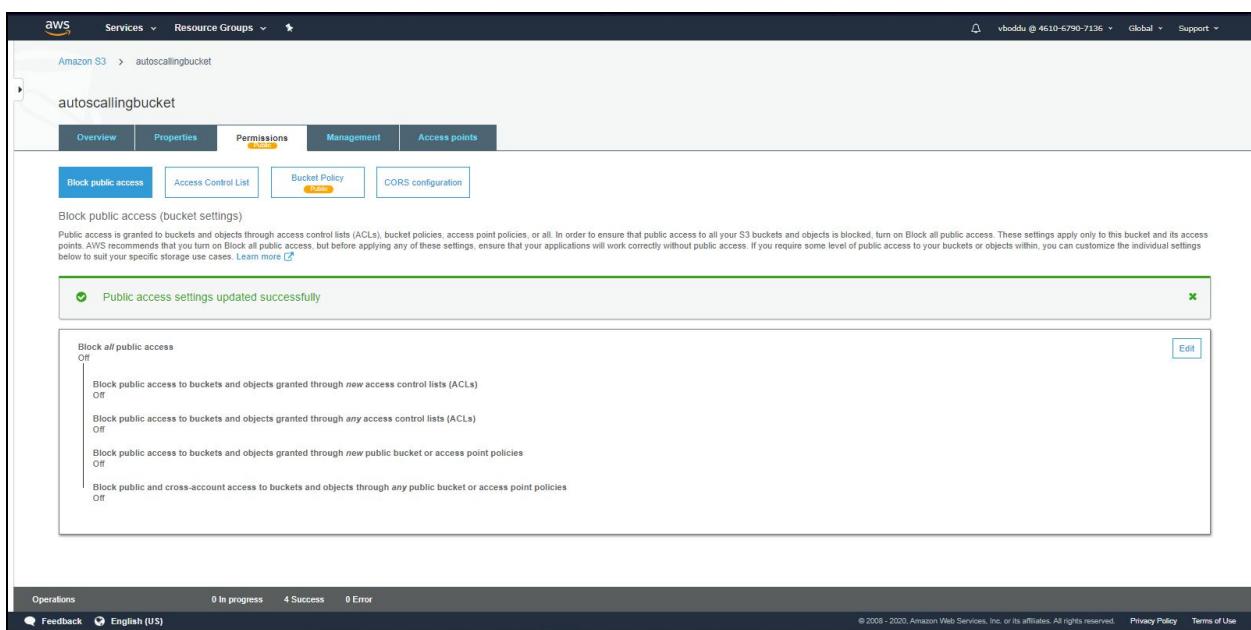
Operations 0 In progress 4 Success 0 Error

Feedback English (US)

It will ask for confirmation so type confirm and click on confirm



Public access setting were updated successfully



Click on bucket policy and provide the below script and then click on save

The screenshot shows the AWS S3 console for the 'autoscallingbucket'. In the 'Bucket Policy' tab, a JSON policy is displayed:

```

1 {
2     "Version": "2012-10-17",
3     "Statement": [
4         {
5             "Sid": "PublicRead",
6             "Effect": "Allow",
7             "Principal": "*",
8             "Action": "s3:GetObject",
9             "Resource": "arn:aws:s3:::autoscallingbucket/*"
10        }
11    ]
12 }

```

The 'Save' button at the bottom right is highlighted with a red box.

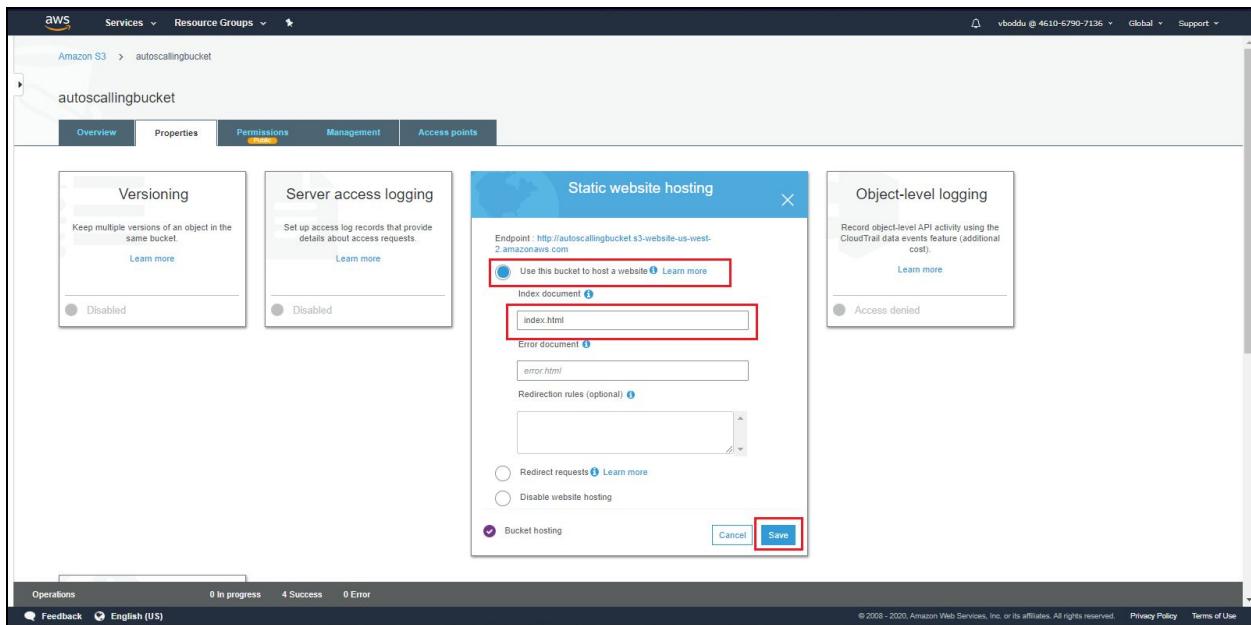
Click on properties and select static web hosting

The screenshot shows the AWS S3 console for the 'autoscallingbucket'. In the 'Properties' tab, the 'Static website hosting' section is highlighted with a red box:

Static website hosting
 Host a static website, which does not require server-side technologies.
 Bucket hosting

The 'Save' button at the bottom right is highlighted with a red box.

Select use the bucket to host a website and provide index.html and then click on save



Now click on the end point to access the application

