Placement Empowerment Program

Cloud Computing and DevOps Centre

Implement Auto-scaling in the CloudSet up an auto- scaling group for your cloud VMs to handle variable workloads.

Name: Kevin Christopher Ruban   Department:AI-ML

Introduction

As modern applications face varying workloads, ensuring optimal performance and availability is critical. Auto Scaling, a feature provided by cloud platforms like AWS, dynamically adjusts computing resources in response to demand changes. This Proof of Concept (PoC) demonstrates how to set up an Auto Scaling Group (ASG) for virtual machines (VMs) to handle fluctuating workloads effectively. It explores defining launch configurations, setting scaling policies, and testing automatic scaling based on CPU usage.

Overview

This PoC focuses on implementing a scalable architecture using AWS Auto Scaling Groups. The workflow includes:

1. **Defining a Launch Template**: Configuring virtual machines (VMs) with required specifications like instance type, AMI, key pairs, and security groups.

2. **Creating an Auto Scaling Group**: Setting initial group size and linking it to the launch template to manage instances dynamically.

3. **Configuring Scaling Policies**: Setting up metrics like CPU utilization to trigger scaling actions (e.g., scaling up during high CPU usage).

4. **Testing Auto Scaling**: Simulating high CPU load to verify that the ASG launches additional instances to handle demand.

This PoC will demonstrate the reliability, flexibility, and cost-efficiency of dynamic scaling in a cloud environment.

Objective

The primary objective of this PoC is to:

1. Implement an **Auto Scaling Group (ASG)** to manage workloads effectively.
2. Define and configure a **Launch Template** for virtual machines.
3. Set up and test **scaling policies** based on predefined metrics, such as CPU utilization.
4. Validate the scaling process by simulating real-world scenarios (e.g., high CPU usage).

By completing this PoC, the goal is to gain hands-on experience with Auto Scaling and to understand its importance in ensuring application availability and cost management.

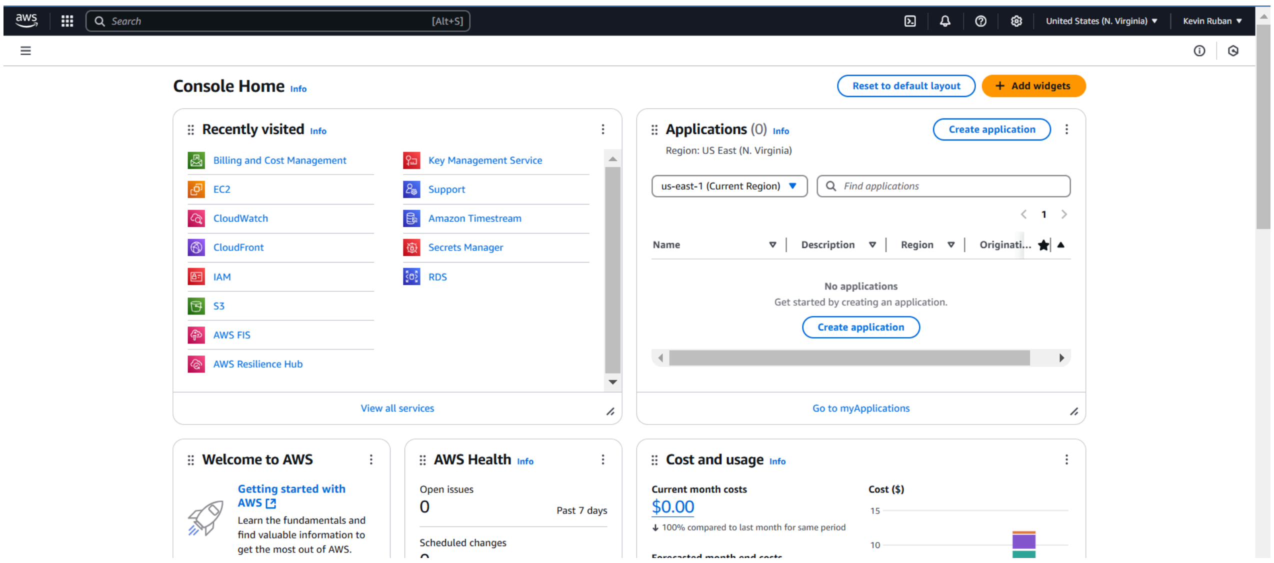
Importance

1. **Improved Application Availability**: Auto Scaling ensures that applications remain available even during traffic spikes by automatically adding more VMs to meet demand.
2. **Cost Optimization**: It dynamically reduces the number of VMs during low traffic periods, minimizing unnecessary costs.
3. **Efficient Resource Utilization**: By scaling resources based on actual demand, Auto Scaling prevents over-provisioning and underutilization.
4. **Resilience to Failures**: Auto Scaling can replace unhealthy instances automatically, ensuring consistent application performance.
5. **Real-World Relevance**: The ability to manage variable workloads is a critical skill in cloud computing and aligns with industry practices.

Step-by-Step Overview

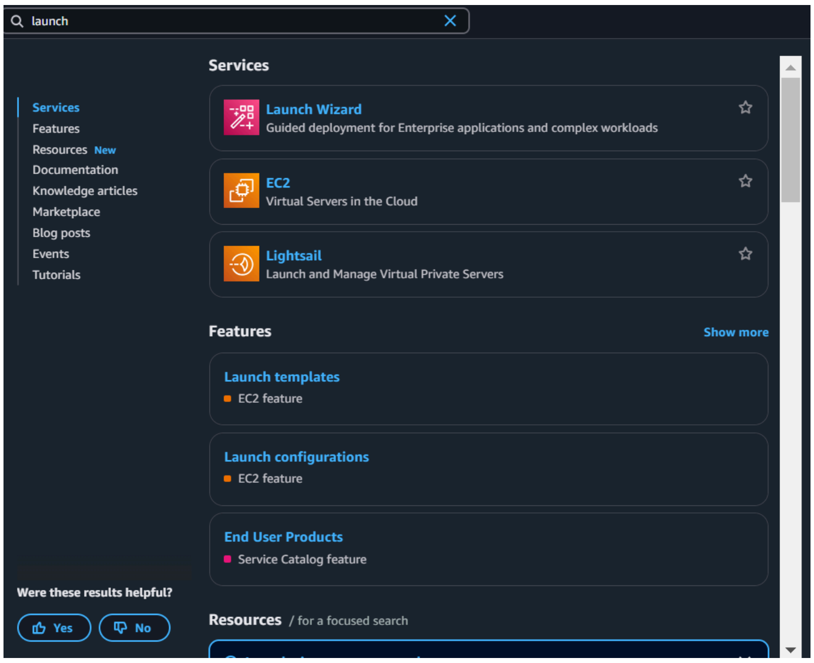
Step 1:

1. Go to [AWS Management Console](https://console.aws.amazon.com).
2. Enter your username and password to log in.



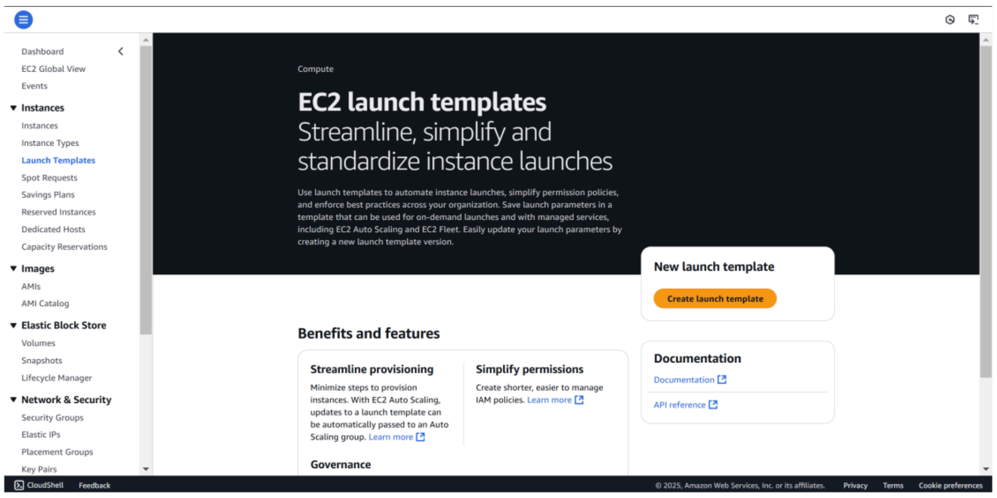
Step 2:

Search for Launch Templates.



Step 3:

Click on the Create launch template.



EC2 launch templates
Streamline, simplify and standardize instance launches

Use launch templates to automate instance launches, simplify permission policies, and enforce best practices across your organization. Save launch parameters in a template that can be used for on-demand launches and with managed services, including EC2 Auto Scaling and EC2 Fleet. Easily update your launch parameters by creating a new launch template version.

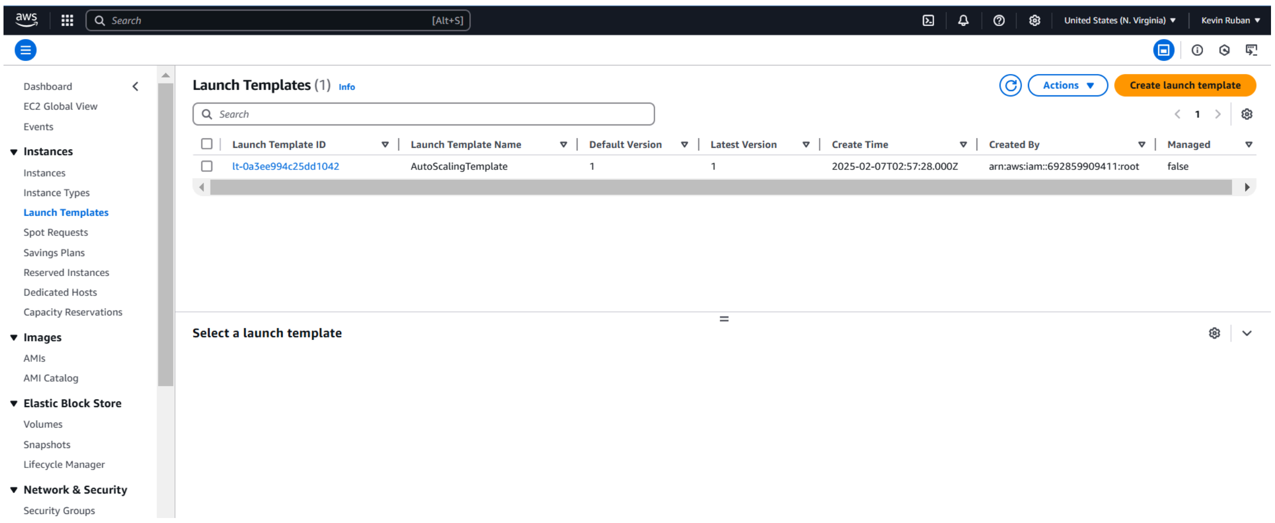
Benefits and features

Streamline provisioning
Minimize steps to provision instances. With EC2 Auto Scaling, updates to a launch template can be automatically passed to an Auto Scaling group. Learn more

Simplify permissions
Create shorter, easier to manage IAM policies. Learn more

Step 4:

Create a **Launch Template** named **AutoScalingTemplate** using an **Amazon Machine Image (AMI)** like Amazon Linux 2 or any default image, and choose an **instance type** such as **t2.micro** for free-tier eligibility. Select an **existing key pair** (or create a new one) to enable SSH access, and configure a **security group** that allows HTTP (port 80) and SSH (port 22). Once all details are filled out, click **Create launch template** to complete the setup.

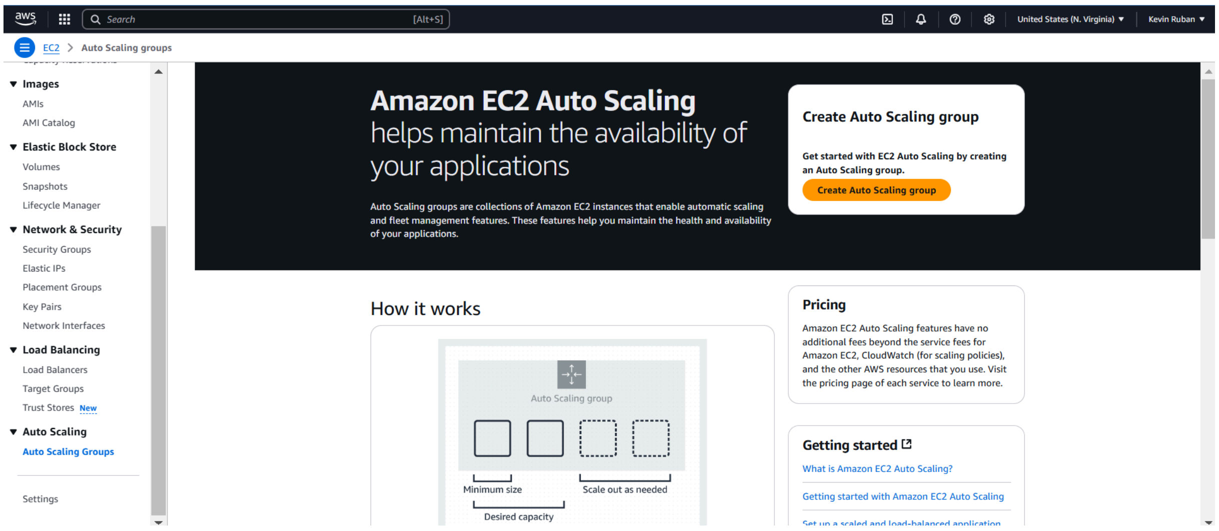


Launch Templates (1)

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By	Managed
lt-0a3ee994c25dd1042	AutoScalingTemplate	1	1	2025-02-07T02:57:28.000Z	arn:aws:iam::692859909411:root	false

Step 5:

Go to the **EC2 Dashboard** . On the left sidebar, click on **Auto Scaling Groups**. Click on **Create an Auto Scaling group**.


Search [Alt+S]
United States (N. Virginia) ▾ Kevin Ruban ▾

EC2 > Auto Scaling groups

▾ Images
AMIs
AMI Catalog
▾ Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
▾ Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
▾ Load Balancing
Load Balancers
Target Groups
Trust Stores New
▾ Auto Scaling
Auto Scaling Groups

Settings

Amazon EC2 Auto Scaling
helps maintain the availability of your applications

Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. These features help you maintain the health and availability of your applications.

Create Auto Scaling group
Get started with EC2 Auto Scaling by creating an Auto Scaling group.
Create Auto Scaling group

How it works
Auto Scaling group
Minimum size
Scale out as needed
Desired capacity

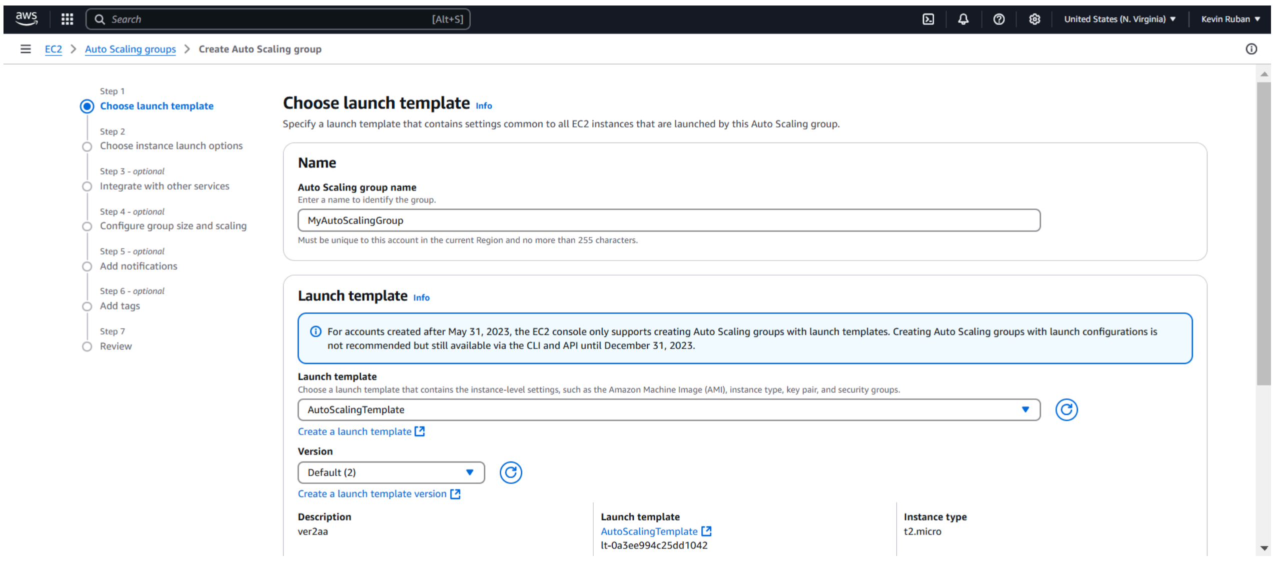
Pricing
Amazon EC2 Auto Scaling features have no additional fees beyond the service fees for Amazon EC2, CloudWatch (for scaling policies), and the other AWS resources that you use. Visit the pricing page of each service to learn more.

Getting started
What is Amazon EC2 Auto Scaling?
Getting started with Amazon EC2 Auto Scaling
Set up a scaled and load-balanced application

Step 6:

Auto Scaling group name: Give it a name (e.g., MyAutoScalingGroup).

Launch Template: Select the launch template you created earlier (AutoScalingTemplate).


Search [Alt+S]
United States (N. Virginia) ▾ Kevin Ruban ▾

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template
Step 2
Choose instance launch options
Step 3 - *optional*
Integrate with other services
Step 4 - *optional*
Configure group size and scaling
Step 5 - *optional*
Add notifications
Step 6 - *optional*
Add tags
Step 7
Review

Choose launch template Info
Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name
Auto Scaling group name
Enter a name to identify the group.
MyAutoScalingGroup
Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info
For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

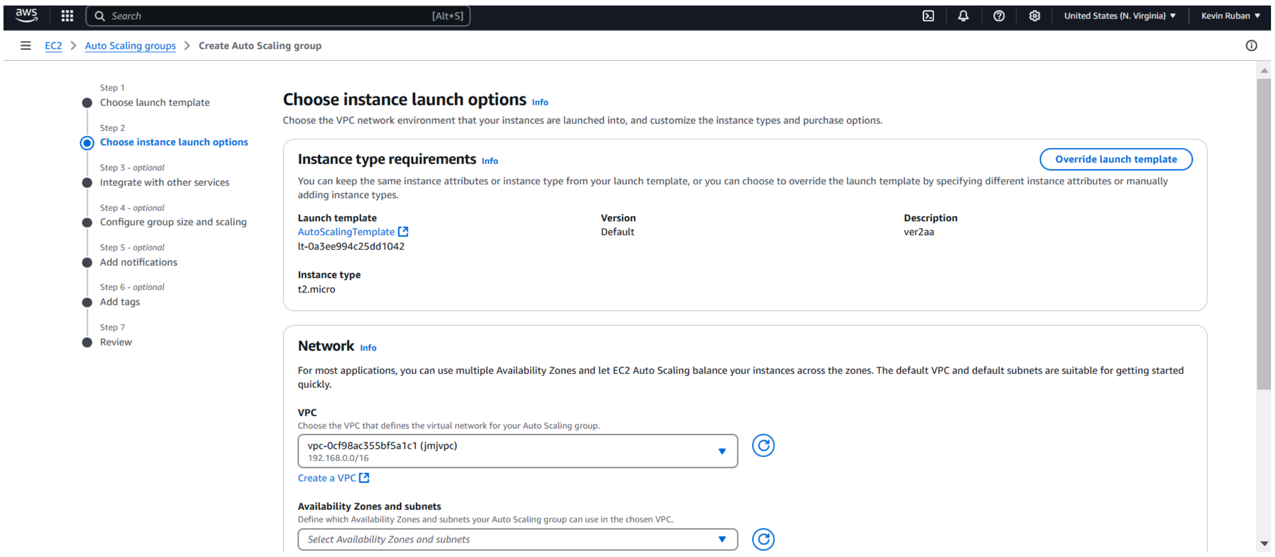
Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
AutoScalingTemplate ▾
Create a launch template

Version
Default (2) ▾
Create a launch template version

Description	Launch template	Instance type
ver2aa	AutoScalingTemplate lt-0a3ee994c25dd1042	t2.micro

Step 7:

VPC and Subnets: Choose your **VPC** (it's fine to use the default one). Select at least two subnets in different Availability Zones (this ensures high availability).



Search [Alt+S] United States (N. Virginia) ▼ Kevin Ruban ▼

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template

Step 2
Choose instance launch options

Step 3 - *optional*
Integrate with other services

Step 4 - *optional*
Configure group size and scaling

Step 5 - *optional*
Add notifications

Step 6 - *optional*
Add tags

Step 7
Review

Choose instance launch options Info

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Instance type requirements Info — Override launch template

You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

Launch template	Version	Description
AutoScalingTemplate lt-0a3ee994c25dd1042	Default	ver2aa

Instance type
t2.micro

Network Info

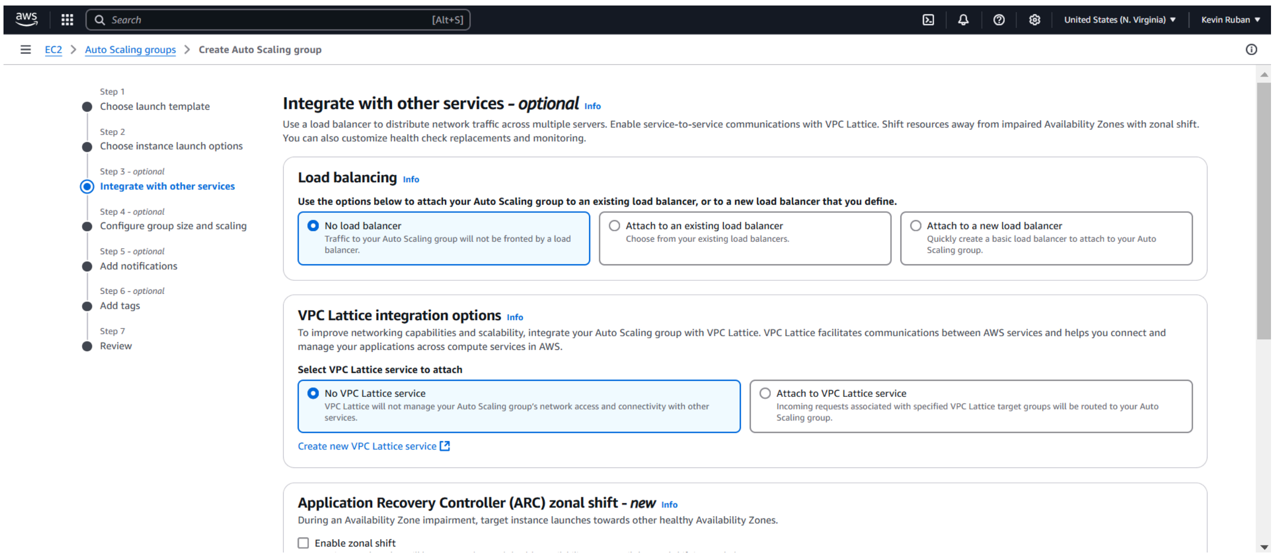
For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
vpc-0cf98ac355bf5a1c1 (jmjvpc)
192.168.0.0/16
Create a VPC

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.
Select Availability Zones and subnets

Step 8:

For this PoC leave the next settings as default and click next .



Search [Alt+S] United States (N. Virginia) ▼ Kevin Ruban ▼

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template

Step 2
Choose instance launch options

Step 3 - *optional*
Integrate with other services

Step 4 - *optional*
Configure group size and scaling

Step 5 - *optional*
Add notifications

Step 6 - *optional*
Add tags

Step 7
Review

Integrate with other services - *optional* Info

Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.

Load balancing Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

- [✓ No load balancer] — Traffic to your Auto Scaling group will not be fronted by a load balancer.
- Attach to an existing load balancer — Choose from your existing load balancers.
- Attach to a new load balancer — Quickly create a basic load balancer to attach to your Auto Scaling group.

VPC Lattice integration options Info

To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

Select VPC Lattice service to attach

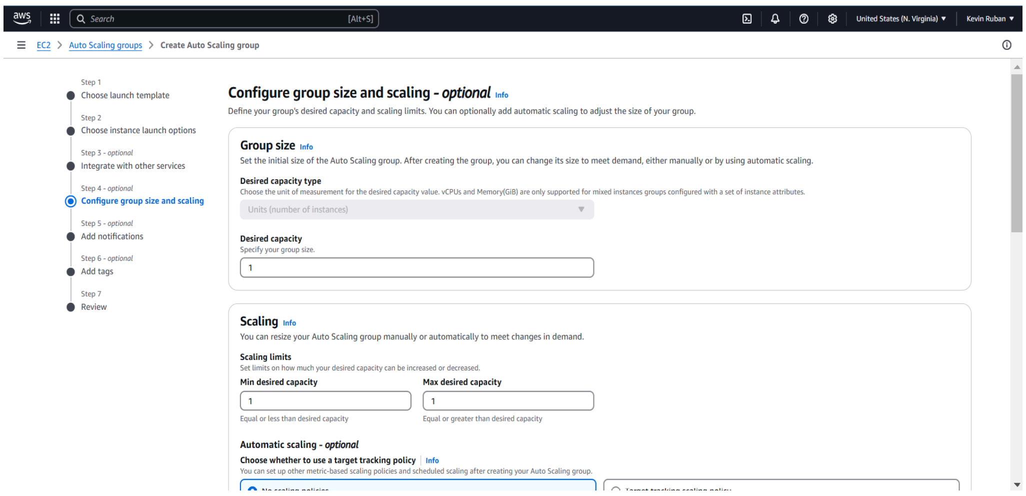
- [✓ No VPC Lattice service] — VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.
- Attach to VPC Lattice service — Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

Create new VPC Lattice service

Application Recovery Controller (ARC) zonal shift - *new* Info

During an Availability Zone impairment, target instance launches towards other healthy Availability Zones.

☐ Enable zonal shift


Search [Alt+S]
United States (N. Virginia)
Kevin Ruban
EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1
Choose launch template
Step 2
Choose instance launch options
Step 3 - optional
Integrate with other services
Step 4 - optional
Configure group size and scaling
Step 5 - optional
Add notifications
Step 6 - optional
Add tags
Step 7
Review

Configure group size and scaling - *optional* Info

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

Group size Info

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances)

Desired capacity

Specify your group size.

1

Scaling Info

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

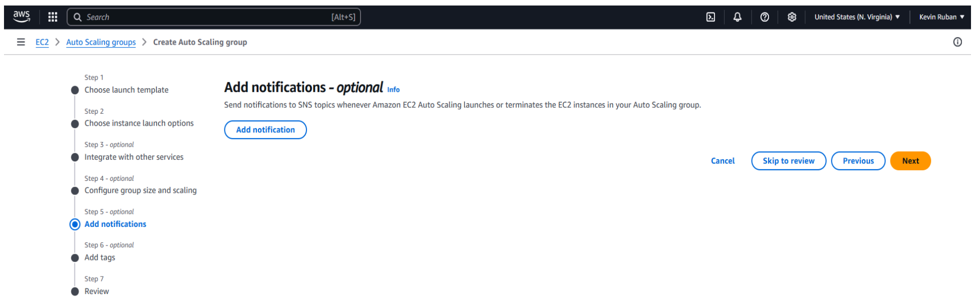
Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity	Max desired capacity
1	1
Equal or less than desired capacity	Equal or greater than desired capacity

Automatic scaling - *optional*

Choose whether to use a target tracking policy Info

You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

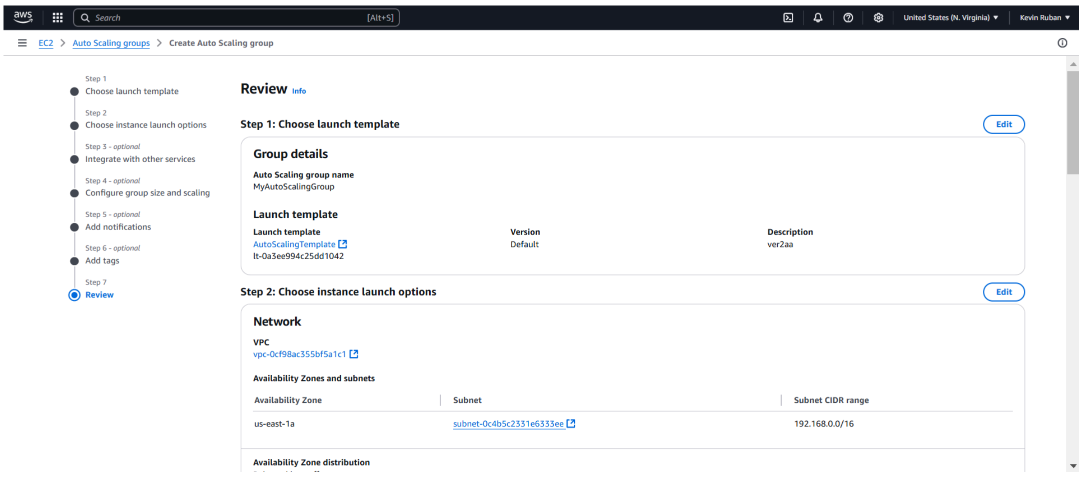


Add notifications - *optional* Info

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

Step 9:

Review all the settings you've configured. Once satisfied, click **Create an Auto Scaling Group**.


EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1
Choose launch template
Step 2
Choose instance launch options
Step 3 - *optional*
Integrate with other services
Step 4 - *optional*
Configure group size and scaling
Step 5 - *optional*
Add notifications
Step 6 - *optional*
Add tags
Step 7
Review

Review Info

Step 1: Choose launch template

Group details

Auto Scaling group name
MyAutoScalingGroup

Launch template

Launch template	Version	Description
AutoScalingTemplate lt-0a3ee994c25dd1042	Default	ver2aa

Step 2: Choose instance launch options

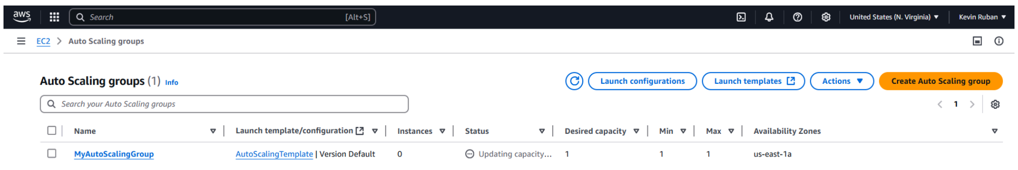
Network

VPC
vpc-0cf98ac355bf5a1c1

Availability Zones and subnets

Availability Zone	Subnet	Subnet CIDR range
us-east-1a	subnet-0c4b5c2331e6333ee	192.168.0.0/16

Availability Zone distribution

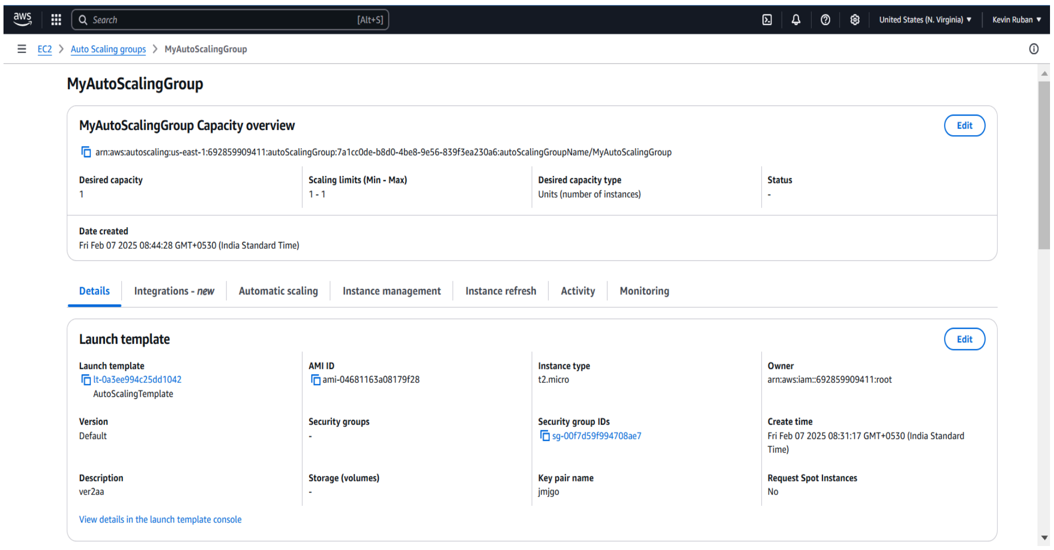

EC2 > Auto Scaling groups

Auto Scaling groups (1) Info

Launch configurations | Launch templates | Actions | Create Auto Scaling group

Search your Auto Scaling groups

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
MyAutoScalingGroup	AutoScalingTemplate | Version Default	0	Updating capacity...	1	1	1	us-east-1a


EC2 > Auto Scaling groups > MyAutoScalingGroup

MyAutoScalingGroup

MyAutoScalingGroup Capacity overview

arn:aws:autoscaling:us-east-1:692859909411:autoScalingGroup:7a1cc0de-b8d0-4be8-9e56-839f3ea230a6:autoScalingGroupName/MyAutoScalingGroup

Desired capacity	Scaling limits (Min - Max)	Desired capacity type	Status
1	1 - 1	Units (number of instances)	-

Date created
Fri Feb 07 2025 08:44:28 GMT+0530 (India Standard Time)

Details | Integrations - *new* | Automatic scaling | Instance management | Instance refresh | Activity | Monitoring

Launch template

Launch template lt-0a3ee994c25dd1042 AutoScalingTemplate	**AMI ID** ami-04681163a08179f28	**Instance type** t2.micro	**Owner** arn:aws:iam::692859909411:root
Version Default	**Security groups** -	**Security group IDs** sg-00f7d59f994708ae7	**Create time** Fri Feb 07 2025 08:31:17 GMT+0530 (India Standard Time)
Description ver2aa	**Storage (volumes)** -	**Key pair name** jmjgo	**Request Spot Instances** No

View details in the launch template console

Step 10:

Testing Auto Scaling :

Important Note

Do Not Perform This Test If You Want to Avoid Costs:

1. Launching and running additional EC2 instances will incur charges beyond the AWS Free Tier.

2. Simulating high CPU usage and triggering scaling may increase costs temporarily due to additional resource allocation.

1. **Simulate High CPU Usage on an EC2 Instance**

   Connect to one of your EC2 instances in the Auto Scaling Group using SSH.

   Run a command to create artificial CPU load. For example:

   ```
   sudo yum install -y stress

   stress --cpu 2 --timeout 300
   ```

   This command will utilize 2 CPU cores for 5 minutes, simulating high CPU usage.

2. **Monitor Scaling Activities**

   Navigate to the **AWS Management Console** > **EC2 Dashboard** > **Auto Scaling Groups**.

   Select your Auto Scaling Group and go to the **Activity History** tab.

Check if a new instance is being launched based on your scaling policy (e.g., CPU utilization exceeding 50%).

3. **Terminate the Stress Test**

   Once testing is done, stop the CPU load by pressing Ctrl+C in the terminal or by terminating the stress process.

4. **Verify Scaling Down**

   After the CPU usage drops, monitor the Auto Scaling Group again to confirm that unnecessary instances are terminated, returning to the desired capacity.

Outcome

This Proof of Concept (PoC) aimed to implement Auto Scaling in AWS to dynamically manage EC2 instances based on workload demand, ensuring efficient resource utilization and cost-effectiveness.

Here's the outcome of the PoC:

1. **Launch Template and Auto Scaling Group Setup**: Successfully created a launch template and configured an Auto Scaling Group with scaling policies to dynamically manage EC2 instances based on workload.

2. **Dynamic Scaling and Monitoring**: Implemented scaling policies triggered by CPU utilization and verified automatic scaling actions using the Auto Scaling Group's Activity History.

3. **Cost Awareness**: Highlighted potential costs of running additional instances beyond the AWS Free Tier during testing and ensured resource usage was optimized.