

## Auto Scaling in AWS

Auto Scaling in AWS is a service that allows you to automatically adjust the number of Amazon EC2 instances in your applications based on demand or a predefined schedule. The goal of Auto Scaling is to ensure that you have the right number of instances available to handle the load for your applications, while also optimizing costs.

Here are the key concepts and features of Auto Scaling:

### 1. Auto Scaling Groups (ASGs):

- Auto Scaling uses groups of instances called Auto Scaling Groups. You define the minimum and maximum number of instances in a group, and Auto Scaling automatically adjusts the number of instances within these bounds.

### 2. Launch Configurations:

- A Launch Configuration is a template that Auto Scaling uses to launch EC2 instances in an Auto Scaling Group. It specifies instance details such as the AMI (Amazon Machine Image), instance type, key pair, security groups, and block device mappings.

### 3. Scaling Policies:

- Scaling policies define the conditions under which Auto Scaling should automatically add or remove instances from an Auto Scaling Group. There are two types of scaling policies: **Simple Scaling Policies** and **Step Scaling Policies**.

### 4. Scaling Adjustments:

- When creating scaling policies, you specify scaling adjustments, which determine how many instances to add or remove based on the defined conditions.

### 5. Cool down Period:

- A cool down period is a configurable time period during which Auto Scaling prevents the group from launching or terminating additional instances. This helps prevent the group from launching or terminating more instances before the effects of the previous activity are visible.

### 6. Dynamic Scaling:

- Auto Scaling can automatically adjust the number of instances based on metrics like CPU utilization, network traffic, or custom Cloud Watch alarms. This ensures that your application can handle varying levels of demand.

### 7. **Scheduled Scaling:**

- You can also set up scheduled scaling to automatically adjust the number of instances based on a defined schedule. For example, you might scale up during business hours and scale down during non-business hours.

### 8. **Integration with Other AWS Services:**

- Auto Scaling integrates with other AWS services, such as Elastic Load Balancing and Cloud Watch, to create a comprehensive solution for managing the performance and availability of your applications.

### 9. **Auto Scaling Plans:**

- Auto Scaling Plans are a feature that allows you to optimize costs and maintain performance automatically. Auto Scaling evaluates historical data to create predictive scaling plans.

Using Auto Scaling, you can ensure that your application scales seamlessly based on demand, improving availability and reliability, while also controlling costs by scaling down during periods of lower demand.

## **Creating Auto scaling group**

### 1. **Sign in to the AWS Management Console:**

- Open the AWS Management Console in your web browser.
- Sign in with your AWS account credentials.

### 2. **Navigate to the EC2 Auto Scaling Console:**

- In the AWS Management Console, navigate to the "EC2 Auto Scaling" service.

### 3. **Click on "Auto Scaling Groups" in the left navigation pane:**

- Select "Auto Scaling Groups" from the left navigation pane.

### 4. **Click "Create Auto Scaling Group":**

- Click the "Create Auto Scaling Group" button.

### 5. **Choose Launch Template or Configuration:**

- Select whether you want to use a launch template or a launch configuration. A launch template is a more modern and flexible option.

**6. Select Launch Template or Configuration:**

- If using a launch template, choose your template from the list. If using a launch configuration, choose your configuration.

**7. Configure Auto Scaling Group Details:**

- Provide details for your Auto Scaling group, including the group name, network settings, and initial capacity.

**8. Configure Scaling Policies:**

- Set up scaling policies to define how the Auto Scaling group should respond to changes in demand. This includes specifying scaling adjustments based on Cloud Watch alarms.

**9. Configure Instance Protection (Optional):**

- Optionally, you can configure instance protection to prevent specific instances from being terminated during scale-in events.

**10. Configure Notifications (Optional):**

- Optionally, you can configure Amazon SNS notifications to be sent when scaling events occur.

**11. Configure Tags (Optional):**

- Optionally, you can add tags to your Auto Scaling group for better organization and resource management.

**12. Review Configuration:**

- Review the configuration settings to ensure they are correct.

**13. Create Auto Scaling Group:**

- Click the "Create Auto Scaling group" button to create the group.

The Auto Scaling group will now launch and manage instances based on the configured policies and settings. Instances will be automatically added or removed based on the conditions you specified, providing automatic scaling to meet demand.

Please note that the exact steps and options may vary based on your specific use case and whether you are using launch templates or launch configurations. Always refer to the AWS documentation for the most up-to-date and detailed instructions.

Group details

Edit

Auto Scaling group name ramaniasg	Desired capacity 10	Desired capacity type Units (number of instances)	Amazon Resource Name (ARN) arn:aws:autoscaling:us-east-1:937351429912:autoScalingGroup:b6d8c4a0-ad14-43c1-b1b2-33b191c6a07c:autoScalingGroupName/ramaniasg
Date created Thu Nov 16 2023 07:46:10 GMT+0530 (India Standard Time)	Minimum capacity 1	Status -	
	Maximum capacity 10		

Network

Edit

Availability Zones us-east-1a, us-east-1b, us-east-1c, us-east-1d, us-east-1e, us-east-1f	Subnet ID subnet-0b85da2b3f1e4341e, subnet-01b1492643313346a, subnet-09ecdc70fae3cb9d7, subnet-0f0dd60e230172d66, subnet-0bf3625245db6b38a, subnet-08bb9b717768ee66e
--	---

Instance type requirements

Edit

Your Auto Scaling group adheres to the launch template for purchase option and instance type.

Load balancing

Edit

Load balancer target groups -	Classic Load Balancers -
----------------------------------	-----------------------------

VPC Lattice integration options

Edit

VPC Lattice target groups -
--------------------------------

Health checks

Edit

Health check type EC2	Health check grace period 30
--------------------------	---------------------------------

## Command to increase CPU Utilization

**yes > /dev/null &**

**command to kill the process: kill -9 process-id**

## Amazon Elastic File System (Amazon EFS)

Amazon Elastic File System (Amazon EFS) is a fully managed file storage service provided by Amazon Web Services (AWS). It is designed to be highly scalable, elastic, and compatible with a variety of file-based workloads. Amazon EFS is particularly useful for scenarios where multiple EC2 instances need shared access to a common file system.

Key features of Amazon EFS include:

### 1. Fully Managed:

- Amazon EFS is a fully managed service, meaning AWS takes care of the underlying infrastructure, durability, and availability. You don't need to worry about server provisioning or maintenance.

### 2. Elastic Scalability:

- EFS can automatically scale its capacity as your storage needs grow. You can easily add and remove files without the need for manual intervention or disruption to your applications.

### 3. Highly Available and Durable:

- EFS is designed for high availability and durability. It stores data across multiple Availability Zones within a region, making it resilient to failures in a single data center.

### 4. Shared File System:

- Multiple EC2 instances can concurrently access the same Amazon EFS file system, making it suitable for scenarios such as content management systems, web serving, and development environments where shared access to files is essential.

### 5. Network File System (NFS) Protocol:

- Amazon EFS uses the Network File System (NFS) protocol, which is a standard file system protocol for distributed file systems. This makes it compatible with a wide range of applications and tools that support NFS.

### 6. Performance Modes:

- EFS supports two performance modes:
  - **General Purpose:** Suitable for a wide variety of workloads.

- **Max I/O:** Optimized for high-performance, latency-sensitive workloads.

#### 7. **Lifecycle Management:**

- Amazon EFS supports lifecycle management policies that automatically move files to lower-cost storage classes, helping to optimize costs.

#### 8. **Integration with AWS Services:**

- EFS can be easily integrated with other AWS services, such as Amazon EC2, AWS Lambda, Amazon ECS, and more.

To use Amazon EFS, you create a file system and then mount it on your EC2 instances using the NFS protocol. Once mounted, multiple instances can read and write to the file system simultaneously, providing a shared storage solution for your applications.

### **Creating EFS in AWS**

#### 1. **Sign in to the AWS Management Console:**

- Open the AWS Management Console in your web browser.
- Sign in with your AWS account credentials.

#### 2. **Navigate to the Amazon EFS Console:**

- In the AWS Management Console, navigate to the "Amazon EFS" service.

#### 3. **Click on "Create file system":**

- Click the "Create file system" button to start the process of creating a new Amazon EFS file system.

#### 4. **Configure File System Settings:**

- In the "Create file system" wizard, you'll need to configure various settings, including:
  - **Name:** Give your file system a descriptive name.
  - **VPC:** Choose the Virtual Private Cloud (VPC) in which you want to create the file system.
  - **Lifecycle Management:** Choose the storage classes and configure lifecycle management policies if needed.

**5. Configure Optional Settings (Optional):**

- You can configure optional settings such as enabling encryption, setting up tags for better organization, and configuring performance mode.

**6. Configure Network Settings:**

- Choose the security groups and subnets for your file system. Ensure that your EC2 instances have the necessary network access to the file system.

**7. Review Configuration:**

- Review the configuration details to ensure they are correct.

**8. Click on "Create file system":**

- Click the "Create file system" button to create the Amazon EFS file system.

**9. Wait for the File System to be Available:**

- It may take a few minutes for the file system to be created and become available.

**10. Mounting EFS to servers 1 and 2 :**

- **`sudo yum install -y amazon-efs-utils -y`**
- **`sudo mkdir efs`**
- **`sudo mount -t efs -o tls fs-097d6a8288a8afd46:/ efs`**

**11. Access the File System:**

- Once the file system is available, you can obtain the DNS name (Endpoint) of the file system from the EFS console. Mount this DNS name on your EC2 instances using the NFS protocol.

Amazon EFS is now ready to use, and you can start storing and accessing files on it. Ensure that your EC2 instances are configured with the necessary permissions and network access to connect to the EFS file system.

### File system

Field	Value	Is editable?
Name	efs	Yes
Performance mode	General Purpose	No
Throughput mode	Elastic	Yes
Encrypted	No	No
KMS Key ID	-	No
Lifecycle management	Transition into IA: 30 day(s) since last access Transition out of IA: None	Yes
Automatic backups	No	Yes
VPC ID	vpc-019a778094c924aad (default)	Yes
Availability Zone	Standard	No

### Step 2: Network access

[Edit](#)

#### Mount targets

Availability zone	Subnet	IP address	Security groups
us-east-1a	subnet-09ecdc70fae3cb9d7	-	sg-0a694c07aad8f20d0, sg-0df09f8f39559ceab
us-east-1b	subnet-0f0dd60e230172d66	-	sg-0a694c07aad8f20d0, sg-0df09f8f39559ceab
us-east-1c	subnet-0b85da2b3f1e4341e	-	sg-0a694c07aad8f20d0, sg-03e2533c6c8077516
us-east-1d	subnet-08bb9b717768ee66e	-	sg-0a694c07aad8f20d0, sg-03e2533c6c8077516
us-east-1f	subnet-01b1492643313346a	-	sg-0a694c07aad8f20d0, sg-03e2533c6c8077516