Auto Scaling Group

This guide provides a step-by-step walkthrough for setting up an Auto Scaling group integrated with an Application Load Balancer (ALB) using **Launch Templates**.

Prerequisites

Before you begin, ensure you have the following:

- **Virtual Private Cloud (VPC)**: A VPC with at least one public subnet in each Availability Zone where your instances will be located.
- **Security Groups**: Security groups that allow necessary inbound and outbound traffic.
- **IAM Role (Optional)**: An IAM role granting your instances access to required AWS services.
- **Amazon Machine Image (AMI)**: An AMI with the required software and configurations.

Step 1: Create a Launch Template

A **Launch Template** defines the configuration for instances in your Auto Scaling Group.

1. **Open the Amazon EC2 Console**
    - Navigate to **Launch Templates** under **Instances**.
2. **Create a New Launch Template**
    - Click **Create launch template**.
    - **Launch template name**: Enter a descriptive name (e.g., `my-launch-template`).
    - **Template version description**: (Optional) Provide a description.
    - **AMI ID**: Enter the ID of your desired AMI.
    - **Instance type**: Choose an instance type (e.g., `t3.micro`).
    - **Key pair**: Select an existing key pair or create a new one.
    - **Networking settings**: Select a security group that allows traffic from the load balancer.
    - **Storage (volumes)**: Configure the root and additional EBS volumes if needed.
    - **Advanced settings**: Add any user data scripts or IAM roles.
3. **Create the Launch Template**
    - Review your settings and click **Create launch template**.

Step 2: Create an Application Load Balancer (ALB)

An ALB distributes incoming traffic across multiple EC2 instances.

1. **Navigate to the Load Balancers page** in the EC2 Console.
2. **Click Create Load Balancer**.
3. **Select Application Load Balancer** and click **Create**.
4. **Basic Configuration**:
    - **Name**: `my-app-load-balancer`
    - **Scheme**: Select **Internet-facing**.
    - **Listeners**: Ensure an HTTP listener on port 80.
    - **VPC and Subnets**: Select at least two public subnets in different Availability Zones.
5. **Security Groups**:
    - Assign a security group allowing inbound traffic on port 80.
6. **Target Group Configuration**:
    - **Target group name**: `my-target-group`
    - **Target type**: Select **Instance**.
    - **Protocol**: HTTP (port 80)
    - **Health check settings**:
        - **Protocol**: HTTP
        - **Path**: `/`
        - **Healthy threshold**: 3
        - **Unhealthy threshold**: 2
        - **Interval**: 30 seconds
        - **Timeout**: 5 seconds
7. **Review and Create the ALB**.

Step 3: Create an Auto Scaling Group

An Auto Scaling Group automatically adjusts the number of EC2 instances based on demand.

1. **Navigate to Auto Scaling Groups** in the EC2 Console.
2. **Click Create Auto Scaling group**.
3. **Basic Configuration**:
    - **Auto Scaling group name**: `my-auto-scaling-group`
    - **Launch template**: Select the launch template created earlier.
4. **Network Configuration**:
    - **VPC**: Select the VPC containing your instances and ALB.
    - **Subnets**: Select at least two subnets in different Availability Zones.
5. **Attach Load Balancer**:

- Select **Attach to an existing load balancer**.
- Choose the previously created target group (`my-target-group`).

6. **Configure Health Checks**:
   - **Health check type**: Select **ELB**.
   - **Health check grace period**: 300 seconds.
7. **Configure Desired Capacity and Scaling Policies**:
   - **Desired capacity**: 2 (adjust as needed)
   - **Minimum capacity**: 1
   - **Maximum capacity**: 4
   - **Scaling policies**: Choose a policy based on CPU utilization or custom metrics.

Step 4: Configure Target Tracking Scaling Policy

1. **Navigate to Auto Scaling Groups** in the EC2 Console.
2. Select your **Auto Scaling Group** and go to the **Automatic scaling** tab.
3. Click **Create a scaling policy**.
4. **Choose Scaling Policy Type**:
   - Select **Target tracking scaling policy**.
5. **Configure the Scaling Policy**:
   - **Metric type**: Choose `Average CPU utilization` .
   - **Target value**: Set a desired percentage (e.g., `50%`).
   - **Instance warm-up time**: Enter a value in seconds (e.g., `300`).
   - Enable **Scale-in protection** if needed.
6. Click **Create** to apply the policy.

Step 5: Modify Instance Type and Perform Instance Refresh

1. **Update Launch Template**:
   - Navigate to **Launch Templates** in the EC2 console.
   - Select your **Launch Template** and click **Create new version**.
   - Change the **Instance type** (e.g., `t3.small` to `t3.medium`).
   - Click **Create template version**.
2. **Update Auto Scaling Group to Use New Launch Template Version**:
   - Navigate to **Auto Scaling Groups**.
   - Select your **Auto Scaling Group**.
   - Click **Edit** and update the **Launch template version**.

    - Save changes.
3. **Perform Instance Refresh**:
    - Go to the **Instance refresh** tab under your Auto Scaling Group.
    - Click **Start instance refresh**.
    - Select the desired settings and click **Start**.
    - Monitor the refresh process under the **Activity** tab.

Step 6: Verification

1. **Check ALB Status**:
    - Navigate to **Load Balancers** and ensure the ALB is **active**.
2. **Test the ALB**:
    - Copy the ALB's **DNS name** and open it in a browser.
    - Refresh multiple times to verify requests are distributed across instances.
3. **Check Auto Scaling Activity**:
    - Navigate to **Auto Scaling Groups** → **Activity History**.
    - Verify instances are launched based on scaling policies.

Troubleshooting

- If instances are not registering as healthy, check **target group health checks**.
- Ensure security groups allow traffic from ALB to EC2 instances.