**Task: Set up a CloudWatch Alarm for EC2 CPU ≥ 50% with Auto Scaling Group**

**Name:** Abhishek Hengade

**Batch:** B-56

This document outlines the procedure for creating a CloudWatch alarm that monitors the average CPU utilization of an EC2 Auto Scaling Group (ASG). When the average CPU utilization is greater than or equal to 50% for a sustained period, the alarm will trigger a scaling policy to launch a new EC2 instance.

**Prerequisites**

Before you begin, ensure you have the following:

1. **An active AWS account**.
2. **An existing EC2 Auto Scaling Group (ASG)**. You should have an ASG configured with a launch template or launch configuration. If you don't have one, you'll need to create one first.
3. **Necessary IAM Permissions**. Your AWS user or role must have permissions for CloudWatch (cloudwatch:\*) and Auto Scaling (autoscaling:\*). The AdministratorAccess policy is sufficient, but for production, it's best to use a more restrictive, least-privilege policy.

**Step 1: Create the Scaling Policies for the ASG**

First, you need to define what the Auto Scaling Group should do when the alarm is triggered (scale-out) and when the CPU utilization returns to normal (scale-in).

**1.1 Create the Scale-Out Policy (Add Instances)**

This policy will be triggered by the high CPU alarm.

1. Navigate to the **EC2** console in AWS.
2. On the left-hand navigation pane, scroll down and select **Auto Scaling Groups**.
3. Select the checkbox next to your target ASG.
4. In the details pane below, click on the **Automatic scaling** tab.
5. In the **Dynamic scaling policies** section, click **Create dynamic scaling policy**.
   * **Policy type**: Select **Simple scaling**.
   * **Scaling policy name**: Give it a descriptive name, like add-one-instance-on-high-cpu.
   * **CloudWatch alarm (optional)**: Leave this blank for now. We will create the alarm separately and link it to this policy.
   * **Take the action**: Select **Add**.
   * **instances**: Enter 1. This means one instance will be added when the policy is executed.
   * **Instance warmup (seconds)**: Set a cooldown period, for example, 300. This prevents the ASG from launching too many instances too quickly. This means after a scaling activity, the ASG will wait 300 seconds before another one can begin.
6. Click **Create**.

**1.2 (Optional but Recommended) Create the Scale-In Policy (Remove Instances)**

It's good practice to create a corresponding policy to remove instances when CPU is low to save costs.

1. In the same **Dynamic scaling policies** section, click **Create dynamic scaling policy** again.
2. Configure the policy:
   * **Policy type**: **Simple scaling**.
   * **Scaling policy name**: remove-one-instance-on-low-cpu.
   * **CloudWatch alarm (optional)**: We will create a separate alarm for this later.
   * **Take the action**: Select **Remove**.
   * **instances**: Enter 1.
   * **Instance warmup (seconds)**: 300.
3. Click **Create**.

**Step 2: Create the CloudWatch Alarm for High CPU**

Now, we'll create the alarm that watches the CPU metric and triggers the scale-out policy we just created.

1. Navigate to the **CloudWatch** console.
2. In the left navigation pane, click on **Alarms**, and then **All alarms**.
3. Click the **Create alarm** button.
4. Click **Select metric**.
   * Under **All metrics**, choose **EC2**.
   * Next, select **By Auto Scaling Group**.
   * Find your Auto Scaling Group and select the metric **CPUUtilization**.
5. Click **Select metric**.

**2.1 Configure Alarm Conditions**

Now you'll configure the specifics of the alarm.

1. **Metric**:
   * **Statistic**: Leave this as **Average**. This calculates the average CPU utilization across all instances in the ASG.
   * **Period**: Select **5 minutes**. This means CloudWatch will aggregate the data points into 5-minute intervals.
2. **Conditions**:
   * **Threshold type**: **Static**.
   * **Whenever CPUUtilization is...**: Select **Greater/Equal**.
   * **than...**: Enter 50. This is our 50% CPU threshold.
3. **Additional configuration**:
   * **Datapoints to alarm**: Set this to 2 out of 3. This helps prevent false alarms from temporary spikes. The alarm will only go into the ALARM state if the CPU has been >= 50% for two out of the last three 5-minute periods (i.e., for at least 10 minutes within a 15-minute window).
   * **Missing data treatment**: Leave as Treat missing data as missing.
4. Click **Next**.

**2.2 Configure the Alarm Action**

This is where you link the alarm to the ASG scaling policy.

1. In the **Configure actions** screen, select **In alarm** for the alarm state trigger.
2. Select **Add notification**.
   * **Select an SNS topic**: You can optionally configure an SNS topic to get an email or SMS notification. For this guide, we will focus on the scaling action.
3. Select **Add Auto Scaling action**.
4. Choose your ASG from the dropdown list.
5. Select the scale-out policy you created in Step 1.1 (e.g., add-one-instance-on-high-cpu).
6. Click **Next**.

**2.3 Name and Create the Alarm**

1. **Alarm name**: Give it a clear name, like High-CPU-Utilization-Alarm-for-My-ASG.
2. **Alarm description (optional)**: Describe what the alarm does, e.g., "Triggers ASG to scale out when average CPU is >= 50%."
3. Review all the settings and click **Create alarm**.

**Step 3: Verification and Testing**

Once the alarm is created, it will be in the INSUFFICIENT\_DATA state for a few minutes as it collects the initial metric data. It will then transition to OK (green).

To test the setup:

1. **Generate Load**: Connect to the instance(s) in your ASG and run a CPU stress tool. For Linux, you can use the stress utility.
   * Install it: sudo amazon-linux-extras install epel -y && sudo yum install stress -y
   * Run it to generate load: stress --cpu 1 --timeout 300s (This will max out 1 CPU core for 300 seconds).
2. **Monitor CloudWatch**: Watch your alarm in the CloudWatch console. After the configured period (e.g., 10-15 minutes), the alarm state should change from OK to In alarm (red).
3. **Check the ASG**: Go to the EC2 Auto Scaling Groups console.
   * Check the **Activity** tab for your ASG. You should see an event indicating that a new instance is being launched because of the CloudWatch alarm.
   * Check the **Instance management** tab. You should see the desired number of instances increase by one.

After your stress test ends, the average CPU utilization will drop. If you created a corresponding scale-in alarm, it will eventually trigger, and the ASG will terminate the extra instance.

