# Auto Scaling EC2 Instances Based on Custom CloudWatch Metrics

## 🎯 Objective

Automatically scale EC2 instances based on the number of users logged into a custom application by:  
- Capturing metrics using AWS CloudWatch  
- Using Auto Scaling Groups (ASG)  
- Automating EC2 launch and termination based on load

## 🧱 Prerequisites

- AWS account  
- Basic understanding of EC2, IAM, CloudWatch, Auto Scaling  
- IAM role with permissions: EC2, CloudWatch, Auto Scaling

## ✅ Step 1: Launch a Base EC2 Instance

1. Go to EC2 Dashboard → Launch Instance  
2. Select Amazon Linux 2 AMI  
3. Choose t2.micro instance type  
4. Create/select an existing Key Pair  
5. In Advanced Details → User Data, paste the following:

#!/bin/bash  
yum update -y  
yum install -y aws-cli  
echo "Installing custom application"  
# (simulate app installation or add actual install commands)  
while true; do  
 users=$(( ( RANDOM % 100 ) + 1 ))  
 aws cloudwatch put-metric-data --metric-name ActiveUsers --namespace "CustomApp" --unit Count --value $users --region us-east-1  
 sleep 60  
done

6. Add a Security Group allowing:  
- Port 22 (SSH)  
- Port 80 (HTTP) if web server is used  
7. Click Launch Instance

## 📷 Step 2: Create an AMI (Amazon Machine Image)

1. Go to EC2 → Instances  
2. Select your instance → Actions → Image → Create Image  
3. Name it: MyApp-AMI  
4. Click Create Image  
5. Wait until image status is available

## ⚙️ Step 3: Create a Launch Template

1. Go to EC2 → Launch Templates → Create Launch Template  
2. Name: AppLaunchTemplate  
3. AMI ID: Use the MyApp-AMI  
4. Instance Type: t2.micro  
5. Key Pair: Same one used before  
6. Security Group: Same one from instance  
7. IAM Role: Attach the role created in Step A below  
8. Click Create Launch Template

## 🛠️ Step A: Create and Attach an IAM Role

Create IAM Role:  
1. Go to IAM Console → Roles → Create Role  
2. Choose Trusted Entity: AWS service → EC2  
3. Permissions:  
- CloudWatchAgentServerPolicy  
- (Optional) AmazonEC2RoleforSSM  
4. Name it: EC2CloudWatchPushRole  
5. Click Create Role  
Attach IAM Role in Launch Template:  
- In Launch Template → Advanced Details, select:  
- IAM Instance Profile: EC2CloudWatchPushRole

## 📈 Step 4: Create a CloudWatch Alarm for Custom Metric

1. Go to CloudWatch → Alarms → Create Alarm  
2. Click Select Metric  
3. Browse → Custom Namespaces → CustomApp → ActiveUsers  
4. Select the metric → Click Select metric  
5. Configure Alarm:  
- Statistic: Average  
- Period: 1 minute  
- Threshold: Greater than 70  
- Datapoints: 1 out of 1  
- Name: ActiveUsersHighAlarm  
6. Configure Actions:  
- Option A: Send SNS notification  
- Option B: Skip and use in Auto Scaling Policy  
7. Click Create Alarm

## 🔄 Step 5: Create Auto Scaling Group (ASG)

1. Go to EC2 → Auto Scaling Groups → Create Auto Scaling Group  
2. Name: MyApp-ASG  
3. Select Launch Template: AppLaunchTemplate  
4. Network: Select your VPC and subnets  
5. Desired Capacity: 1  
6. Min: 1, Max: 3  
7. Scaling Policies:  
- Target Tracking Policy  
 - Metric Type: Custom Metric  
 - Namespace: CustomApp  
 - Metric Name: ActiveUsers  
 - Target Value: 50  
 - Instance warm-up: 60s  
- OR Step Scaling Policy using ActiveUsersHighAlarm  
8. Click Create Auto Scaling Group

## 🧪 Step 6: Test the Setup

- Check if ActiveUsers metric is visible in CloudWatch  
- Watch EC2 → Auto Scaling → Activity for scale-out/in  
- View alarm state: OK (below threshold), ALARM (above 70)

## 🧹 Step 7: Clean Up (Optional)

1. Delete:  
- EC2 instances  
- Auto Scaling Group  
- Launch Template  
- CloudWatch alarms and metrics  
2. Remove any SNS subscriptions and IAM roles if not needed

## 📝 Summary

|  |  |
| --- | --- |
| Component | Description |
| EC2 | Runs the application and sends metrics |
| CloudWatch | Captures ActiveUsers metric |
| IAM Role | Allows EC2 to push metrics |
| AMI | Contains pre-installed app for new EC2s |
| Launch Template | Used by ASG to spin up EC2 instances |
| Auto Scaling | Automatically adjusts EC2 count based on usage |
| Alarm | Triggers scaling actions based on user metric |