Lab 05: EC2 & Auto Scaling

Lab Overview

Launch EC2 instances using a Launch Template with a bootstrap User Data script, deploy them behind an Application Load Balancer across two Availability Zones, and configure an Auto Scaling Group with a Target Tracking policy. Verify load balancing by watching instance IDs change on each page refresh.

Service	Purpose	Free Tier
Amazon EC2	Virtual machines running in the AWS cloud	750 hrs/mo t2.micro
Application Load Balancer (ALB)	Distributes HTTP traffic across multiple EC2 instances	750 hrs/mo
EC2 Auto Scaling	Launches or terminates instances based on demand or health	Free (pay for instances)
VPC / Security Groups	Network isolation and virtual firewall rules for instances	Free

■ **WARNING:** EC2 and Load Balancers incur charges if left running. Delete all resources immediately after finishing.

Key Concepts

Concept	Definition	Exam Weight
EC2 Instance	A virtual server. Choose OS, CPU, memory, and storage.	Very High
AMI	Amazon Machine Image — OS template used to launch an instance.	High
Security Group	Virtual firewall with stateful inbound and outbound rules.	Very High
Launch Template	Saved instance configuration used by Auto Scaling Groups.	High
User Data	Bootstrap script that runs once automatically at instance launch.	High
ASG	Maintains desired capacity and replaces unhealthy instances.	Very High
ALB	Distributes traffic across instances in multiple Availability Zones.	High
Target Tracking	Scaling policy that maintains a target metric value (e.g. 50% CPU).	High

1 Amazon VPC — Security Group
Create a Security Group

1. EC2 → left sidebar → Security Groups → Create security group
2. Name: lab-web-sg → default VPC
3. Inbound rule 1: Type HTTP, Source Anywhere-IPv4 (0.0.0.0/0)
4. Inbound rule 2: Type SSH, Source My IP
5. Click Create security group

■ **NOTE:** HTTP allows internet traffic on port 80. SSH is restricted to your IP only — Principle of Least Privilege.

2 Amazon EC2 Create a Launch Template

1. EC2 → Launch Templates → Create launch template
2. Name: lab-web-template
3. AMI: Amazon Linux 2023 (Quick Start)
4. Instance type: t2.micro
5. Key pair: create or select an existing key pair
6. Network settings: select existing security group → lab-web-sg
7. Advanced details → User data → paste the script below:

```
#!/bin/bash
yum update -y && yum install -y httpd
systemctl start httpd && systemctl enable httpd
TOKEN=$(curl -s -X PUT 'http://169.254.169.254/latest/api/token' \
  -H 'X-aws-ec2-metadata-token-ttl-seconds: 21600')
INSTANCE_ID=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
  http://169.254.169.254/latest/meta-data/instance-id)
AZ=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
  http://169.254.169.254/latest/meta-data/placement/availability-zone)
cat > /var/www/html/index.html << HTML
<html><body style='font-family:Arial;text-align:center;padding:40px;'>
<h1>Cloud Practice Labs</h1>
<p><strong>Instance ID:</strong> $INSTANCE_ID</p>
<p><strong>Availability Zone:</strong> $AZ</p>
<p>Refresh to see the load balancer routing to different instances.</p>
</body></html>
HTML
```

8. Click Create launch template

✓ **TIP:** The User Data script installs Apache and creates a page showing the Instance ID and AZ. Different IDs on each refresh prove load balancing is working.

3 EC2 Auto Scaling Create an Auto Scaling Group with Load Balancer

1. EC2 → Auto Scaling Groups → Create Auto Scaling group
2. Name: lab-asg → Launch template: lab-web-template → Next
3. Select default VPC → choose at least 2 subnets in different AZs → Next
4. Load balancing → Attach to a new load balancer
5. Type: Application Load Balancer → Name: lab-alb → Scheme: Internet-facing
6. Create a target group: lab-tg → Next
7. Desired: 2 → Min: 1 → Max: 4 → Next through remaining steps → Create

■ **NOTE:** Desired=2, Min=1, Max=4. The ASG always runs 2 instances and automatically replaces any that become unhealthy.

4 EC2 Auto Scaling — Policy Create a Target Tracking Scaling Policy

1. Click lab-asg → Automatic scaling tab → Create dynamic scaling policy
2. Policy type: Target tracking scaling

3. Policy name: lab-cpu-policy
4. Metric: Average CPU utilization → Target value: 50
5. Instance warmup: 300 seconds → Create

✓ **TIP:** When average CPU rises above 50%, the ASG adds instances. When it falls below, instances are removed — down to the minimum of 1.

5 Application Load Balancer Test Load Balancing

1. EC2 → Load Balancers → lab-alb → copy the DNS name
2. EC2 → Target Groups → lab-tg → Targets tab
3. Wait for both instances to show status: healthy (3–5 minutes)
4. Paste the ALB DNS name into your browser
5. Refresh the page — the Instance ID and AZ change, confirming load balancing

Verification Checklist

- ■ Security group lab-web-sg created with HTTP and SSH inbound rules
- ■ Launch template lab-web-template with Amazon Linux 2023 and user data script
- ■ Auto Scaling group lab-asg: Desired=2, Min=1, Max=4
- ■ Application Load Balancer lab-alb created as Internet-facing
- ■ Target group lab-tg: both instances show healthy status
- ■ ALB DNS name loads the web page in a browser
- ■ Refreshing shows different Instance IDs (load balancing confirmed)
- ■ Target tracking policy lab-cpu-policy created at 50% CPU

What You Learned

- • Amazon EC2 — instance types, AMIs, and the Free Tier t2.micro
- • Security Groups — virtual firewalls with stateful inbound and outbound rules
- • Launch Templates — reusable instance configurations with User Data scripts
- • Auto Scaling Groups — self-healing, elastic capacity management
- • Application Load Balancer — traffic distribution across multiple Availability Zones
- • Target Tracking Policies — automatic scaling based on CPU utilization metrics
- • High Availability — multi-AZ deployments eliminate single points of failure

Lab Cleanup

✗ **IMPORTANT:** EC2 and Load Balancers incur ongoing charges. Delete all resources immediately.

#	Resource	How to Delete
1	Auto Scaling Group	EC2 → Auto Scaling Groups → lab-asg → Delete (terminates all instances)
2	Load Balancer	EC2 → Load Balancers → lab-alb → Actions → Delete

3	Target Group	EC2 → Target Groups → lab-tg → Actions → Delete
4	Launch Template	EC2 → Launch Templates → lab-web-template → Actions → Delete
5	Security Group	EC2 → Security Groups → lab-web-sg → Actions → Delete
6	Key Pair	EC2 → Key Pairs → lab-keypair → Delete (also delete the .pem file locally)