This study guide demonstrates the lesson from *Elasticity with EC2 Auto Scaling Groups.*
My full AWS Architect Associate course can be found here:
https://www.udemy.com/course/ultimateaws/?referralCode=7ED214B795C444141361

Understanding Auto Scaling in AWS Study Guide

Auto Scaling is a critical feature in AWS that dynamically adjusts the number of EC2 instances in response to changing demand. This ensures that applications remain highly available and cost-efficient by scaling out when demand increases and scaling in during periods of low demand. In this lesson, we'll explore the fundamental components of Auto Scaling, including Launch Templates, Launch Configurations, and Auto Scaling Groups (ASGs), and how they work together.

What is Auto Scaling?

Auto Scaling automatically manages the number of EC2 instances in a defined group to match current demand. This elasticity allows workloads to scale efficiently, improving application availability while optimizing costs.

Key Features of Auto Scaling:

1. **Dynamic Scaling**: Automatically adjusts resources in response to predefined metrics or events (e.g., CPU usage, network traffic).
2. **Elasticity**: Scales out (adds instances) during high demand and scales in (removes instances) during low demand.
3. **Integration with Load Balancers**: Ensures new instances are automatically added to target groups and begin handling traffic immediately.
4. **Fault Tolerance**: Detects and replaces unhealthy instances.

Key Components of Auto Scaling

1. Launch Templates

A **Launch Template** is a blueprint for creating EC2 instances. It defines the configuration of the instances that the Auto Scaling Group will manage.

Key Attributes of a Launch Template:

- **AMI (Amazon Machine Image)**: Determines the operating system (e.g., Amazon Linux, Ubuntu, Windows) and pre-installed software.
- **Instance Type**: Specifies the hardware configuration (e.g., t2.micro, m5.large).
- **User Data Scripts**: Automates instance setup by running commands during boot (e.g., installing web servers).
- **Storage Configuration**: Defines EBS volume size and type.
- **Network Settings**: Specifies the VPC, subnets, and security groups.
- **Key Pair**: Enables SSH access to instances.

Launch Templates provide flexibility because they can be used to:

- Launch individual EC2 instances.
- Serve as a template for Auto Scaling Groups.

---

2. Launch Configurations

A **Launch Configuration** is similar to a Launch Template but is exclusively used with Auto Scaling Groups. Unlike Launch Templates:

- It **cannot** be used to launch individual EC2 instances.
- Once created, it **cannot be modified** (you must create a new one for changes).

3. Auto Scaling Groups (ASG)

An **Auto Scaling Group** is a logical grouping of EC2 instances managed by Auto Scaling. It uses the configuration defined in a Launch Template or Launch Configuration to scale resources dynamically.

Key Attributes of an ASG:

- **Minimum, Maximum, and Desired Capacity**:
  - **Minimum**: The least number of instances the ASG should maintain.
  - **Maximum**: The upper limit of instances the ASG can scale out to.
  - **Desired Capacity**: The number of instances the ASG tries to maintain under normal conditions.
- **Scaling Policies**:
  - **Dynamic Scaling**: Adds or removes instances based on performance metrics (e.g., CPU utilization).
  - **Scheduled Scaling**: Adjusts capacity at specific times (e.g., increasing capacity during business hours).
- **Health Checks**: Automatically replaces unhealthy instances.
- **Integration with Load Balancers**: Ensures traffic is distributed to instances efficiently.

How Auto Scaling Works

Step 1: Create a Launch Template

The Launch Template defines the EC2 instances' specifications:

1. Choose an **AMI** for the instance.
2. Specify the **instance type** (e.g., t2.micro).
3. Add **User Data** for automated setup:
    - Example: Install Apache on Amazon Linux:

```
#!/bin/bash
sudo yum update -y
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd
echo "Hello from Auto Scaling!" > /var/www/html/index.html
```

4. Configure **network settings**: Assign subnets and security groups.
5. Save the Launch Template.

Step 2: Configure the Auto Scaling Group

1. Define the ASG's **capacity settings**:
    - Minimum: 1
    - Desired: 2
    - Maximum: 5
2. Attach the ASG to a **Load Balancer Target Group**:
    - Ensures new instances are automatically registered and begin receiving traffic.
3. Set **scaling policies**:
    - Example: Add an instance if CPU utilization exceeds 70%.
    - Example: Remove an instance if CPU utilization drops below 30%.

Step 3: Monitor and Adjust

1. Auto Scaling launches instances when demand increases and terminates them when demand decreases.
2. The ASG continuously checks instance health and replaces any failing instances.

Elastic Scaling: Scale In and Scale Out

Scale Out (Add Instances)

- Triggered when demand increases, such as:
    - High CPU utilization.
    - Spikes in network traffic.
- Example: During a flash sale, Auto Scaling might launch 10 additional instances to handle the load.

Scale In (Remove Instances)

- Triggered during periods of low demand.
- Reduces operational costs by shutting down unnecessary instances.

Elasticity Analogy:

Think of Auto Scaling like a rubber band that stretches (scales out), or contracts (scales in) based on demand. It ensures you have just the right amount of resources at any given time.

Benefits of Auto Scaling

1. **Cost Efficiency**:
   - Scale out during peak demand to ensure availability.
   - Scale in during low demand to reduce costs.
2. **High Availability**:
   - Maintains sufficient capacity to meet demand at all times.
3. **Fault Tolerance**:
   - Automatically replaces unhealthy instances.
4. **Seamless Integration**:
   - Works with Elastic Load Balancers to distribute traffic efficiently.

Launch Template vs. Launch Configuration

Feature	Launch Template	Launch Configuration
Usage	Launch instances or ASGs	Only for ASGs
Modifiable	Yes	No
Advanced Features	Supports newer features	Limited capabilities
Reusability	Can be reused for multiple purposes	Single-purpose

---

Summary

In this lesson, you learned how Auto Scaling in AWS dynamically adjusts EC2 instance capacity based on demand. By creating a Launch Template as a blueprint and configuring an Auto Scaling Group, you can ensure your applications are highly available and cost-efficient. Auto Scaling's ability to elastically scale resources provides flexibility, fault tolerance, and operational efficiency, making it an essential tool in cloud computing.

See slides below:

Auto Scaling

BLUEPRINT

Launch Template or
Launch Configuration

"What should the EC2 instances look like? Blueprint!"

AMI
User Data
Instance Type (T3? C5?)
EBS volume?
Subnet, VPC

ASG

Auto Scaling Group

"How many instances should be created and when?

ELB?

Auto Scaling

- Right number EC2 instances to handle the workload
- Configurable minimum and maximum number of instances
- Auto Scaling expands or contracts based on demand
- This is called “Scaling In” or “Scaling Out”

Launch Template

- Provides a template for EC2 Instances
- Specifies the source AMI
- You can create boot scripts to customize each instance

Launch Configuration

- Basically the same as a Launch Template
- Launch Templates can be used to create individual EC2 instances or Auto Scaling Groups
- Launch Configuration can only be used with Auto Scaling

For more details see my full AWS Architect Associate course:
https://www.udemy.com/course/ultimateaws/?referralCode=7ED214B795C444141361