**AWS Auto Scaling Group (ASG) Setup with Interlinking**

1. **Create VPC** - A custom VPC is required to manage the networking of the infrastructure.
2. **Create Subnets**
   * 2 Public Subnets: Subnet 1 (1c) and Subnet 2 (1c) with an Internet Gateway (IG).
   * 2 Private Subnets: Subnet 3 (1c) and Subnet 4 (1a) for internal instances.
   * Ensure subnets are created across at least two Availability Zones (AZs) to ensure high availability.
3. **Create Public Server**
   * Deploy a sample application on the public instance.
   * Verify access using the public IP.
4. **Create AMI from Public Server**
   * This ensures all instances created from the Launch Template have the same OS and application setup.
5. **Create Launch Template**
   * Reference the AMI created in step 4.
   * Choose instance type (e.g., t2.micro), key name, and networking details.
   * Instances launched from this template will have the same configuration.
6. **Create Target Group (TG)**
   * This is required to route traffic to private instances.
   * Health check path: Default /var/www.html.
   * Select the private server created from the Launch Template.
7. **Create Load Balancer (LB)**
   * Use an Application Load Balancer (ALB) to distribute traffic.
   * Select the custom VPC and subnets:
     + Public Subnets: 1a (Subnet 5) and 1c (Subnet 1 or 2).
     + Private Subnets: 1a (Subnet 4) and 1c (Subnet 3).
   * Associate the Target Group with the Load Balancer.
8. **Create Auto Scaling Group (ASG)**
   * ASG references the Launch Template.
   * Assign networking details and Target Group.
   * Configure scaling policy:
     + Target Tracking Policy.
     + Metric Type: Average CPU Utilization.
     + Target Value: 90%.
   * Set desired instance count to 1 (ensuring one instance is always running).
   * Min and max instances scale based on load.

**Interlinking Summary:**

* **Launch Template** references **AMI** to ensure identical instances.
* **ASG** takes configuration from **Launch Template** and ensures high availability.
* **Load Balancer** routes traffic to instances via **Target Group**.
* **Target Group** ensures that only healthy **private instances** receive traffic.
* **Networking setup** ensures that only public subnets have Internet Gateway (IG) access, while private subnets are accessible via Load Balancer.

**Verification Steps:**

* Server lands in the correct **network** due to VPC & subnet setup.
* Server registers in the **Target Group** since the Load Balancer is correctly configured.
* Server runs the **same application** due to the AMI in the Launch Template.
* Server has the **same instance type and key** due to Launch Template settings.
* If access issues occur, ensure all **four interlinking steps** are correctly configured.

**Breakdown & Explanations**

1. **Custom VPC Creation**
   * A **VPC (Virtual Private Cloud)** is created to provide an isolated network environment for the application.
   * It allows full control over IP addressing, subnetting, and network configurations.
2. **Subnet Configuration**
   * **Public Subnets** (Subnet 1, 2, 5) → These have internet access via an **Internet Gateway (IG)**.
   * **Private Subnets** (Subnet 3, 4) → These are not associated with route tables that allow internet traffic.
3. **Private Server in Subnet 3**
   * Since **Subnet 3 is private** and does not have a route table associated with external access, the private server inside it **cannot be accessed directly**.
   * To allow access, we need a **Target Group (TG) and an Application Load Balancer (ALB)**.
4. **Why Use a Load Balancer?**
   * The **Application Load Balancer (ALB)** distributes traffic efficiently across multiple instances.
   * It allows access to the private server while maintaining its **private nature**.
5. **Why Create Another Public Subnet (Subnet 5)?**
   * The public subnets (Subnet 1 and 2) were in the same **Availability Zone (AZ)** (1c).
   * A **new public subnet (Subnet 5) in AZ 1a** ensures **high availability** and **redundancy**.
6. **Launch Template Usage**
   * The **Launch Template** standardizes the instance configuration (AMI, instance type, key pair, etc.).
   * It ensures that any server created through **Auto Scaling Group (ASG)** has the same configurations.
7. **Auto Scaling Group (ASG) & Target Group (TG) Interlinking**
   * **Auto Scaling Group (ASG)** manages instance scaling based on load.
   * The **Target Group (TG)** ensures new instances launched via ASG **automatically register** with the Load Balancer.
8. **Scaling Policy (Target Tracking)**
   * The **Target Tracking Policy** adjusts the number of instances **based on CPU utilization (90%)**.
   * The **desired capacity is set to 1**, meaning at least **one server is always running**.
9. **Step-by-Step Deployment Summary**
   * **VPC and Subnets** → Create public and private subnets.
   * **Public Server & Application Deployment** → Deploy a test application.
   * **AMI Creation** → Snapshot of the server to ensure consistency.
   * **Launch Template** → Pre-configured instance settings.
   * **Target Group** → Directs traffic to the correct instances.
   * **Application Load Balancer (ALB)** → Routes traffic based on TG.
   * **Auto Scaling Group (ASG)** → Automatically creates new servers as needed.