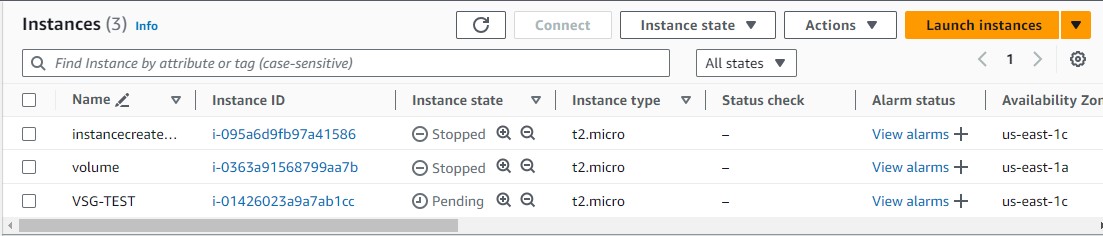
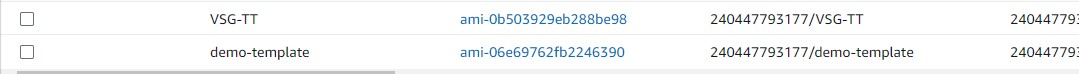
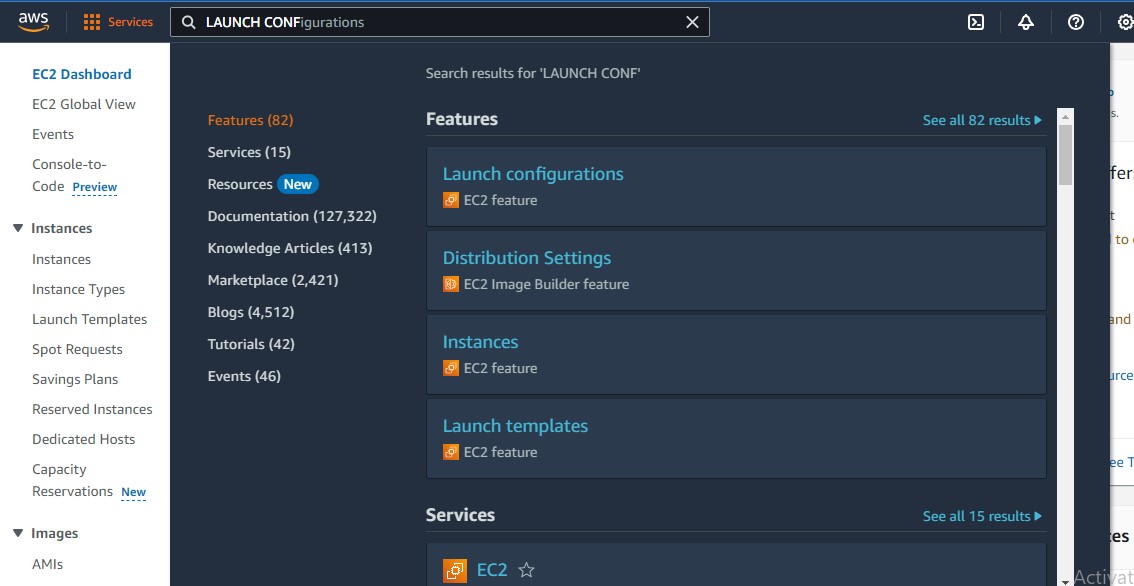
1. **Steps for creating Vertical scaling Groups…**
2. **Have to created new Ec2 instance..as VSGG-TEST**
3. **As shown in below image…**
4. **Next move on Launch configurations..**
5. **For that have to search in search bar option..**

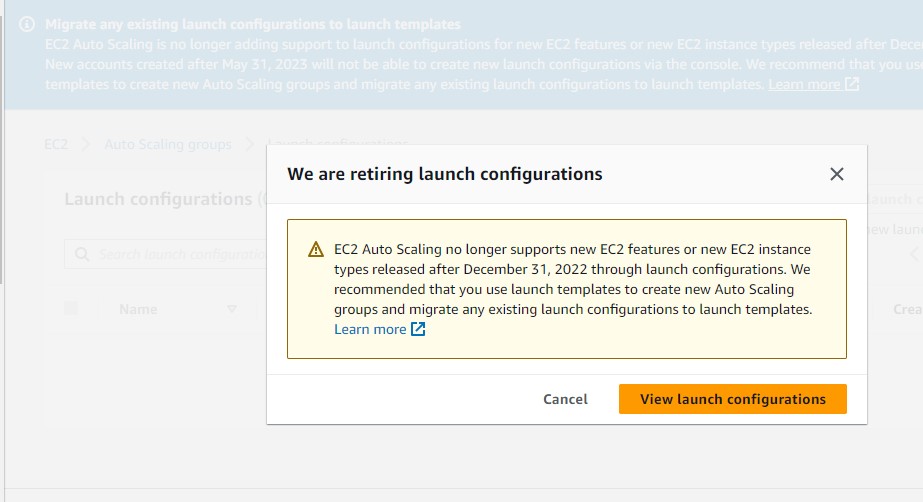


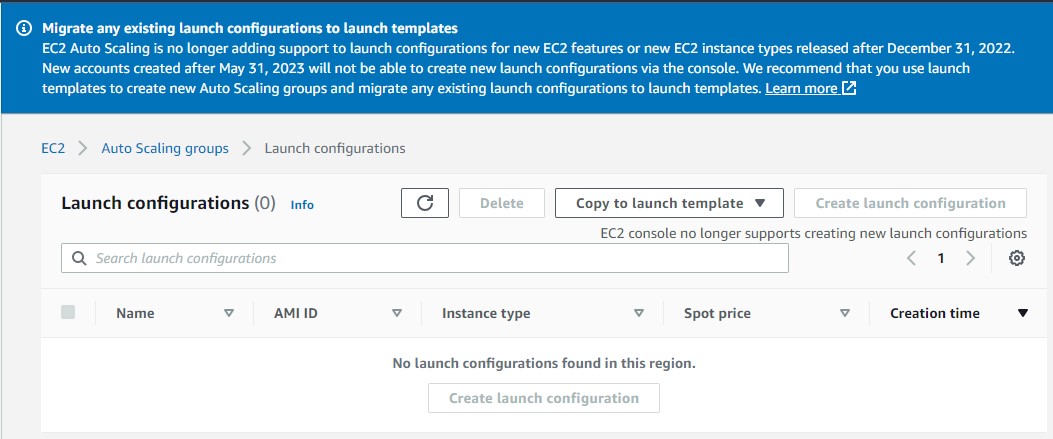


1. **Here the launch configuration interface.. as shown in below image…**



1. **Have to select launch configuration option after selecting here the popup display like this …**





**Vertical Scaling**

For the initial users up to 100, a single EC2 instance would be sufficient, e.g. t2.micro/t2.nano. The one instance would run the entire web stack, for example, web app, database, management, etc. The original architecture is fine until your traffic ramps up. Here you can scale vertically by increasing the capacity of your EC2 instance to address the growing demands of the application when the users grow up to 100. Vertical scaling means that you scale by adding more power (CPU, RAM) to an existing machine. AWS provides instances up to 488 GB of RAM or 128 virtual cores.

There are few challenges in basic architecture. First, we are using a single machine which means you don’t have a redundant server. Second, machine resides in a single AZ, which means your application health is bound to a single location.

To address the vertical scaling challenge, you start with decoupling your application tiers. Application tiers are likely to have different resource needs and those needs might grow at different rates. By separating the tiers, you can compose each tier using the most appropriate instance type based on different resource needs.

Now, try to design your application so it can function in a distributed fashion. For example, you should be able to handle a request using any web server and produce the same user experience. Store application state independently so that subsequent requests do not need to be handled by the same server. Once the servers are stateless, you can scale by adding more instances to a tier and load balance incoming requests across EC2 instances using Elastic Load Balancing (ELB).