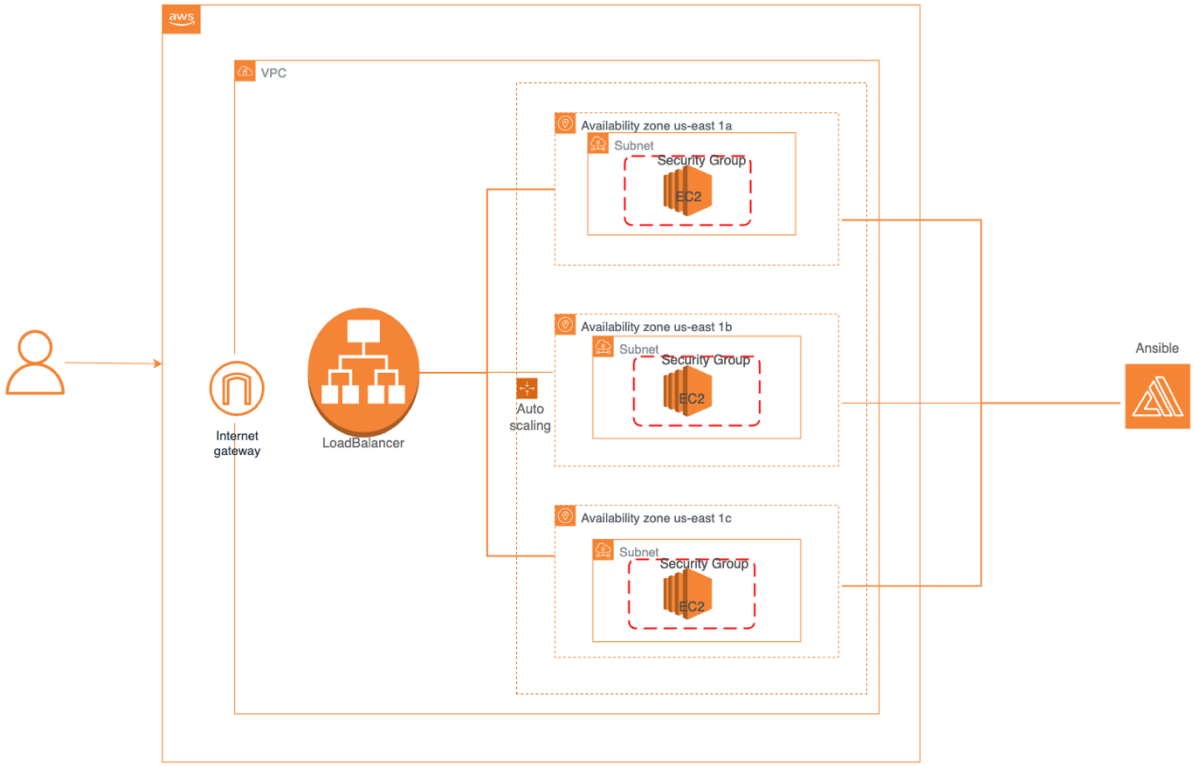
TASK 1: HOST A SIMPLE WEBPAGE ON AWS

conception phase



In this phase, I designed the architecture I planned to implement.

The journey is as follows

- **User:** This is the person, gadget that will request for resources on my architecture

- **AWS:** I am going to use AWS cloud which is the world most populous and widely use cloud infrastructure and going to use terraform to make the Infrastructure as Code (IaC) flexible and adaptable to other cloud infrastructure providers

- **Virtual Private Cloud (VPC):** Because I hold security so highly, I am going to have a private Cloud so that not all the have access to my AWS can manipulate the resources

- **Internet Gateway:** This gives internet access to the VPC which enables the VPC and the resources therein to be access by the outside world
- **LoadBalancer:** As the name indicates, they balance load based on the default or customized protocol defined among the subnets

- **Auto Scaling:** This is a feature that automatically adjusts the number of EC2 (Elastic Compute Cloud) instances in a group based on demand. This ensures that applications have the right amount of resources to handle traffic fluctuations, optimizing cost and performance.

- **Availability Zones**: To make my resources highly available against downtime, I choose three availability zones,

- **Public Subnets:** These are subnets where resources can communicate directly with the internet. Typically, you place resources like web servers in public subnets. Public subnets are associated with an internet gateway, which allows outbound and inbound internet traffic.

- **Security Group:** This is use to give a security to EC2, to define the rule and protocol that EC2 can communicate and can be communicated to

- **Elastic Cloud Computing (EC2):** This provides scalable computing capacity in the cloud. It allows users to run virtual servers, known as instances, on-demand.

- **Ansible:** To deploy my resources, Ansible is use which simplifies the management and configuration of computer systems, applications, and networks. It enables users to automate repetitive tasks, deploy software, and manage infrastructure through a simple, declarative language. Here, I'm going to build docker image, use kubernetes (k8s) for it's orchestration and deploy it to EC2