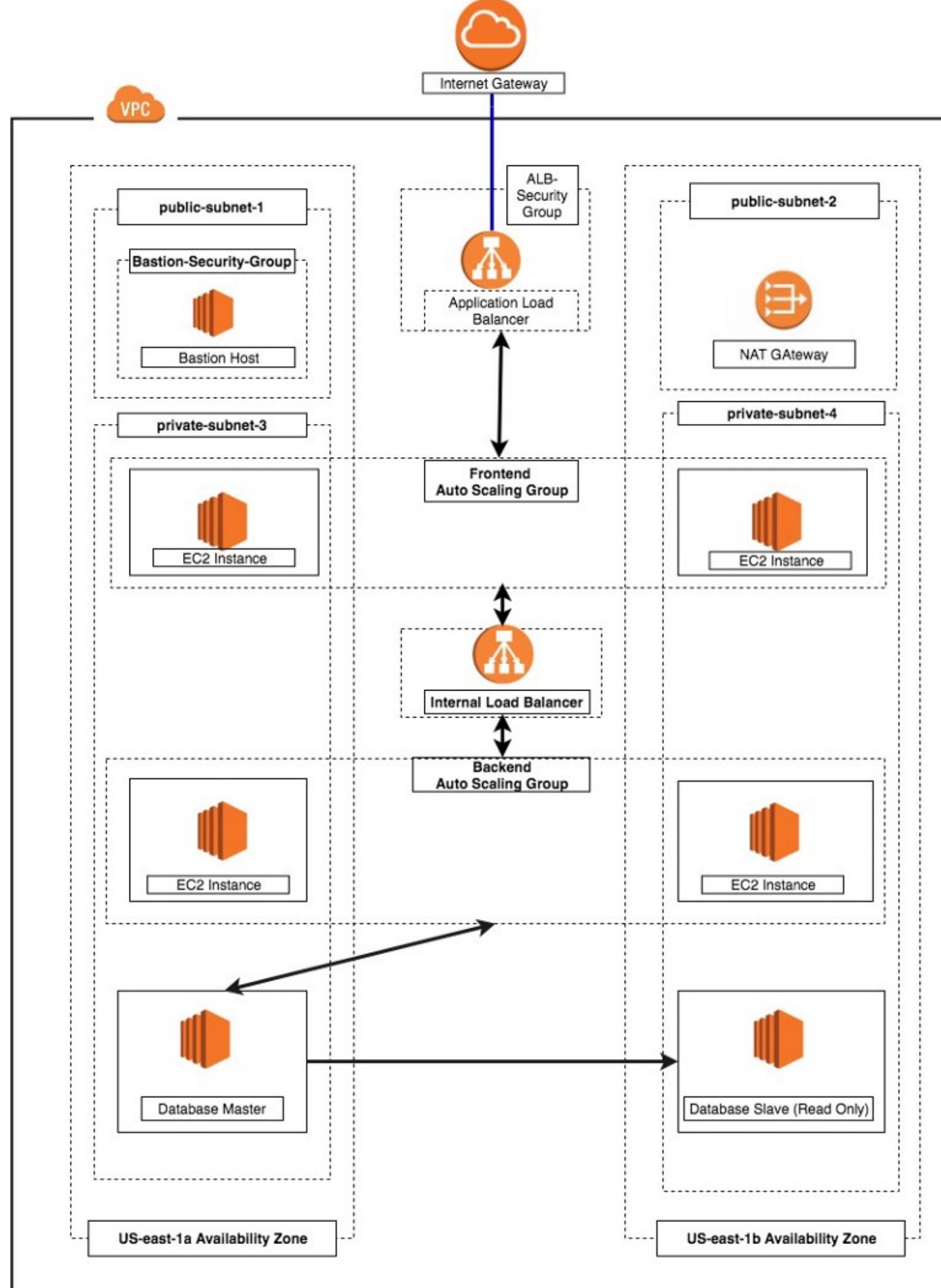


# Three Tier HA Application in AWS

Nirupama K.

# Three Tier HA Application

- The architecture is specifically for client-server web applications that has the frontend, the backend and the database.
- The application is logically broken down into three layers. Each of these layer does a specific task and can be managed independently of each other.
  - ◆ The Fronted Presentation Layer
  - ◆ Application Layer
  - ◆ Backend Database/Datastore Layer
- This is shift from traditional monolithic way of building application where frontend, backend and database sitting in one place.
- For High Availability purpose, it is a good idea to deploy application component in multiple zones or further into multiple regions depend on the critically and requirement.
- The following infrastructure is designed for High Availability and fault tolerant.



# Three Tier HA Application AWS Resources

- The application design consist following AWS resources :
  - Elastic Compute Cloud (EC2 instances) : AWS provided VM instances comes in various configurations of CPU, memory, storage, and networking capacity for your instances, known as *instance types*. In our case we have we have ***frontend tier, backend tier, bastion host deployed*** using EC2 instances.
  - AWS Autoscaling Group : An instance template configuration used to launch EC2 instances for the purpose of auto-scaling. An Auto Scaling group contains a collection of EC2 instances that are treated as a logical grouping for the purposes of automatic scaling. An Auto Scaling group also enables to use EC2 Auto Scaling features such as health check replacements and scaling policies. In our case we have been using ***Auto scaling group for fronted and backend application.***

# Three Tier HA Application AWS Resources

- The application design consist following AWS resources :
  - Security Groups: An AWS **security group** acts as a virtual firewall that controls the traffic for one or more instances. In our case we **allow HTTP port 80 and/or HTTPS port 443** on our application load balancer. For internal load balancer we **open port that the backend runs** on (for example: port 8000) and the make this port for **frontend's security group**. This will allow only the frontend to have access to that port to backend in our architecture. Another **bastion host security group** to allow SSH from outside AWS. A bastion host is just an EC2 instance that sits in the public subnet. It a best practice is to only allow SSH to this instance from trusted subnets.
  - Elastic Load Balancer (ELB): For automatically distributes incoming application traffic across multiple targets. In the above diagram the fronted application ELB is used to distribute traffic among frontend autoscaling instance group. Further on the **Internal facing load balancer** to communicate with the frontend and distribute traffic between backend autoscaling group. For **Application Load Balancer**, we select the **two public subnets** while for our **Internal Load Balancer**, we select the **two private subnet** in different availability zones.

# Three Tier HA Application AWS Resources

- The application design consist following AWS resources :
  - Virtual Private Cloud (VPC) : A networking component, logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. This provides a complete control on networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. In our case we are using a **custom VPC to deploy all application related resources within that VPC**. For security purpose end users can only reach the **frontend through the application load balancer** which is directly connected via Internet gateway. The backend and the **database tier will be in the private subnet inside same VPC**, so can communicate with frontend tier.
  - Internet Gateway: A VPC component that allows communication between instances in our VPC and the outside world or Internet. In our architecture, our **frontend tier can only accept traffic from the elastic load balancer** which connects directly with the **Internet gateway** while our backend tier will receive traffic through the **internal load balancer**.

**Thank You**

**Nirupama K.**