Q1. A Food Based Startup wants to setup its internal network

a) Difference between Public IP and Private IP

Public IP Address

- Public IP is used to connect the startup's network to the internet.
- It is unique and provided by the Internet Service Provider (ISP).
- It allows customers to access the company's website or online food ordering system.
- Example: 8.8.8.8
- Used for public-facing services like website and cloud servers.

Private IP Address

- Private IP is used inside the startup's internal network.
- It is not accessible directly from the internet.
- It is used for internal systems such as billing computers, kitchen order systems, and employee laptops.
- Example: 192.168.1.1
- Same private IP range can be reused in other organizations.

b) CIDR Block and Usable IPs

CIDR stands for **Classless Inter-Domain Routing**.
It is used to divide a network into subnets and manage IP addresses efficiently.

If the food startup is assigned 192.168.0.0/24:

- Total IP addresses = 256
- Network address = 192.168.0.0
- Broadcast address = 192.168.0.255
- **Usable IP addresses = 254**

These IPs can be used for staff systems, POS machines, servers, and printers.

c) Blocking Websites in Company Network

To ensure employees focus on work, the startup can block certain websites by configuring:

- Firewall
- Router
- Proxy server
- DNS filtering

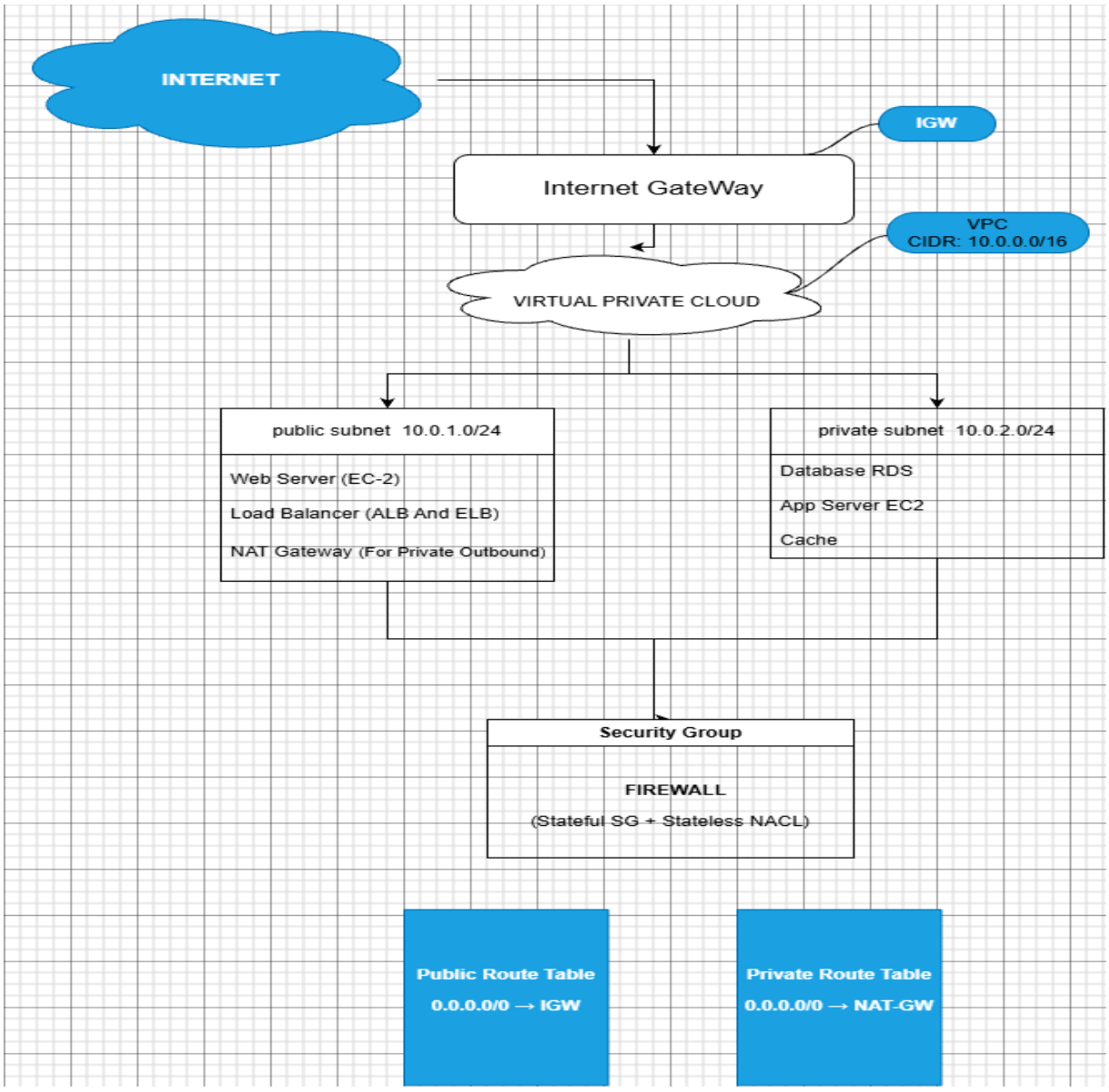
This configuration is usually done at the **firewall or router level** while setting up the company network.

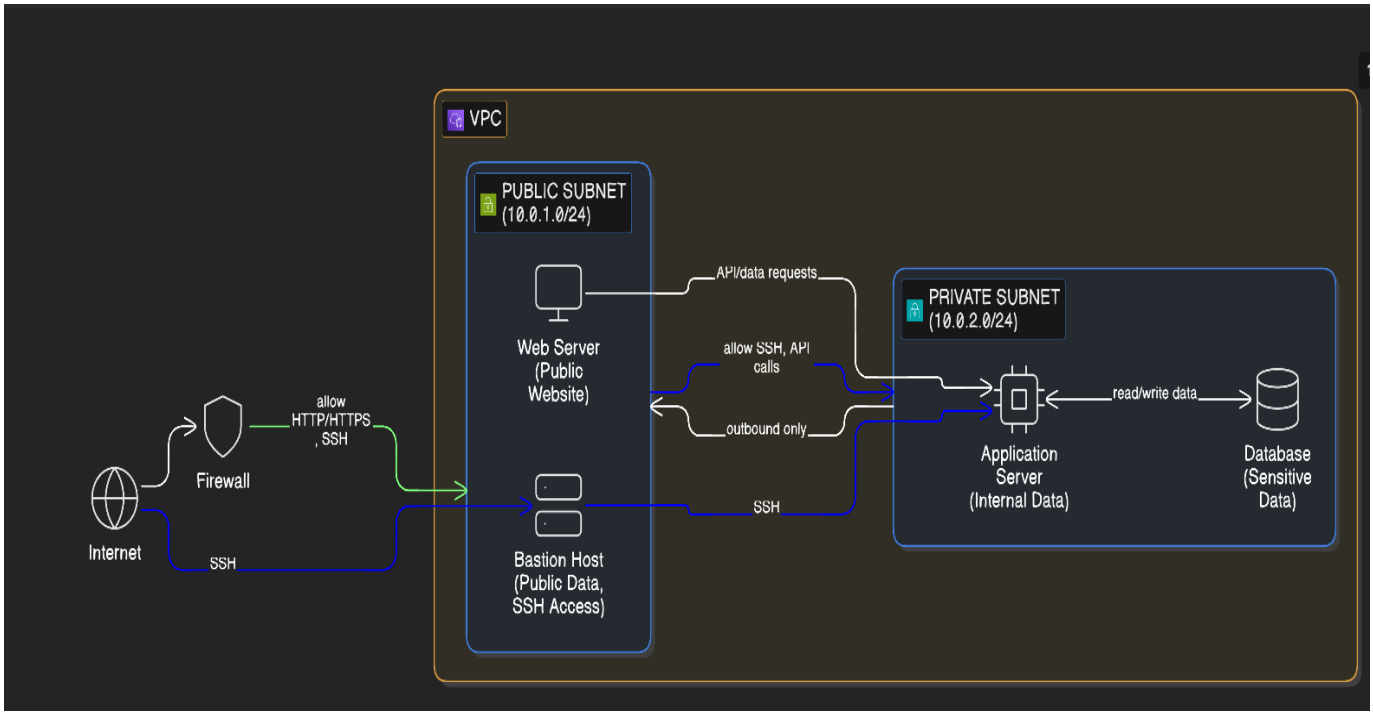
2. Draw a simple diagram showing:

• A VPC with one public subnet and one private subnet. o Divide the IPs between public and private subnet

• Firewall with inbound and outbound rules to configure access

• Which data of an organization will reside in public and private subnet. o Explain how user can reach from the internet to private subnet.





IP Division Example

- VPC CIDR: 10.0.0.0/16 (provides a large address space)
- Public Subnet: 10.0.1.0/24 (first 256 addresses, e.g., 10.0.1.0 – 10.0.1.255)
- Private Subnet: 10.0.2.0/24 (next 256 addresses, e.g., 10.0.2.0 – 10.0.2.255) This leaves the rest of the VPC CIDR (e.g., 10.0.3.0/24 onward) available for future subnets.

Firewall / Access Control In AWS, access is controlled primarily via **Security Groups** (instance-level stateful firewalls) and **Network ACLs** (subnet-level stateless firewalls).

- Inbound rules: Allow specific traffic (e.g., HTTP/HTTPS on port 80/443 to public subnet web servers).
- Outbound rules: Often allow all (default), but can be restricted. Example Security Group for public web server:
- Inbound: Allow TCP 80/443 from 0.0.0.0/0 (internet)
- Inbound: Allow SSH (22) from your IP or bastion only
- Outbound: Allow all Example for private database:
- Inbound: Allow TCP 3306 (MySQL) only from public subnet's security group
- Outbound: Allow all (or restricted)

Data Placement in Organization

- **Public Subnet** (internet-facing):
    - Web servers / front-end applications

    - Load balancers (ALB/ELB)
    - Bastion hosts / jump servers (for secure admin access)
    - Any resource that needs direct public IP / internet exposure
- Private Subnet (internal, not directly accessible from internet):
    - Databases (RDS, etc.)
    - Application/business logic servers
    - Internal services, caches (ElastiCache), queues
    - Sensitive data storage/processing (to minimize exposure)

How Users Reach Private Subnet from the Internet Private subnet resources have no public IPs and no direct route to the internet, so direct access is blocked for security. Common methods:

1. Bastion Host / Jump Server (most common for admin access):
    - Deploy a hardened EC2 instance in the public subnet (with public IP).
    - Users SSH/RDP to the bastion from internet (allowed via security group).
    - From bastion, SSH/RDP to private instances (using private IPs, allowed via security groups).
2. VPN / AWS Client VPN or Site-to-Site VPN:
    - Connect your on-premises network or client devices to the VPC via VPN → access private resources as if on the internal network.
3. AWS Systems Manager Session Manager (passwordless, no bastion needed):
    - Use IAM permissions to start secure shell sessions to private instances via the AWS console/CLI (no inbound ports open).
4. Application Access (e.g., web apps):
    - Users access via public load balancer (in public subnet) → forwards traffic to private app/database servers (no direct private access).