# AWS Network and Compute Workload

Ahsan Farooqui

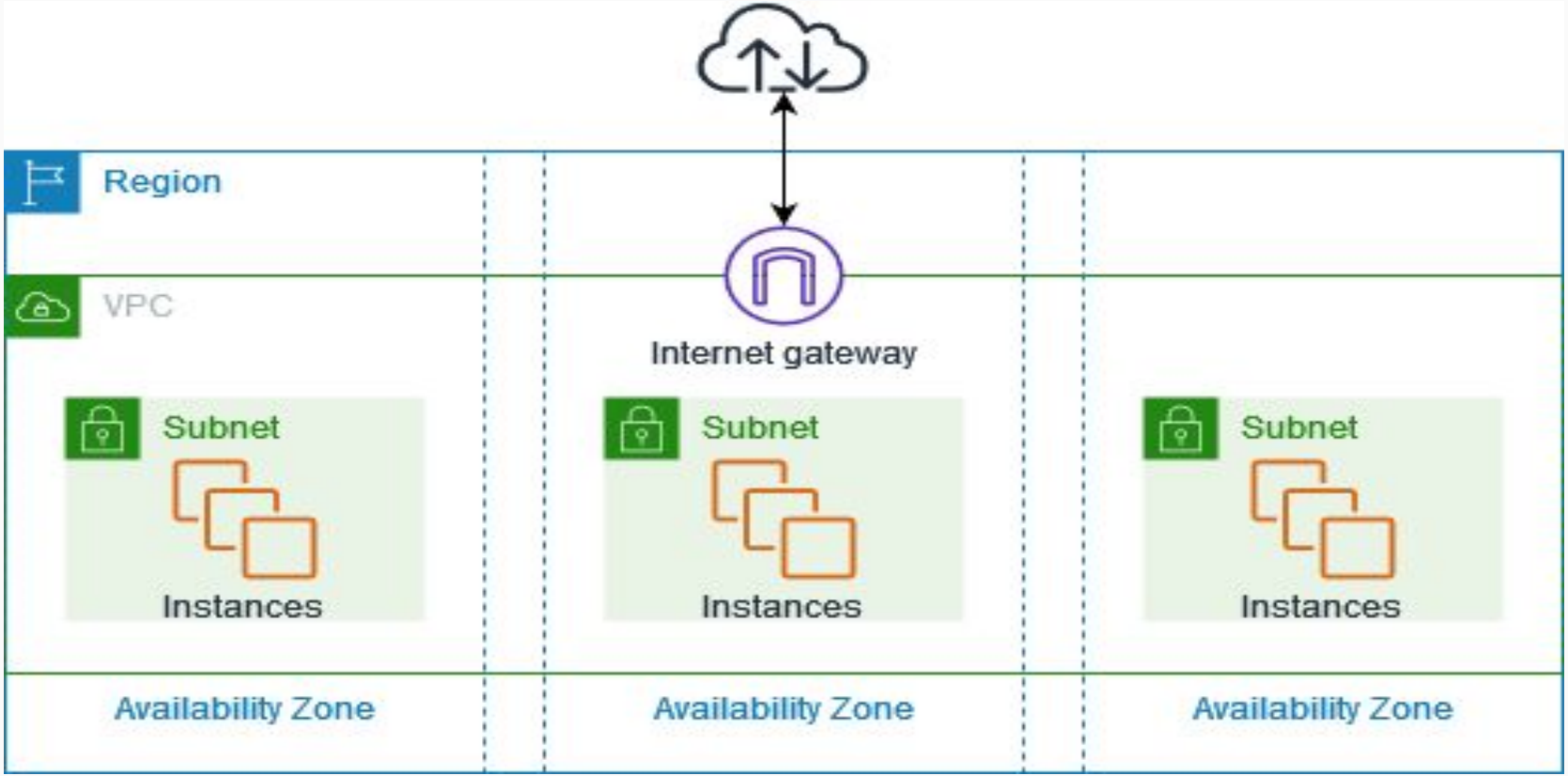
# Roadmap

- What is Amazon VPC?
- Amazon VPC Features
- Shared Security Model
- Infrastructure Security in Amazon VPC
- Security Group Vs Network ACL
- VPC with Other Services
- VPC Example
  - Test Environment
  - Webservers and Database servers
  - Private Server

# What is Amazon VPC?

- With Amazon Virtual Private Cloud (Amazon VPC), you can launch AWS resources in a logically isolated virtual network that you've defined.
- This virtual network closely resembles a traditional network that yo 'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.
- The following diagram shows an example VPC. The VPC has one subnet in each of the Availability Zones in the Region, EC2 instances in each subnet, and an internet gateway to allow communication between the resources in your VPC and the internet.

# What is Amazon VPC?



# Amazon VPC Features

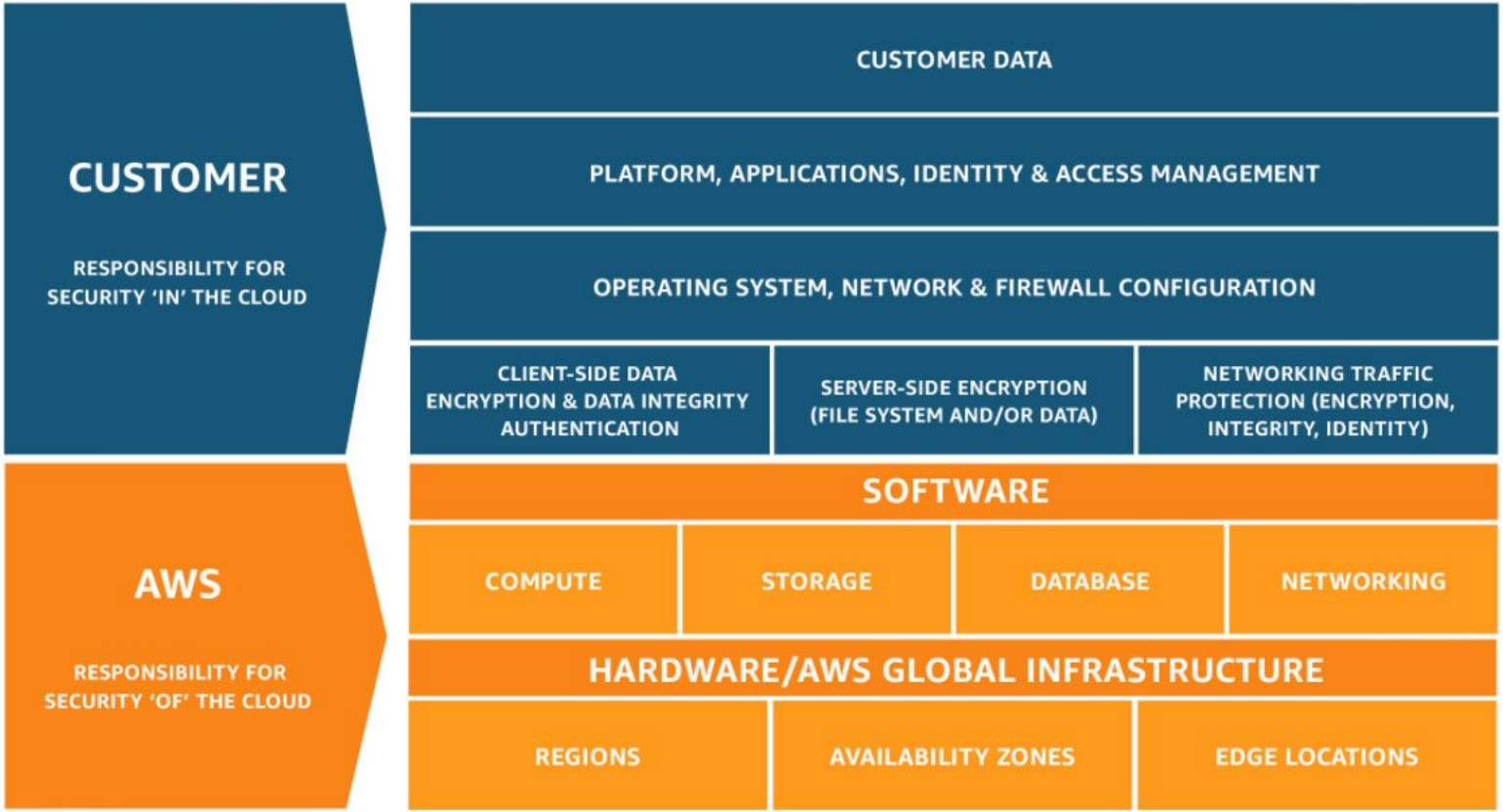
**The following features help you configure a VPC to provide the connectivity that your applications need:**

- **Virtual private clouds (VPC)**
  - **A VPC is a virtual network that closely resembles a traditional network that you'd operate in your own data center. After you create a VPC, you can add subnets.**
- **Subnets**
  - **A subnet is a range of IP addresses in your VPC. A subnet must reside in a single Availability Zone. After you add subnets, you can deploy AWS resources in your VPC.**
- **IP addressing**
  - **You can assign IP addresses, both IPv4 and IPv6, to your VPCs and subnets. You can also bring your public IPv4 and IPv6 GUA addresses to AWS and allocate them to resources in your VPC, such as EC2 instances, NAT gateways, and Network Load Balancers.**
- **Routing**
  - **Use route tables to determine where network traffic from your subnet or gateway is directed.**

# Amazon VPC Features

- Gateways and endpoints
  - A gateway connects your VPC to another network. For example, use an internet gateway to connect your VPC to the internet. Use a VPC endpoint to connect to AWS services privately, without the use of an internet gateway or NAT device.
- Peering connections
  - Use a VPC peering connection to route traffic between the resources in two VPCs.
- Traffic Mirroring
  - Copy network traffic from network interfaces and send it to security and monitoring appliances for deep packet inspection.
- Transit gateways
  - Use a transit gateway, which acts as a central hub, to route traffic between your VPCs, VPN connections, and AWS Direct Connect connections.
- VPC Flow Logs
  - A flow log captures information about the IP traffic going to and from network interfaces in your VPC.
- VPN connections
- Connect your VPCs to your on-premises networks using AWS Virtual Private Network (AWS VPN).

# Shared Security Model



As a managed service, Amazon Virtual Private Cloud is protected by the AWS global network security.

# Network isolation

A virtual private cloud (VPC) is a virtual network in your own logically isolated area in the AWS Cloud. Use separate VPCs to isolate infrastructure by workload or organizational entity.

A subnet is a range of IP addresses in a VPC. When you launch an instance, you launch it into a subnet in your VPC. Use subnets to isolate the tiers of your application (for example, web, application, and database) within a single VPC. Use private subnets for your instances if they should not be accessed directly from the internet.

You can use AWS PrivateLink to enable resources in your VPC to connect to AWS services using private IP addresses, as if those services were hosted directly in your VPC. Therefore, you do not need to use an internet gateway or NAT device to access AWS services.

# Control network traffic

**Consider the following options for controlling network traffic to the resources in your VPC, such as EC2 instances:**

- **Use security groups as the primary mechanism for controlling network access to your VPCs. When necessary, use network ACLs to provide stateless, coarse-grain network control. Security groups are more versatile than network ACLs, due to their ability to perform stateful packet filtering and create rules that reference other security groups. Network ACLs can be effective as a secondary control (for example, to deny a specific subset of traffic) or as high-level subnet guard rails. Also, because network ACLs apply to an entire subnet, they can be used as defense-in-depth in case an instance is ever launched without the correct security group.**
- **Use private subnets for your instances if they should not be accessed directly from the internet. Use a bastion host or NAT gateway for internet access from instances in private subnets.**

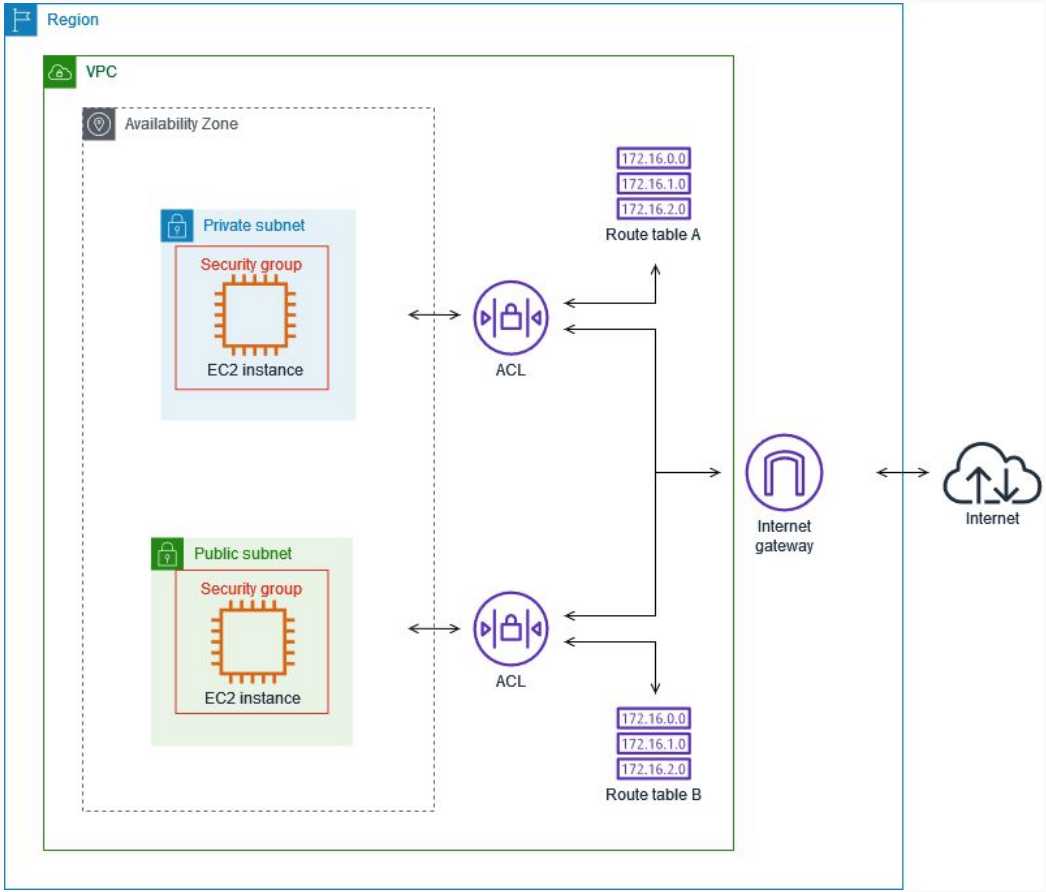
# Control network traffic

- Configure subnet route tables with the minimum network routes to support your connectivity
- requirements.
- Consider using additional security groups or network interfaces to control and audit Amazon EC2 instance management traffic separately from regular application traffic. Therefore, you can implement special IAM policies for change control, making it easier to audit changes to security group rules or automated rule-verification scripts. Multiple network interfaces also provide additional options for controlling network traffic, including the ability to create host-based routing policies or leverage different VPC subnet routing rules based on an network interfaces assigned to a subnet.
- Use AWS Virtual Private Network or AWS Direct Connect to establish private connections from your remote networks to your VPCs. For more information, see Network-to-Amazon VPC connectivity options.
- Use VPC Flow Logs to monitor the traffic that reaches your instances.

# Infrastructure Security Security Group vs Network ACL

| Security group | Network ACL |
| --- | --- |
| Operates at the instance level | Operates at the subnet level |
| Applies to an instance only if it is associated with the instance | Applies to all instances deployed in the associated subnet (providing an additional layer of defense if security group rules are too permissive) |
| Supports allow rules only | Supports allow rules and deny rules |
| Evaluates all rules before deciding whether to allow traffic | Evaluates rules in order, starting with the lowest numbered rule, when deciding whether to allow traffic |
| Stateful: Return traffic is allowed, regardless of the rules | Stateless: Return traffic must be explicitly allowed by the rules |

# Infrastructure Security Security Group vs Network ACL




**The following diagram illustrates the layers of security provided by**

- **security groups and**
- **network ACLs.**

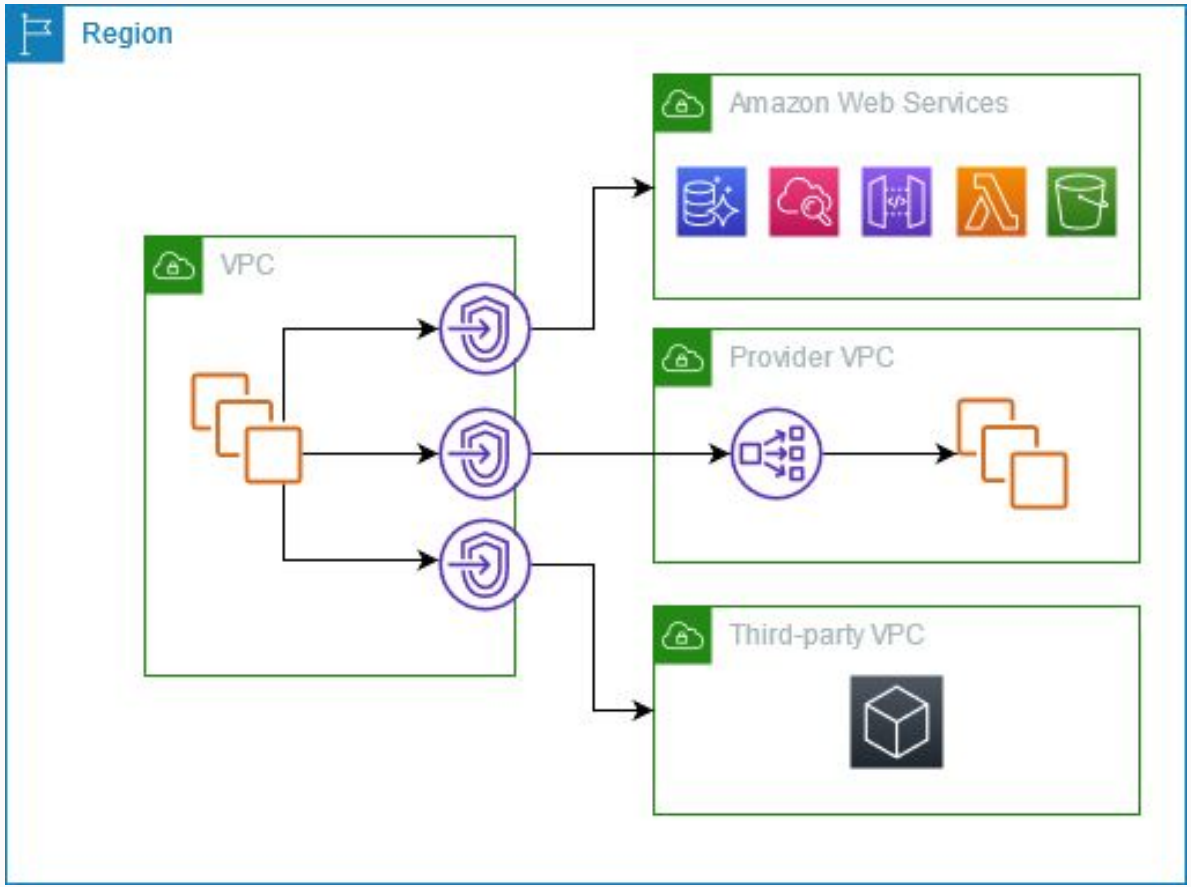
**For example,**

- **traffic from an internet gateway is routed to the appropriate subnet using the routes in the routing table.**
  - **The rules of the network ACL that is associated with the subnet control which traffic is allowed to the subnet.**
- **The rules of the security group that is associated with an instance control which traffic is allowed to the instance.**

# VPC With Other Services

- **AWS PrivateLinks**
  - AWS PrivateLink establishes private connectivity between virtual private clouds (VPC) and supported AWS services, services hosted by other AWS accounts, and supported AWS Marketplace services. You do not need to use an internet gateway, NAT device, AWS Direct Connect connection, or AWS Site-to-Site VPN connection to communicate with the service
- AWS Network Firewall.
  - You can filter network traffic at the perimeter of your VPC using AWS Network Firewall. Network Firewall is a stateful, managed, network firewall and intrusion detection and prevention service.
- Filter DNS Traffic Using Route 53  Resolver  DNS Firewall
  - With DNS Firewall, you define domain name filtering rules in rule groups that you associate with your VPCs. You can specify lists of domain names to allow or block, and you can customize the responses for the DNS queries that you block.

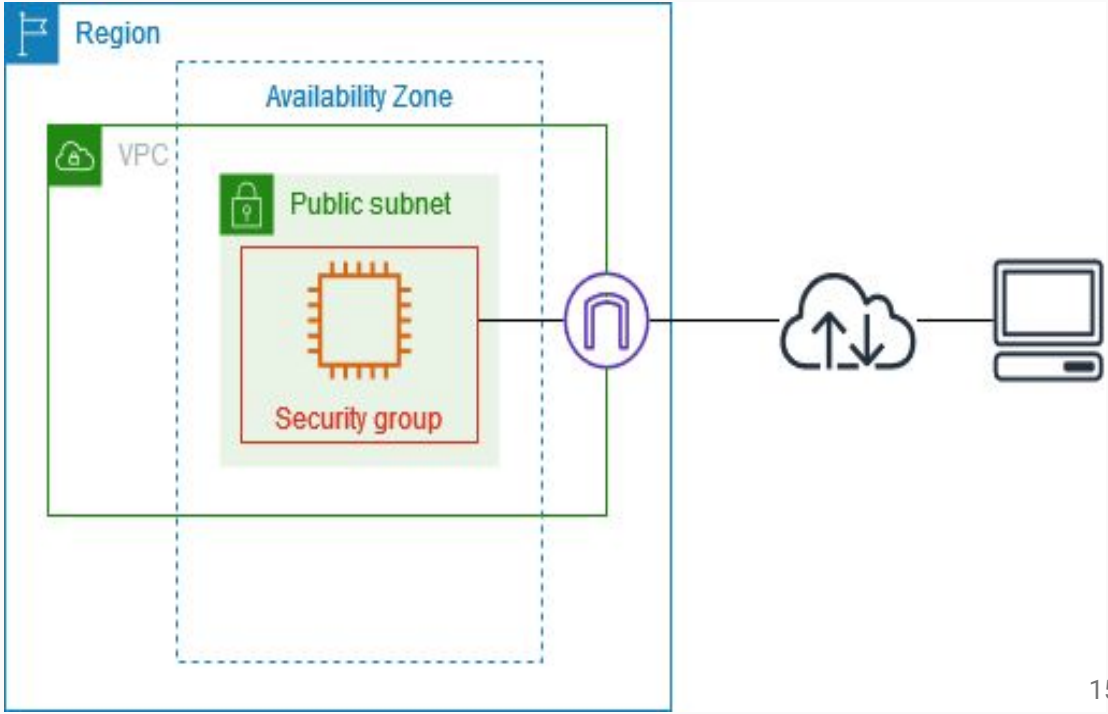
# VPC & AWS PrivateLinks



# VPC Example-Test Environment

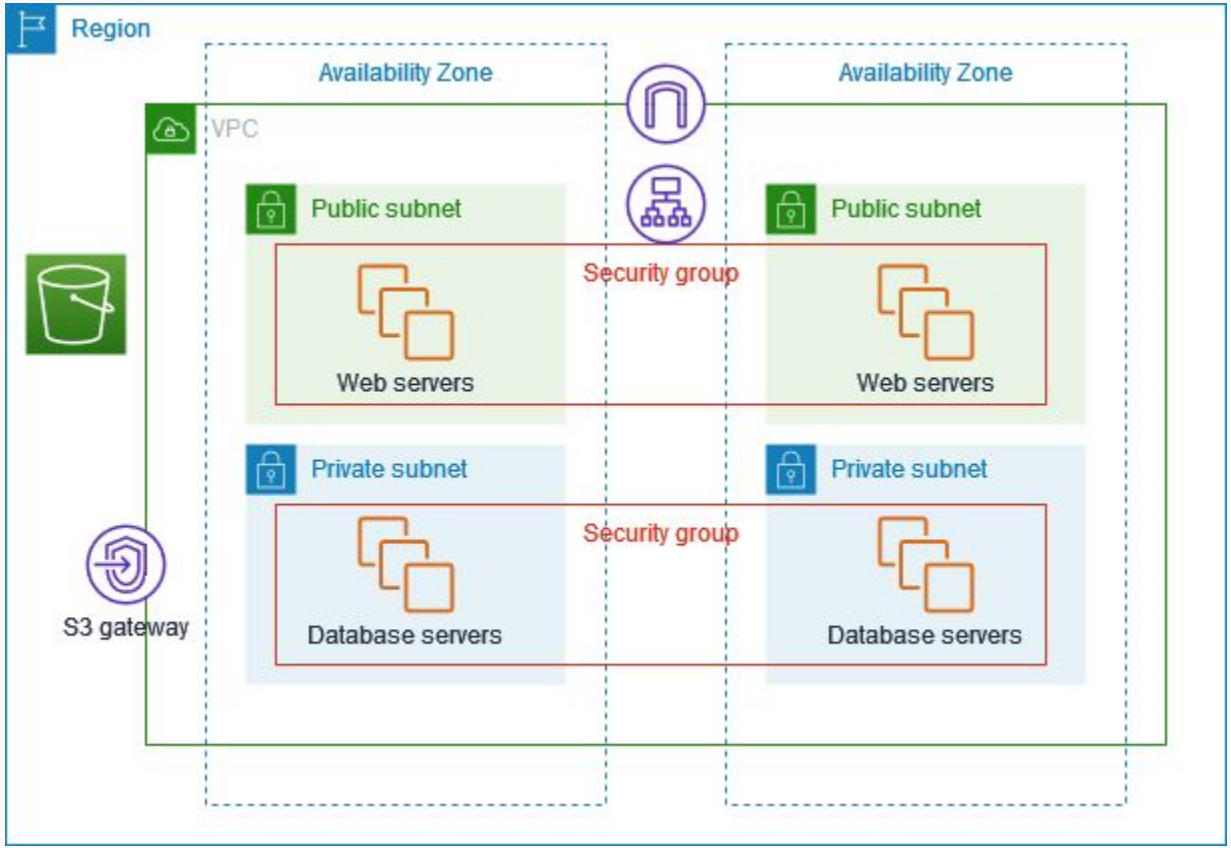
**This example demonstrates how to create a VPC that you can use as a development or test environment. Because this VPC is not intended to be used in production, it is not necessary to deploy your servers in multiple Availability Zones. To keep the cost and complexity low, you can deploy your servers in a single Availability Zone.**

**The following diagram provides an overview of the resources included in this example. The VPC has a public subnet in a single Availability Zone and an internet gateway. The server is an EC2 instance that runs in the public subnet. The security group for the instance allows SSH traffic from your own computer, plus any other traffic specifically required for your development or testing activities.**



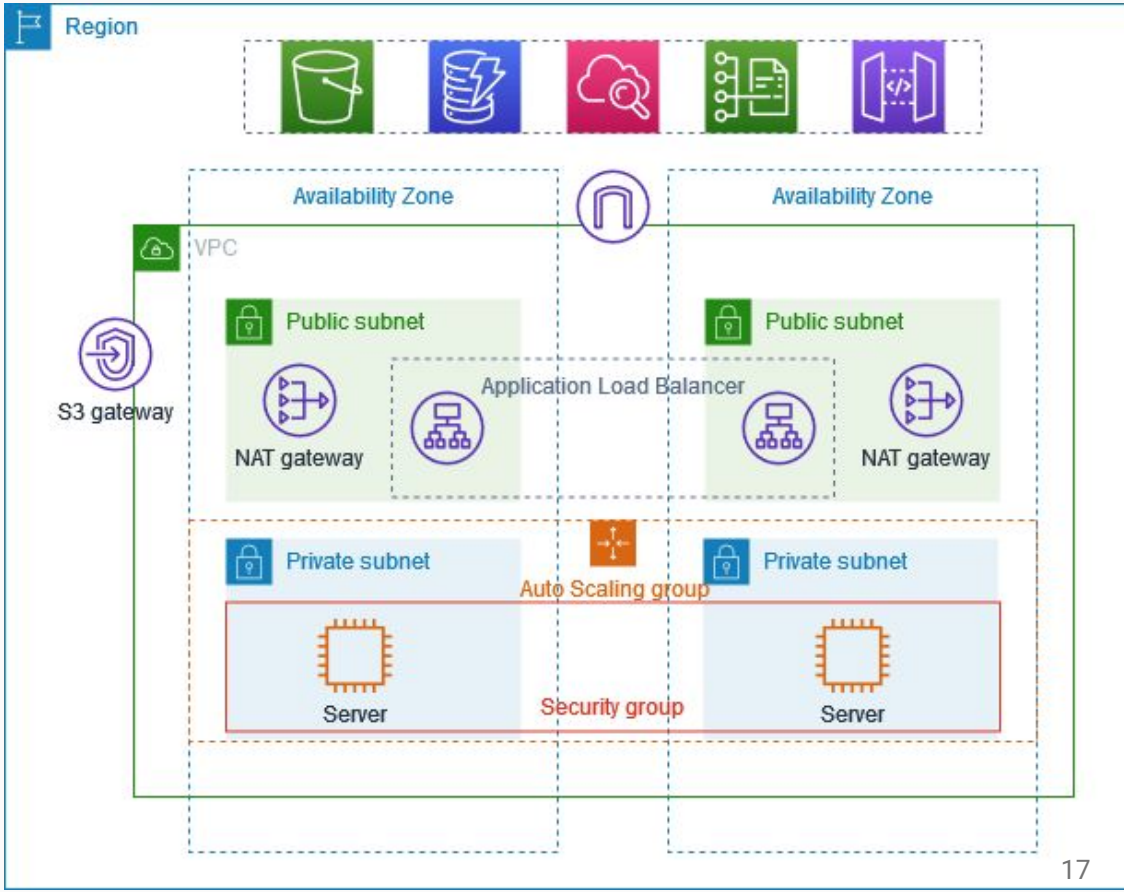
# VPC Example-Web and database server

The following diagram provides an overview of the resources included in this example. The VPC has public subnets and private subnets in two Availability Zones. The web servers run in the public subnets and receive traffic from clients through a load balancer. The security group for the web servers allows traffic from the load balancer. The database servers run in the private subnets and receive traffic from the web servers. The security group for the database servers allows traffic from the web servers. The database servers can connect to Amazon S3 by using a gateway VPC endpoint.



# VPC Example-Private server

The following diagram provides an overview of the resources included in this example. The VPC has public subnets and private subnets in two Availability Zones. Each public subnet contains a NAT gateway and a load balancer node. The servers run in the private subnets, are launched and terminated by using an Auto Scaling group, and receive traffic from the load balancer. The servers can connect to the internet by using the NAT gateway. The servers can connect to Amazon S3 by using a gateway VPC endpoint.



# Protecting Compute Resources

Compute resources in your workload require multiple layers of defense to help protect from external and internal threats. Compute resources include

- EC2 instances
- Containers,
- AWS Lambda functions,
- Database services
- IoT devices, and more.

# Protecting Compute Resources

- Perform Vulnerability Assessment
  - Regularly scan and patch resources such as Amazon EC2 instances, Amazon Elastic Container Service (Amazon ECS) containers, and Amazon Elastic Kubernetes Service (Amazon EKS) workloads. Configure maintenance windows for AWS managed resources, such as Amazon Relational Database Service (Amazon RDS) databases.
  - AWS services
    - Amazon Inspector continually scans AWS workloads for software issues and unintended network access.
    - AWS Systems Manager Patch Manager helps manage patching across your Amazon EC2 instances.
    - AWS Security Hub, a cloud security posture management service that helps automate AWS security checks and centralize security alerts.
    - Amazon CodeGuru can help identify potential

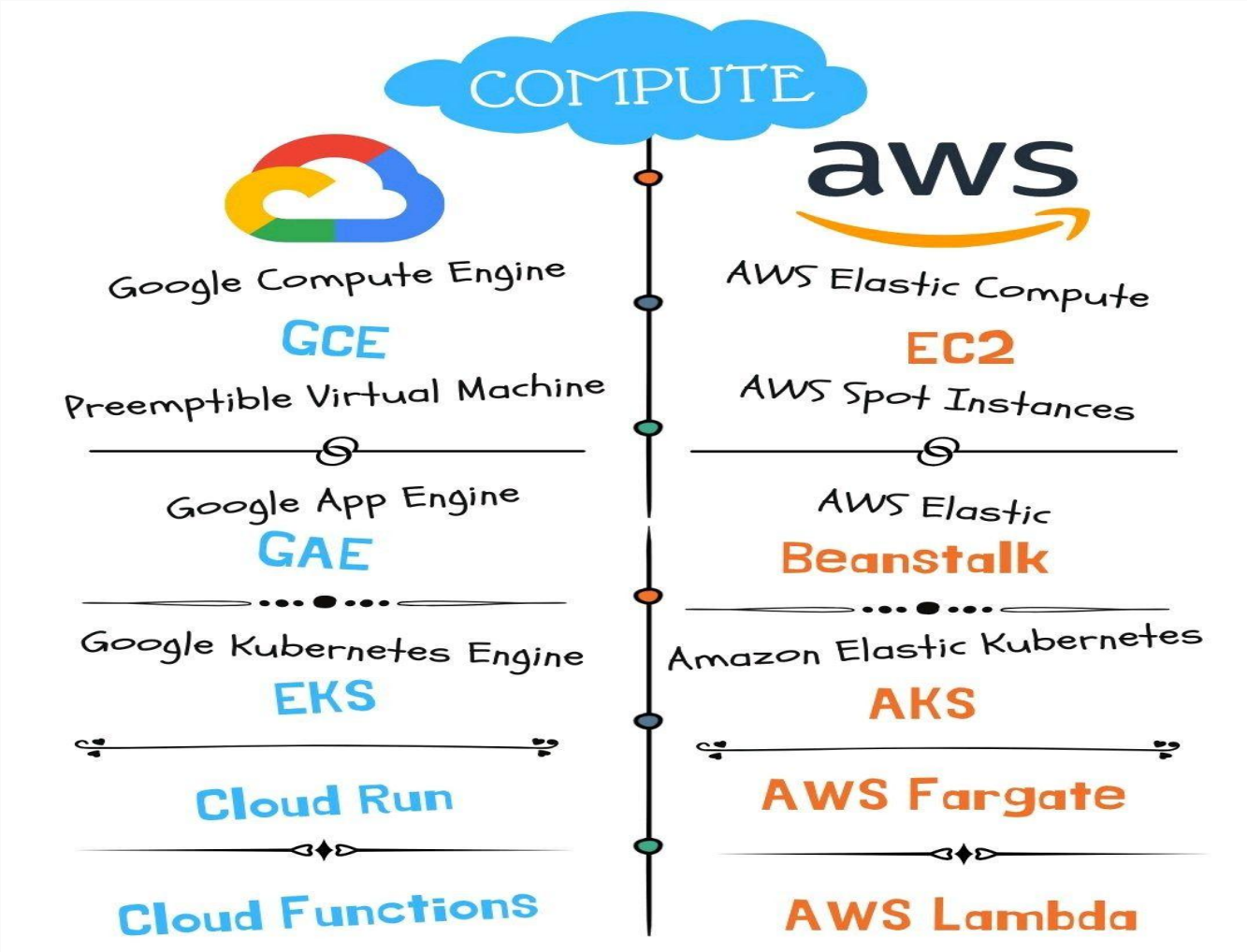
- Reduced Attack Surface
  - Reduce your exposure to unintended access by hardening operating systems and minimizing the components, libraries, and externally consumable services in use
  - In Amazon EC2, you can create your own Amazon Machine Images (AMIs), which you have patched and hardened, to help you meet the specific security requirements for your organization.
  - You can simplify the process of building secure AMIs with EC2 Image Builder.
  - When software updates become available, Image Builder automatically produces a new image without requiring users to manually initiate image builds.

# Protecting Compute Resources

- Implement Managed Services
  - Implement services that manage resources, such as Amazon Relational Database Service (Amazon RDS),AWS Lambda, and Amazon Elastic Container Service (Amazon ECS), to reduce your security maintenance tasks as part of the shared responsibility model.
- Automate Compute Protection
  - Automate your protective compute mechanisms including vulnerability management, reduction in attack surface, and management of resources. The automation will help you invest time in securing othera spects of your workload, and reduce the risk of human error.
- Enable Peoples to perform at Distance
  - Removing the ability for interactive access reduces the risk of human error, and the potential for manual configuration or management. For example, use a change management workflow to deploy Amazon Elastic Compute Cloud (Amazon EC2) instances using infrastructure-as-code, then manage Amazon EC2 instances using tools such as AWS Systems Manager instead of allowing direct access or through a bastion host. AWS Systems Manager can automate a variety of maintenance and deployment tasks.

# Protecting Compute Resources

- **Validate Software Integrity**
  - **Implement mechanisms (for example, code signing) to validate that the software, code and libraries used in the workload are from trusted sources and have not been tampered with.**
  - **For example, you should verify the code signing certificate of binaries and scripts to confirm the author, and ensure it has not been tampered with since created by the author.**
  - **AWS Signer can help ensure the trust and integrity of your code by centrally managing the code- signing lifecycle, including signing certification and public and private keys.**


COMPUTE
aws
Google Compute Engine
GCE
AWS Elastic Compute
EC2
Preemptible Virtual Machine
AWS Spot Instances
Google App Engine
GAE
AWS Elastic
Beanstalk
Google Kubernetes Engine
EKS
Amazon Elastic Kubernetes
AKS
Cloud Run
AWS Fargate
Cloud Functions
AWS Lambda