

AI Networking For the Cloud (GPT-4)

Prompts

- Each CSP will use 1 simple prompt and 1 detailed prompt.
- The simple prompt will see how GPT fares with very little context provided.
- The detailed prompt will see how GPT does with overly explicit prompts.

AWS

Simple Prompt

Put together an AWS networking solution design for a company called CornHub. They provide a streaming service. Their customers are mostly in Eastern US and Eastern Australia. They need multiple VPCs deployed with best practices for several 3-tier apps. They also need to provide connectivity to their existing on-premises network as well.

Simple Response

Sure, here is a potential AWS networking solution for CornHub:

****1. Virtual Private Cloud (VPC) Setup:****

We will create two VPCs, one in the US East region (N. Virginia) and another one in the Asia Pacific region (Sydney) as these are the closest regions to CornHub's primary customer bases.

Each VPC will have multiple subnets (each for different tiers of our applications). Subnets will be separated into at least two availability zones for high availability and fault tolerance:

- Public Subnets for the web tier.
- Private Subnets for the application and data tiers.

The use of Network ACLs and Security Groups will be applied to each subnet to control incoming and outgoing traffic at the subnet and instance levels.

****2. Application Load Balancers (ALB):****

Each application deployed will have an ALB in front of the web tier for efficient distribution of incoming traffic across multiple targets, such as EC2 instances.

****3. Auto Scaling Groups (ASG):****

The web and application tier should have ASGs to ensure scalability and high availability. The ASG can scale out (increase the number of instances) when demand increases and scale in (decrease the number of instances) when demand decreases.

****4. Amazon RDS:****

For the database tier, Amazon RDS is recommended with Multi-AZ deployment for failover support and enhanced availability and durability.

****5. AWS Direct Connect:****

To provide a dedicated network connection from CornHub's on-premises network to AWS, we can use AWS Direct Connect. This service provides a more consistent network experience than internet-based connections and allows seamless integration between the cloud and on-premises resources.

****6. AWS Transit Gateway:****

To manage connectivity across multiple VPCs and the on-premises network, we can use AWS Transit Gateway. This service simplifies the process of scaling connectivity across multiple VPCs and offers enhanced security and routing controls.

****7. AWS CloudFront and Route53:****

AWS CloudFront, a content delivery network (CDN), can be used to deliver the streaming content to the end users. This will help reduce latency and improve the overall user experience.

Route53 will be used for DNS management, routing the end users to the appropriate CloudFront edge locations.

****8. AWS WAF and AWS Shield:****

To protect the application from common web exploits, we will use AWS WAF (Web Application Firewall) and to safeguard against DDoS attacks, we will use AWS Shield.

****9. Monitoring and Logging:****

Use CloudWatch for monitoring resources and applications, collecting and tracking metrics, and setting alarms. Use CloudTrail to log, continuously monitor, and retain account activity related to actions across your AWS infrastructure.

Remember that while this solution should be robust enough to handle CornHub's requirements, the implementation details may vary based on exact needs, such as the specifics of the on-premises network or the types of applications being deployed.

Detailed Prompt

Put together an AWS networking solution design for a company called CornHub. They specialise in a vegetable streaming service offered to customers around the world. Here are the requirements:

- Two regions will be used: Eastern US and Sydney, Australia
- AWS Transit Gateway should be used to connect all deployed VPCs together
- AWS Direct Connect should be used to provide hybrid connectivity to our regional data centres in Ashburn, Virginia and Sydney, Australia
- Each region should contain dedicated "PROD" and "NON-PROD" VPC to house public-facing applications that will sit behind Application Load Balancers. These VPCs should follow a 3-tier app architecture
- The PROD and NON-PROD environments should not be able to communicate with each other
- The data centres should be able to communicate with each other over the Direct Connect links, as we will use this as a backup for the MPLS network

Detailed Response

The following is a proposed AWS networking solution for CornHub, taking into account all the requirements provided:

****1. AWS Direct Connect Setup:****

Establish AWS Direct Connect connections from your on-premises data centers in Ashburn, Virginia and Sydney, Australia to the nearest AWS Direct Connect locations. This will provide a dedicated network connection from your regional data centres to AWS.

****2. VPC Design:****

Each region (Eastern US and Sydney, Australia) will have two Virtual Private Clouds (VPCs): PROD and NON-PROD.

These VPCs should be designed in a 3-tier architecture model, consisting of:

- A Public Subnet: Hosting the Application Load Balancers (ALB) to distribute incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones.
- An Application Subnet: This tier would contain application servers such as web servers and application servers.
- A Database Subnet: This tier would contain database servers for applications.

Each tier will have multiple subnets across different Availability Zones to ensure high availability and fault tolerance.

****3. Network Isolation:****

To ensure that the PROD and NON-PROD environments do not communicate with each other, different VPCs will be created for each. AWS Network ACLs (NACLs) will be configured to restrict traffic between these VPCs.

****4. AWS Transit Gateway:****

Implement an AWS Transit Gateway in each region to act as a network transit hub, allowing you to manage and control network traffic across your Amazon VPCs and VPN connections. Attach all the VPCs in their respective regions to their regional Transit Gateways.

****5. VPC Routing Tables:****

Modify the routing tables of your VPCs to ensure they can reach each other through the Transit Gateway, excluding the routes between PROD and NON-PROD as per the requirements.

****6. VPN Connections (Backup):****

As MPLS backup, create VPN connections over your Direct Connect links. If Direct Connect goes down, you can failover to VPN connections.

****7. Security Groups and NACLs:****

Implement Security Groups and Network Access Control Lists (NACLs) to provide stateful and stateless traffic control respectively, helping to provide a secure environment for your applications and data.

****8. Data Encryption:****

Consider using AWS Key Management Service (KMS) to handle encryption and decryption of your data at rest and in transit.

Please note that this is a high-level design and does not contain the detailed configurations required for implementing the solution. It is recommended to work closely with an experienced AWS solution architect or partner when building out this solution.

Azure

Simple Prompt

Put together an Azure networking solution design for a company called CornHub. They provide a streaming service. Their customers are mostly in Eastern US and Eastern Australia. They need multiple VNets deployed with best practices for several 3-tier apps. They also need to provide connectivity to their existing on-premises network as well.

Simple Response

To design an Azure networking solution for CornHub, a streaming service company with customers in Eastern US and Eastern Australia, we will follow best practices and ensure connectivity to their existing on-premises network. Here's an outline of the Azure networking solution design:

1. Virtual Networks (VNETs):

- Create two separate VNETs, one for the Eastern US region and another for the Eastern Australia region. This helps with regional redundancy and reduces latency for customers in each region.
- Choose appropriate IP address ranges for each VNET, ensuring they don't overlap with each other or the on-premises network.

2. Subnets:

- Within each VNET, create subnets to accommodate the three-tier application architecture (web, application, and database tiers).
- Assign appropriate IP address ranges to each subnet based on the expected number of resources and expected load.

3. Network Security Groups (NSGs):

- Create NSGs for each subnet to control inbound and outbound traffic.
- Define rules to allow necessary traffic for the three-tier applications, while restricting access to other resources as per security requirements.

4. Virtual Network Peering:

- Establish VNET peering between the US and Australia VNETs to enable seamless communication between resources in different regions.
- This allows for data replication, failover, and load balancing between the regions if required.

5. ExpressRoute or VPN Gateway:

- Set up an ExpressRoute or VPN Gateway to establish connectivity between the Azure VNETs and the on-premises network.
- ExpressRoute provides a private and dedicated connection with higher bandwidth, lower latency, and better reliability.
- VPN Gateway provides a secure connection over the internet using IPSec VPN tunnels.

6. Traffic Manager:

- Use Azure Traffic Manager to optimize user experience by directing customer traffic to the nearest streaming server location based on their geographic location.
- Configure endpoint monitoring to ensure high availability and failover in case of server or region failures.

7. Load Balancers:

- Deploy Azure Load Balancers to distribute traffic across multiple instances of the web and application tiers.
- Configure load balancing rules and health probes to ensure optimal resource utilization and availability.

8. Azure Firewall:

- Implement Azure Firewall to provide centralized network security and control outbound internet access from the VNets.
- Define appropriate rules to allow necessary outbound traffic and block unauthorized access.

9. Azure Private DNS Zones:

- Create Azure Private DNS Zones to manage and resolve internal domain names for the three-tier applications.
- Configure DNS forwarding or conditional forwarding for name resolution between Azure VNets and the on-premises network.

10. Network Monitoring and Diagnostics:

- Enable Azure Network Watcher to monitor and troubleshoot network connectivity, performance, and security.
- Utilize Azure Monitor to collect and analyze network-related logs and metrics for proactive monitoring and alerting.

Remember to consider scalability, security, and high availability requirements while implementing this Azure networking solution for CornHub. It's also important to regularly review and update the design to align with changing business needs and Azure best practices.

Detailed Prompt

Put together an Azure networking solution design for a company called CornHub. They specialise in a vegetable streaming service offered to customers around the world. Here are the requirements:

- Two regions will be used: Eastern US and Sydney, Australia
- Azure VWAN should be used to connect all deployed VNets together
- Azure ExpressRoute should be used to provide hybrid connectivity to our regional data centres in Ashburn, Virginia and Sydney, Australia
- Each region should contain dedicated "PROD" and "NON-PROD" VPC to house public-facing applications that will sit behind Azure Load Balancers. These VNets should follow a 3-tier app architecture

- The PROD and NON-PROD environments should not be able to communicate with each other
- The data centres should be able to communicate with each other over the ExpressRoute links, as we will use this as a backup for the MPLS network

Detailed Response

Based on the provided requirements, below is a proposed Azure networking solution design for CornHub.

Azure Networking Design

Regions and VNets

1. Set up VNets for the two regions: Eastern US and Sydney, Australia. Each region will host two Virtual Networks (VPCs) - PROD and NON-PROD, separated to prevent communication between production and non-production environments.
2. In each VNet, establish a three-tiered architecture for the applications: Web tier, Application tier, and Database tier. This will provide a highly scalable, secure, and reliable environment for the applications.

Azure VWAN

1. Set up Azure Virtual WAN (VWAN) to simplify large-scale branch connectivity. VWAN will bring together all the VNets deployed in both Eastern US and Sydney regions, providing a unified network management plane.

ExpressRoute

1. Use Azure ExpressRoute for secure and reliable connectivity between the regional data centres in Ashburn, Virginia and Sydney, Australia. It offers higher security, reliability, and speeds with lower latencies than typical internet connections.
2. Set up ExpressRoute circuits in both regions and peer them with your on-premises network for hybrid connectivity.

Load Balancers and Network Security Groups

1. Deploy Azure Load Balancers within the VNets of each region to distribute network traffic to the applications evenly, increasing availability and reliability of the services.

2. Implement Network Security Groups (NSGs) for each VNet and subnet within the VNets. These will act as built-in firewalls, controlling inbound and outbound traffic.

3. Set up NSG rules to block traffic between the PROD and NON-PROD VNets, ensuring these environments cannot communicate with each other.

****Backup for MPLS Network****

1. Use the ExpressRoute links to provide a backup for the MPLS network, enabling data centres to communicate with each other securely and reliably, even if the primary network fails.

Final Thoughts

This Azure networking design aims to provide a secure, scalable, and reliable solution for CornHub's global vegetable streaming service. By leveraging Azure's VWAN, ExpressRoute, Load Balancers, and NSGs, CornHub can ensure its service is always available and performing optimally for its customers around the world.

GCP

Simple Prompt

Put together a Google Cloud Platform (GCP) networking solution design for a company called CornHub. They provide a streaming service. Their customers are mostly in Eastern US and Eastern Australia. They need multiple VPCs deployed with best practices for several 3-tier apps. They also need to provide connectivity to their existing on-premises network as well.

Simple Response

Sure, I can sketch out a basic GCP networking solution for CornHub based on the requirements you've provided.

1. ****Google Cloud Virtual Private Cloud (VPC):****

- CornHub should set up at least three VPCs in each region (Eastern US and Eastern Australia): One for development, one for staging, and one for production. This allows for a more secure and isolated testing and deployment process.

- Each VPC should be set up with auto-mode IP range to automatically create subnets in each region. This eases VPC management and allows for simple expansion to other regions if necessary.

2. ****Subnets & Firewalls:****

- Each VPC will contain three subnets corresponding to the 3-tier architecture: Presentation, Application, and Database.
- Firewalls should be configured at the VPC level, with security rules defining the communication across the tiers. The database subnet should have the strictest rules, only allowing necessary traffic from the application subnet.

3. ****Interconnectivity & VPN:****

- To connect the GCP infrastructure with CornHub's on-premise network, Cloud VPN or Cloud Interconnect can be used.
- If the on-premises network has high-throughput demands, Dedicated Interconnect or Partner Interconnect might be more suitable.

4. ****Load Balancing & CDN:****

- To distribute the streaming service traffic, Cloud Load Balancing should be used. Global load balancing might be most suitable, considering that the audience is spread across different geographical regions.
- Cloud CDN (Content Delivery Network) should be implemented for delivering streaming content to end-users. This can greatly reduce latency and improve the streaming experience for users in the Eastern US and Eastern Australia.

5. ****Private Google Access & Cloud NAT:****

- Enable Private Google Access to let VM instances in subnets send traffic to Google APIs and services without external IP addresses.
- Set up Cloud NAT (Network Address Translation) to enable instances without an external IP address to access the Internet.

6. ****Shared VPC and VPC Peering:****

- For resources that need to be shared across different VPCs (like a common logging or monitoring server), a Shared VPC can be used.
- VPC Peering can also be utilized for communication across different VPCs without going through the public Internet.

7. ****Security Controls:****

- IAM (Identity and Access Management) policies should be implemented to control who can do what on the GCP environment.
- To protect against DDoS attacks, Cloud Armor should be deployed along with the load balancers.

8. ****Monitoring and Logging:****

- Use Cloud Operations (formerly Stackdriver) for real-time monitoring, troubleshooting, and logging. This will help CornHub to gain insight into their GCP environment and the behavior of their apps.

This is a high-level design and depending on the specific requirements, the actual setup might vary. Also, the configuration should follow the principle of least privilege, and always ensure that only necessary permissions are granted.

Detailed Prompt

Put together a Google Cloud Platform (GCP) networking solution design for a company called CornHub. They specialise in a vegetable streaming service offered to customers around the world. Here are the requirements:

- Two regions will be used: Eastern US and Sydney, Australia
- Google Cloud Network Connectivity Center should be used to connect all deployed VPCs together
- GCP Dedicated Interconnect should be used to provide hybrid connectivity to our regional data centres in Ashburn, Virginia and Sydney, Australia
- Each region should contain dedicated “PROD” and “NON-PROD” VPC to house public-facing applications that will sit behind Application Load Balancers. These VPCs should follow a 3-tier app architecture
- The PROD and NON-PROD environments should not be able to communicate with each other
- The data centres should be able to communicate with each other over the Dedicated Interconnect links, as we will use this as a backup for the MPLS network

Detailed Response

Based on your requirements, here is the Google Cloud Platform (GCP) networking solution design for CornHub:

****1. Google Cloud Network Connectivity Center:****

CornHub will need to set up a Network Connectivity Center. This hub will connect all Virtual Private Clouds (VPCs) deployed across the two regions: Eastern US and Sydney, Australia. This will ensure seamless connectivity between all VPCs.

****2. Google Cloud Dedicated Interconnect:****

For hybrid connectivity, CornHub should utilize Google Cloud's Dedicated Interconnect to provide a link from the company's regional data centres in Ashburn, Virginia and Sydney, Australia to their respective GCP regions. This will ensure a private, high-performance, and low-latency connection.

****3. VPC Setup:****

In both regions (Eastern US and Sydney), CornHub should set up two VPCs each – "PROD" and "NON-PROD". Each VPC will be configured following a 3-tier app architecture comprising of a presentation layer, application layer, and data layer.

****4. Application Load Balancers:****

The public-facing applications will be located in both the PROD and NON-PROD VPCs and will sit behind Google Cloud Load Balancers. These load balancers will distribute incoming application traffic across multiple instances to ensure high availability and reliability.

****5. Network Isolation:****

Security groups will be set up to ensure that PROD and NON-PROD environments cannot communicate with each other. This is crucial to avoid any potential contamination of production data and to maintain the integrity of the production environment.

****6. Inter Data Center Communication:****

Your data centres in Ashburn and Sydney will be able to communicate with each other over the Dedicated Interconnect links. This can be achieved by setting up Cloud Router and BGP (Border Gateway Protocol) for dynamic routing. This connection will serve as a backup for the existing MPLS network, ensuring redundancy and reliability of your network connections.

In summary, the architecture of CornHub's network on GCP would involve a combination of Network Connectivity Center, Dedicated Interconnect, multiple VPCs following a 3-tier architecture, and Application Load Balancers, all of which are designed to meet the needs of high availability, performance, security, and compliance for CornHub's unique vegetable streaming service.