Interaction Session

JJ Tech Glorious Batch

Direct Question

- What is the maximum number of AWS Regions in which an Amazon Virtual Private Cloud (Amazon VPC) can be deployed?

Answer

One (1)

An Amazon VPC is launched into one AWS Region and cannot be spread across multiple regions.

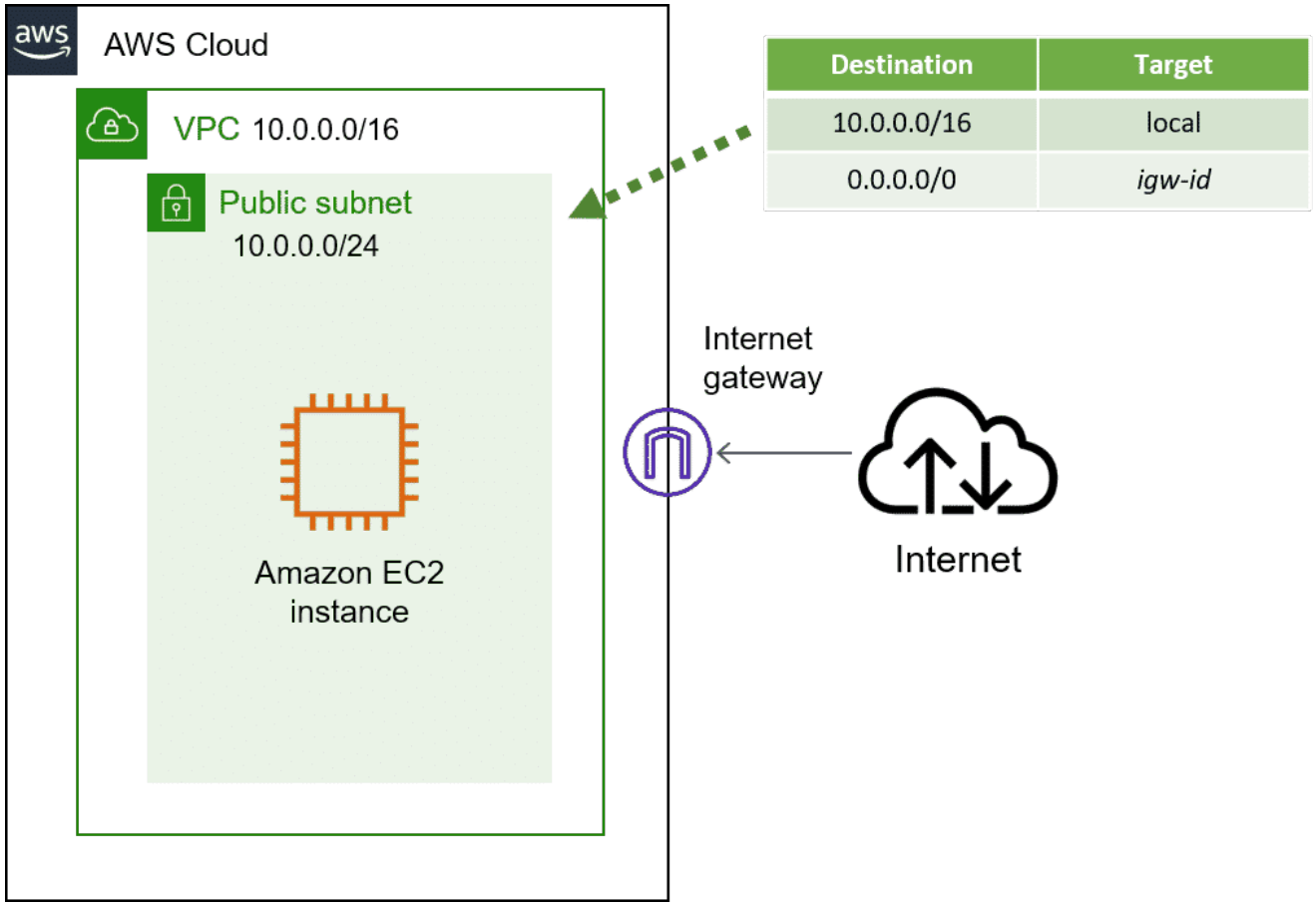
Direct Question

- There is one key architectural difference between a *public* and *private* subnet. What is that difference?

Answer

- The only architectural difference between a *public* and *private* subnet is that a public subnet has a route to an internet gateway.

Details on Public Subnet



Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*igw-id*

- In this example, the subnet is associated with a route table. The first row takes all traffic from that subnet, intended to remain within the VPC (10.0.0.0/16), and routes it within the Amazon VPC (local). The second row takes all traffic coming from that subnet and directs it to the internet gateway (igw-id). This association is what makes this subnet public.

Direct Question

- By default, all subnets in an Amazon VPC can access each other. How can you restrict traffic into and out of your subnets?

Answer

- You can use network network ACLs to restrict traffic into and out of your subnets.

Direct Question

- A single Amazon VPC can't live in more than one Region. So, you must choose the region carefully. What are the factors that you would consider when recommending the choice of a region to your customer?

Answer

- The Region in which we place our infrastructure impacts costs and, depending on where our end users live, latency.
- I will check that the services our applications need are available in the Region we select. Some Regions (especially newer ones) don't have every AWS service available.
- In summary, I will consider the following factors when I select the optimal region or regions where we store our data and want to use AWS services.
    - Data governance & legal requirement: Local laws might require that certain information must be kept within the geographical boundary. Such laws might restrict regions where you can offer content or serviced. For example, consider the European Union data protection directive.
    - Proximity to customer(latency).
    - Services available within the region.
    - Cost (vary by region).

Talking Point

- As part of our VPC design strategy, I usually make sure we choose the region carefully give than a single VPC can't live in more than one Region and the region in which we place our infrastructure impacts costs and, depending on where our end users live, latency as well.
- So, when recommending the chose of a region in which an individual VPC should be hosted, I wil consider factors such as:
  - Data governance & legal requirement
  - Proximity to customer as that will impact latency.
  - Services available within the region as some Regions (especially newer ones) don't have every AWS service available and
  - Cost as it varies by region.

Direct Question

- Internet gateways let your VPC resources reach the internet. As an SA, how do you make sure the IGW is highly available?

Answer

Internet gateways are by default:

- Horizontally scaled
- Redundant
- Highly available

This means that even though each Amazon VPC has a single internet gateway, this internet gateway is not a bottleneck nor a single point of failure.

Scenario Question

- As an Architect, you're hosting a public website on AWS. Following the requirements, you want to have the database and the app server in a private subnet on the VPC. You want to set up a database or app servers that can connect to the Internet for any patch upgrade but cannot receive any request from the internet. How can you accomplish this?

Answer

- Setup DB & App server in a private subnet which is connected to the internet via NAT for outbound.

Scenario Question

A company uses Amazon S3 to store documents that may only be accessible to an Amazon EC2 instance in a certain virtual private cloud (VPC).

The company's goal is to limit access to the S3 bucket to just the EC2 instance in that specific VPC, but it's concerned that privileged users could also set up EC2 instances in other VPCs to access these documents.

Provide a solution that will provide the required protection?

Answer

- Use an S3 VPC endpoint and an S3 bucket policy to limit access to this VPC endpoint.

https://docs.aws.amazon.com/AmazonS3/latest/userguide/example-bucket-policies-vpc-endpoint.html

Story! (Talking Point)

In one of my project, we were using Amazon S3 to store documents. The documents needed to be accessible to Amazon EC2 instances in a certain virtual private cloud (VPC).

The company's goal was to limit access to the S3 bucket from only the EC2 instances in that specific VPC but was concerned that privileged users could also set up EC2 instances in other VPCs to access the documents in the S3 bucket. They were also concerned about the traffic between our EC2 instances and the S3 bucket objects traversing the internet.

To address this concerns, I created a Gateway VPC endpoint to route traffic to S3; and created an S3 bucket policy to limit access to only this VPC endpoint.

Direct Question

- *You have an existing infrastructure in AWS running in a single VPC. You sized the VPC CIDIR at the time of creation to 192.168.1.0/24 and you have now run out of IP addresses to launch new servers. How should you enable launch of new servers with least optimal overhead?*

Answer

- *Add a secondary CIDIR 192.168.2.0/24 to the VPC.*

Direct Question

- *You're running an EC2 instance within a private subnet on your VPC. The EC2 instance needs access to S3 buckets. Can you recommend a solution that will allow the EC2 instance to access the S3 buckets without routing through the internet gateway.*

Answer

- *I will create a Gateway VPC endpoint to route traffic to S3.*

Direct Question

- *How can you control what resources can be accessed through the VPC endpoint?*

Answer

- *Configure a policy associated with the VPC endpoint*

- https://docs.aws.amazon.com/vpc/latest/privatelink/vpc-endpoints-s3.html#vpc-endpoints-policies-s3

Direct Question

- What is the purpose of a NAT Gateway?

Answer

- A NAT gateway is a Network Address Translation (NAT) service. You can use a NAT gateway so that instances in a private subnet can connect to services outside your VPC but external services cannot initiate a connection with those instances.

Direct Question

- From your experience, what's one possible reason that your VPC peering could fail?

Answer

- The VPCs have overlapping IP addresses.

Direct Question

- To ensure high availability, you should make your subnet span across more than one Availability Zone (make them Multi-AZ). Do you agree?

Answer

- No, because it is not possible to make a subnet Multi-AZ.
  - Subnets cannot span more than one Availability Zone.

Direct Question

- In order for resources in your public subnet to reach the internet, they need to be provided with what kind of target?

Answer

Internet gateway

An internet gateway serves two purposes:

- To provide a target in your VPC route tables for internet-routable traffic
- To perform network address translation (NAT) for instances that have been assigned public IPv4 addresses

Direct Question

- Compare Stateless to stateful controls as it relates to your VPC

Answer

Network ACLs are stateless, which means if traffic is allowed in, the outbound response to that traffic is NOT allowed out by default.

- For network ACL rules, inbound and outbound address and port will need to be explicitly added.
- Network ACLs only see the traffic going one way, so if there is an allow for an inbound rule, there must also be an allow for the outbound rule. Then the network ACL will explicitly see that traffic that was allowed inbound is also allowed out.
- Network ACLs see the traffic as two different streams, two rules are needed, one rule for each stream. If an outbound rule is not added, the traffic will only be allowed in.

Security groups, however, are stateful. If traffic is allowed in, the outbound response to that traffic is allowed out automatically.

- Security groups have inbound and outbound rules, but because security groups are stateful, that means if traffic is allowed in, that traffic is automatically allowed back out.
- Security groups see both the inbound and outbound traffic as part of the same stream.
- A difference between security groups and network ACLs is that security groups recognize AWS resources. So for an Amazon EC2 instance, the instance ID could be added to the security group rule for that instance to allow traffic from the instance. Customers can also add rules for other security groups or add a rule for the security group themselves.
- Another big distinction is that security groups have a hidden explicit deny, which means that anything that is not explicitly allowed is denied.

Direct Question

- What are some differences between Security Groups and NACLs

Security Groups

Security groups are stateful — if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules.

Responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules.

Network ACLs

- **Default Network ACLs:** Your VPC automatically comes with a default network ACL, and by default it allows all outbound and inbound traffic.
- **Custom Network ACLs:** You can create custom network ACLs. By default, each custom network ACL denies all inbound and outbound traffic until you add rules.
- **Subnet Associations:** Each subnet in your VPC must be associated with a network ACL. If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default network ACL.
- **Block IP Addresses:** Block IP addresses using network ACLs, not security groups.

Network ACLs

- You can associate a network ACL with multiple subnets; however, a subnet can be associated with **only 1 network ACL** at a time. When you associate a network ACL with a subnet, the previous association is **removed**.
- Network ACLs contain a **numbered list of rules** that are evaluated in order, starting with the **lowest** numbered rule.
- Network ACLs have **separate** inbound and outbound rules, and each rule can either **allow or deny traffic**.
- Network ACLs are **stateless**; responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa).

Scenario Question

- As an SA/Cloud Engineer, you have been asked by your organization to configure and deploy a new accounting application with a fault-tolerant design that is highly available and scalable. You meet with the development team and learn that the new application has been designed as an n-tier application architected for failover.

- How will you architect this from a network standpoint to meet your organization's needs?

Answer

- I will deploy the application to a single Amazon Virtual Private Cloud (Amazon VPC) with multiple subnets configured across multiple Availability Zones to meet our organization's needs.

Direct Question

- List the components of Auto Scaling

Answer

- Auto Scaling Launch Configuration- Specifies EC2 instance size and AMI name
- Auto Scaling Group
  - Reference the launch configuration
  - Specifies min, max, and desired size of the Auto Scaling Group
  - May reference an ELB
  - Health check type
- Auto Scaling Policy
  - Specifies how much to scale in or out
  - One or more maybe attached to an ASG.

Scenario Question

- A radio station runs a contest where every day at noon they make an announcement that generates an immediate spike in traffic that requires 8 EC2 instances to process. All other times, the web site requires 2 EC2 instances. What is the most cost-effective way to meet these requirements?

Answer

- **Create an Auto Scaling Group with a minimum capacity of 2 and set a schedule to scale at about 11:40 am.**

Direct Question

- How can an EBS volume that is currently attached to an EC2 instance be migrated from one Availability Zone to another?

Answer

- Create a snapshot of the volume, and create a new volume from the snapshot in the other AZ.

Question

- How can you maximize the durability and availability of Amazon EBS data?

Answer

- To maximize both the durability and availability of Amazon EBS data, we create snapshots of our EBS volumes frequently. Taking snapshots of our EBS volumes increases the durability of the data stored on our EBS volumes. EBS snapshots are incremental, point-in-time backups containing only the data blocks changed since the last snapshot.
- All EBS volume types offer durable snapshot capabilities and are designed for 99.999 percent availability. If our EBS volume does fail, all snapshots of that volume remain intact, and we can recreate our volume from the last snapshot point.
- A snapshot of a volume is available across all of the Availability Zones within a Region. You can use a snapshot to create one or more new EBS volumes in any Availability Zone in the AWS Region.
- The EBS snapshots can also be copied from one AWS Region to another and can be shared with other user accounts. EBS snapshots provide an easy-to-use disk clone or disk image mechanism for backup, sharing, and disaster recovery.

Story Time! (Talking Point)

- To maximize both durability and availability of Amazon EBS data, I have created snapshots of our EBS volumes frequently based on our RTO and RPO. For cases where our EBS volume fails, I have recreated our volume from the last snapshot point.
- The good news is that a snapshot of a volume is available across all of the Availability Zones within a Region. So, I have used snapshots to create one or more new EBS volumes in any Availability Zone in the AWS Region as needed.
- I have also copied EBS snapshots from one AWS Region. In short, I have used EBS snapshots to provide an easy-to-use disk clone or disk image mechanism for backup, sharing, and disaster recovery.

Question

- How do you choosing the correct EBS volume type to use for your workload in your environment?

Answer

- When preparing to make decisions; understanding our workload characteristics is usually a key consideration in the selection process of our EBS volume types.
- If our workload is IOPS- intensive, I usually recommend starting with the SSD volume types and reviewing the performance characteristics. If our workload is more throughput-intensive, I usually recommend starting with HDD volume types to see if their performance can meet our performance requirements.
- If workload requirements exceed the maximum performance characteristics for a selected EBS volume type, we will eliminate the volume type from consideration for that volume. We will then review the characteristics for the next higher performing EBS volume type.
- Next, we will look at the application's latency sensitivity. If it is very low and sub-millisecond to single-digit millisecond latency is needed, io2 Provisioned IOPS could be required. If single-digit to low two-digit latency is tolerable, gp3 General Purpose SSD could be our correct choice. If our workload is not latency sensitive, HDD volume types could be our most cost-effective choice.
- Finally, I will consider whether we prefer to optimize for price or performance. When comparing the EBS volume types, multiple volumes types could satisfy the requirements; so, I will consider the configuration which is more cost-effective.
- From my experience, our volume type decision is usually made at the individual volume level. With EBS Elastic Volumes, we usually make changes as needed to optimize our EBS volumes. This is easy since EBS Elastic Volumes let us change the volume type, dynamically increase the volume size, and modify performance characteristics. For example, for gp3 and io2 volumes types, we have dynamically changed the provisioned IOPS or provisioned throughput performance settings for our volume.
- So, while we make EBS volume selection decisions base on the characteristics of our workload, we use actual test results and our actual performance data to optimize our EBS volume types for price and performance.

Story Time (Talking Point)

- I have been involved in selecting a suitable EBS volume types base on our workload characteristics.
- For example, if our workload is IOPS- intensive, I usually recommend starting with the SSD volume types and reviewing the performance characteristics. If our workload is more throughput-intensive, I usually recommend starting with HDD volume types to see if their performance can meet our performance requirements.
- If workload requirements exceed the maximum performance characteristics for a selected EBS volume type, we will eliminate the volume type from consideration for that volume. We will then review the characteristics for the next higher performing EBS volume type.
- Next, we will look at the application's latency sensitivity. If it is very low and sub-millisecond to single-digit millisecond latency is needed, io2 Provisioned IOPS could be required. If single-digit to low two-digit latency is tolerable, gp3 General Purpose SSD could be our correct choice. If our workload is not latency sensitive, HDD volume types could be our most cost-effective choice.
- Finally, I will consider whether we prefer to optimize for price or performance. When comparing the EBS volume types, multiple volumes types could satisfy the requirements; so, I will consider the configuration which is more cost-effective.
- From my experience, our volume type decision is usually made at the individual volume level. With EBS Elastic Volumes, we usually make changes as needed to optimize our EBS volumes. This is easy since EBS Elastic Volumes let us change the volume type, dynamically increase the volume size, and modify performance characteristics. For gp3 and io2 volumes types, we have dynamically changed the provisioned IOPS or provisioned throughput performance settings for our volume.
- While we make EBS volume selection decisions base on the characteristics of our workload, we usually use actual test results and our actual performance data to optimize our EBS volume types for price and performance.

Question

- What is the required relationship to attach an Amazon EBS volume to an Amazon EC2 instance?

Answer

- The Amazon EC2 instance and the Amazon EBS volume must be in the same AWS Region and in the same Availability Zone.

Question

- Which Amazon EBS volume types support attachment to multiple Amazon EC2 instances at the same time?

Answer

- Only Provisioned IOPS io1 and io2 volume types can be used to attach a single volume to multiple EC2 instances.

Question

- Which AWS tool would you use to estimate the cost of an Amazon EBS volume with configured snapshots before implementation?

Answer

- **AWS pricing calculator** calculates estimated costs for EBS volumes and AWS snapshots. You do not need to implement the EBS volumes to use the tool. You can use the estimates to budget, compare different configuration costs, and apply usage scenarios.

- https://calculator.aws/#/
- https://calculator.aws/#/createCalculator/EBS

Question

What is the main difference between Amazon EC2 instance stores and Amazon EBS volumes?

Answer

Instance stores are ephemeral (temporary),Amazon EBS volumes are persistent

There are several differences between Amazon EC2 instance stores and Amazon EBS volumes. The main difference is that instances stores are ephemeral (temporary) and Amazon EBS volumes are persistent.

Instance stores are deleted when the associated EC2 instance is deleted. Amazon EBS volumes exist separate from the EC2 instance and can be moved to another instance if required. Your data remains intact regardless of the state of the attached EC2 instance.

Question

- Which AWS service monitors configuration and utilization metrics for your Amazon EBS volumes and generates recommendations of which volumes to modify in order to provide the best price-performance trade-off?

Answer

AWS Compute Optimizer.

Compute Optimizer uses Amazon CloudWatch metrics to analyze your EBS volumes and provide recommendations to assist you in optimizing your Amazon Elastic Block Store (Amazon EBS) costs.

Question

- Which AWS service provides statistical data that you can use to view, analyze, and set alarms on the operational behavior of your volumes?

Answer

Amazon CloudWatch

Amazon CloudWatch collects statistical data for us to use to view and analyze our Amazon EBS volumes.

Question

- We have an Engineer who have been doing a fine job of taking weekly snapshots of our EBS volumes. Following our new RPO of 24 hours, we will need to start taking snapshots every 24 hours. The manual process is no longer feasible, and I would like you to recommend a process we can use to automate the creation, retention, and deletion of snapshots that we use to back up our EBS volumes and Amazon EBS-backed Amazon Machine Images (AMIs).

Answer

- We can use Amazon Data Lifecycle Manager (Amazon DLM) to automate the creation, retention, and deletion of snapshots that we use to back up our EBS volumes and Amazon EBS-backed Amazon Machine Images (AMIs).
- With DLM, we can create Lifecycle policies including core settings such as
  - **Target tags** - which specifies the tags that must be assigned to an EBS volume or an Amazon EC2 instance for it to be targeted by the policy.
  - **Schedules** - which includes the start times and intervals for creating snapshots or AMIs. The first snapshot or AMI creation operation starts within one hour after the specified start time.
  - **Retention -** which specifies how snapshots or AMIs are to be retained. You can retain snapshots or AMIs based on their total count (count-based) or their age (age-based)

Story Time! (Talking Point)

- I have created EBS volumes and attached them to relevant EC2 instances making sure the required relationship to attach an Amazon EBS volume to an Amazon EC2 instance is considered.
- Also, I have leveraged AWS pricing calculator to estimate the cost of an Amazon EBS volume with configured snapshots before implementation.
- Further, I have implemented Compute Optimizer to analyze our EBS volumes and provide recommendations to assist us in optimizing our Amazon EBS costs.
- In addition, I have used CloudWatch to get statistical data that we can use to view, analyze, and set alarms on the operational behavior of our volumes.
- Moreover, I have implemented Amazon Data Lifecycle Manager (Amazon DLM) to automate the creation, retention, and deletion of snapshots that we use to back up our EBS volumes and Amazon EBS-backed Amazon Machine Images or AMIs.

Direct Question

- As an SA, could you describe how you would make your VPC resources highly available, scalable, and fault-tolerant.

Answer

- First, I will recommend multiple subnets and keeping our second subnet in a separate Availability Zone for redundancy and fault-tolerance. This spreads out our risk, so if resources in one Availability Zone become unavailable, resources in the second Availability Zone are not affected.
- I will also recommend Auto Scaling. If our application becomes unavailable, with AWS Auto Scaling, I will designate that a new set of resources be launched automatically into our second subnet.
- Our users can automatically be rerouted to the new resources when they're available, using a traffic management option such as Elastic Load Balancing
- I will also implement ELB health checks capability on our infrastructure to automatically fail over connections from unhealthy resources to healthy ones

Summary

- To increase the availability of our Amazon VPC based applications, I will make sure to use strategies such:
  - Use a second subnet, either for more capacity or as a backup option.
  - Deploy your second subnet to a second Availability Zone to maximize availability.
  - Manage traffic between resources using Elastic Load Balancing.
  - Use Elastic Load Balancing to detect unhealthy resources and automatically fail over to healthy ones in your other subnet(s).

Scenario Question

- An Engineer has launched two EBS backed EC2 instances in us-east-1a region. The Engineer wants to change the zone of one of the instances. Assuming you're that Engineer, How can you change it?

Answer

- It is not possible to change the zone of an instance after it is launched

Scenario Question

- You want to create a group of Amazon EC2 instances in an application tier that accepts HTTP traffic only from instances in the web tier (a group of instances in a different subnet sharing a web-tier security group). What design option will help you to achieve this?

Answer

- Associate each instance in the application tier with a security group that allows inbound HTTP traffic from the web-tier security group.

Scenario Question

A company is running multiple applications on Amazon EC2. Each application is deployed and managed by multiple business units. All applications are deployed on a single AWS account but on different virtual private clouds (VPCs). The company uses a separate VPC in the same account for test and development purposes.

Production applications suffered multiple outages when users accidentally stopped/terminated and modified resources that belonged to another business unit. As a Solutions Architect in the team, you have been asked to improve the availability of the company's applications while allowing the Developers access to the resources they need.

What architecture design option will you recommend to satisfy the requirements?

Answer

- I will Implement a tagging policy based on business units. Create an IAM policy so that each user can terminate instances belonging to their own business units only.

- https://docs.aws.amazon.com/IAM/latest/UserGuide/access_tags.html

Story Time-Talking point

I have used tags and IAM policies to control access to AWS resources. In one of my previous engagement, the company was running multiple applications on Amazon EC2 instances. Each application was deployed and managed by multiple business units and owners. At the time, all applications were deployed on a single AWS account, in a single region but on different virtual private clouds (VPCs). The company was using a separate VPC in the same account and region for test and development purposes.

Production applications suffered multiple outages when users accidentally stopped/terminated and modified resources that belonged to another business unit or owner. As an SA in the team, I was asked to improve the availability of the company's applications while allowing the Developers access to the resources they needed; so, I needed to develop an architecture design option that will address the challenges.

I Implemented a tagging policy based on business units. I created an IAM policy so that each user can stop/terminate or modify instances belonging to their own business units only. I added conditions in the identity-based policies that will allow these operations only if the instance tag Owner has the value of the username.

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_tags.html

Scenario Question

- An application runs on EC2 instances in an Auto Scaling Group. The application runs optimally on 9 EC2 instances, and must have at least 6 running instances to maintain minimally acceptable performance for a short period. Recommend the most cost efficient Auto Scaling group configuration that meets the requirements taking into account the possibility of an AZ failure.

Answer

- A desired capacity of 9 instances across 3 Availability Zones.

Scenario Question

- The web tier for an application is running on 6 EC2 instances spread across 2 availability zones behind an ELB Classic Load Balancer. The data tier is a MySQL database running on an EC2 instance. What changes can you suggest to increase the availability of the application?

Answer

- Migrate the MySQL database to a Multi-AZ RDS database instance.
- Launch the web tier EC2 instance in an Auto Scaling Group.

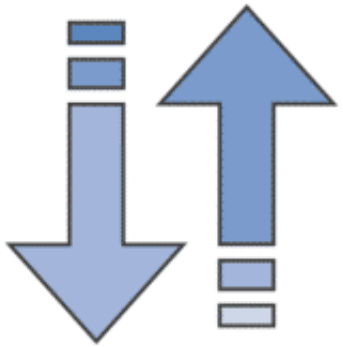
Direct Question

- Explain your strategy for Cost Control in AWS

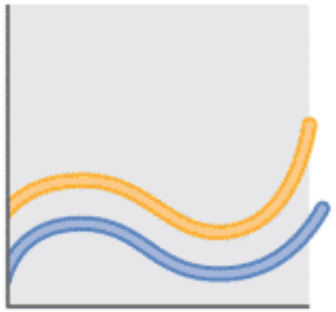
Direct Question

- One of the many differences between an on-premises environment and AWS is that on AWS, customers pay only for what they use. Cost optimization is also making sure customers only pay for what they need, running the exact amount of resources required to get their business done. In that regards, AWS has identified four key best practices of cost optimization. Could you list and explain these best practices

Answer



Right-sizing instances



Increasing elasticity

Choosing the right pricing model

Optimizing storage

Answer Explanation

- **Rightsizing instances**
  Rightsizing is the process of reviewing deployed resources and seeking opportunities to downsize when possible. For example, if an application instance is consistently underutilizing its RAM and CPU, switching that to a smaller instance can offer significant savings while maintaining the same performance.
- **Increasing application elasticity**
  An example is using auto scaling to ensure that the correct number of instances are available to handle the workload of an application. Scale out during high demand and scale in during low demand.
- **Choosing the right pricing model**
  An example is using Reserved Instances for workloads that need to run most or all the time, such as production environments. This can have a significant impact on savings compared to on-demand; in some cases, up to 75 percent.
- **Optimizing storage**
  An example is the S3 Intelligent-Tiering storage class, which is designed to optimize costs by automatically moving data to the most cost-effective storage tier.

Answer

- *Trusted Advisor*
- *AWS Budget Alarm*
- *EC2 Pricing models*
- *Intelligent start and stop of compute resources*

Cost Optimization Story (Talking Point)

- In my current and previous engagements, I have designed solutions that have allowed us to achieve business outcomes at the lowest price point by selecting cost effective resources such as right-sizing instances; managing supply and demand of resources by increasing elasticity with Auto Scaling, creating expenditure awareness by categorizing and tracking our AWS costs with AWS cost explorer; leveraging managed services including using Amazon RDS, DynamoDB and more; acquiring resources using the pricing model that best fits our needs; and optimizing over time by reassessing our existing architectural decisions to ensure they continue to be the most cost effective using tools like Trusted Advisor.

Direct Question

- *What is RTO and RPO?*

Answer

RTO: How long it takes for the systems to recover and come back online again. It's generally measured in seconds minutes hours etc.

RPO is how much data is lost if the system fails. It can be measure in GB etc. It's also commonly measured in units of time like hours.

Question

- *You're running an EC2 instance which uses EBS for storing state. You take an EBS snapshot everyday. When your systems crashes, it takes you 10 minutes to bring it up again from the snapshot. What is your RTO and RPO going to be?*

Answer

Since we are taking a snapshot everyday, it means we will lose at most 1days worth of data, so our RPO is 1day. Since it takes 10 minutes to bring the system back online, our RTO is 10 minutes.

“When you're operating at scale, failure is not an exception event; it needs to be treated as an operational event-

as something that can happen at anytime.”

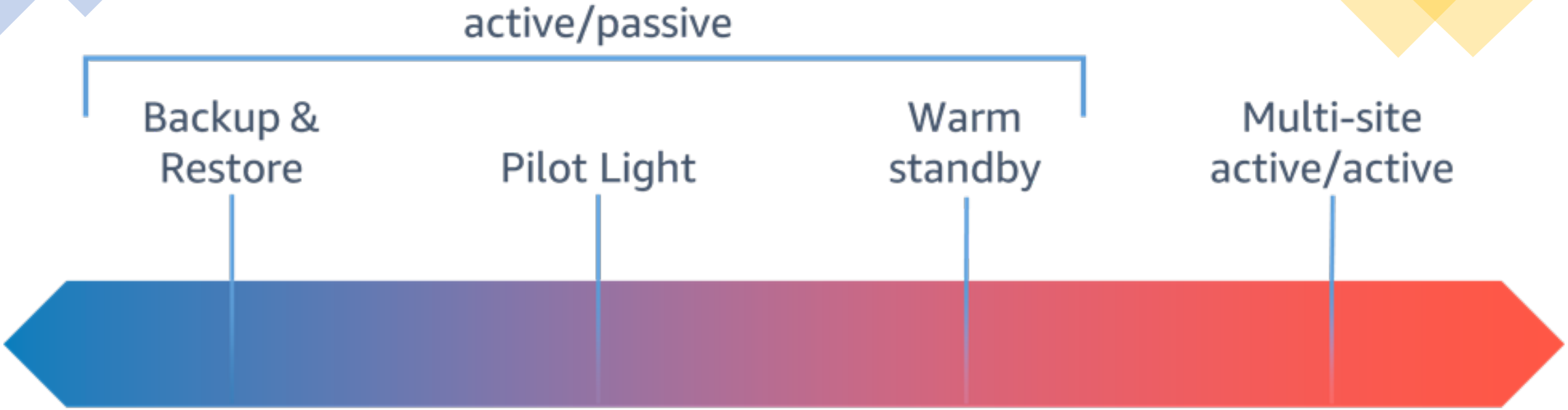
Direct Question

- There can be various disaster recovery designs that we can implement, this directly depends on how quickly we want to recover from a disaster; in short RTO and RPO. List the DR models

Answer

- Broadly classified into 4 types :

1. Backup & Restore
2. Pilot Light
3. Warm Standby
4. Multi-Site



RPO / RTO: Hours

- Lower priority use cases
- Provision all AWS resources after event
- Restore backups after event
- Cost $

RPO / RTO: 10s of minutes

- Data live
- Services idle
- Provision some AWS resources and scale after event
- Cost: $$

RPO / RTO: Minutes

- Always running, but smaller
- Business critical
- Scale AWS resources after event
- Cost $$$

RPO / RTO: Real-time

- Zero downtime
- Near zero data loss
- Mission Critical Services
- Cost $$$$

- *List EC2 instance pricing options*

Answer

On-demand -pay by the hour or second depending on the type of instance you run.

Spot- Purchase unused capacity at a discount of up to 90%. Prices fluctuate with supply and demand. Great for applications with flexibility start and end times

Reserved- Reserved capacity for 1 or 3 years. Up to 72% discount on the hourly charge. Great if you have known, fixed requirements.

Dedicated- A physical EC2 server dedicated for your use. Great if you have server-bound licenses to reuse or compliance requirements.

Direct Question

- *What is the template that Auto Scaling uses to launch a fully configured instance automatically?*

Answer

- **Launch Configuration**

Direct Question

What is the difference between high availability and fault-tolerant?

Answer

High Availability means the system is up and available but might perform in a degraded state.

Fault Tolerate means the user does not experience any impact from fault.

Answer

- High Availability aims for your application to run 99.999% of the time. Its design ensures that the entire system can quickly recover if one of its components crashed. It has an ample number of redundant resources to allow failover to another resource if the other one fails. This concept accepts that a failure will occur but provides a way for the system to recover fast.
- Fault Tolerance, on the other hand, has the goal of keeping your application running with zero downtime. It has a more complex design, and higher redundancy to sustain any fault in one of its components. Think of it as an upgraded version of High Availability. As its name implies, it can *tolerate* any component fault to avoid any performance impact, data loss, or system crashes by having redundant resources beyond what is typically needed. The caveat for implementing a fault-tolerant system is its cost as companies have to shoulder the capital and operating expenses for running its required numerous resources.
- For example, an application requires 6 EC2 instances to handle the expected load. A system can be considered highly available as long as it has several EC2 instances running in 2 different AZs. On the other hand, a fault-tolerant architecture is the one that still has 6 running instances, even if one AZ goes down. It could mean that the application has a total of 9 running instances on a regular basis, where there are 3 running instances on 3 different AZs. So, if one AZ experienced an outage, there are still 6 instances running that can handle the load.

Direct Question

- Why would you want to spin up an EC2 Dedicated Host?

Answer

- Because your application has hardware-specific licensing requirements
- Dedicated Hosts are designed to solve hardware-specific licensing requirements.

Direct Question

- You've been tasked with creating a highly performant database on your EC2 instance. Which type of EBS volume will support the high level of IOPS that you require?

Answer

- Provisioned IOPS

Scenario Question

- Your company has been running its core application on a fleet of r4.xlarge EC2 instances for a year. You are confident that the application has a steady-state performance and now you have been asked to purchase Reserved Instances (RIs) for a further 2 years to cover the existing EC2 instances, with the option of moving to other Memory or Compute optimised instance families when they are introduced. You also need to have the option of moving Regions in the future. Which of the following options meet the above criteria whilst offering the greatest flexibility and maintaining the best value for money.

Answer

- Purchase a 1-year Convertible RI for each EC2 instance, for 2 consecutive years running

Direct Question

- What is the difference between Multi-AZ & Read Replica

Multi-AZ

- An exact copy of your production database in another Availability Zone.
- Used for disaster recovery.
- In the event of a failure, RDS will automatically failover to the standby instance.

VS

Read Replica

- A read-only copy of your primary database in the same Availability Zone, cross-AZ, or cross-region.
- Used to increase or scale read performance.
- Great for read-heavy workloads and takes the load off your primary database for read-only workloads (e.g., Business Intelligence reporting jobs).

Direct Question

- Tell me about your experience on AWS security

Discuss Security

Answer

- I have extensive experience on security including infrastructure protection, Identity and Access Management, compliance, Data protection, detective controls, and incidence response.
- For infrastructure protection, I have implemented AWS Network Firewalls, AWS Shield, AWS WAF, & AWS Firewall Manager. Regarding IAM, I have used AWS IAM, AWS Single Sign-on, AWS Directory Service and Amazon Cognito. On data protection, I have leveraged AWS KMS, AWS Certificate Manager, AWS Secrets Manager and Amazon Macie. Also, I have used Amazon Guard Duty, Amazon Inspector, AWS CloudTrail, AWS Security Hub and AWS Config for Detection. On compliance, I have leverage AWS Artifact and AWS Audit Manager.
- In one of my previous assignment, the company had multiple accounts but was using configurations that did not meet some security governance policies among which included:
- Preventing ingress from port 22 to any Amazon EC2 instance.
- Providing the Required billing and application tags for resources.
- Encrypting all Amazon EBS volumes and more
- As a result, I needed to provide suitable preventive and detective controls, including notifications about specific resources if there were policy deviations.
- To address these requirements, I used AWS Service Catalog to build a portfolio with products that are in compliance with the governance policies in a central account. I restricted users across in all accounts to AWS Service Catalog products and shared a compliant portfolio to other accounts. Also, I used AWS Config managed rules to detect deviations from the policies; and configured an Amazon CloudWatch Events rule to send a notification to an SNS topic when a deviation occurs.

Direct Question

- What is the difference between high availability and fault-tolerant?

Answer

High Availability means the system is up and available but might perform in a degraded state.

Fault Tolerate means the user does not experience any impact from fault.

Answer

- High Availability aims for your application to run 99.999% of the time. Its design ensures that the entire system can quickly recover if one of its components crashed. It has an ample number of redundant resources to allow failover to another resource if the other one fails. This concept accepts that a failure will occur but provides a way for the system to recover fast.
- Fault Tolerance, on the other hand, has the goal of keeping your application running with zero downtime. It has a more complex design, and higher redundancy to sustain any fault in one of its components. Think of it as an upgraded version of High Availability. As its name implies, it can *tolerate* any component fault to avoid any performance impact, data loss, or system crashes by having redundant resources beyond what is typically needed. The caveat for implementing a fault-tolerant system is its cost as companies have to shoulder the capital and operating expenses for running its required numerous resources.
- For example, an application requires 6 EC2 instances to handle the expected load. A system can be considered highly available as long as it has several EC2 instances running in 2 different AZs. On the other hand, a fault-tolerant architecture is the one that still has 6 running instances, even if one AZ goes down. It could mean that the application has a total of 9 running instances on a regular basis, where there are 3 running instances on 3 different AZs. So, if one AZ experienced an outage, there are still 6 instances running that can handle the load.

Direct Question

- Tell me about your experience on architecting for fault-tolerant and high availability.

Answer

- I have implemented a highly available and fault-tolerant architecture using various services, features, and techniques in AWS. I have ensured high availability by deploying our application to multiple Availability Zones or several AWS Regions. I have also used Auto Scaling to dynamically scale our systems depending on incoming demand or traffic and implemented an active-active or active-passive failover policy in Route 53 to reduce downtime. I have also leveraged Amazon RDS Automated Snapshots, Read Replica, and Multi-AZ Deployments to strengthen our database tier to remove a single point of failure in our system. In some use cases, I have opted to use the Amazon Aurora Global database or DynamoDB Global tables for our globally-accessible applications. In addition, I have leveraged on the self-healing capabilities of AWS services to achieve fault-tolerance and high availability.

Question

- Your web service has a performance SLA to respond to 99% of requests in <1 second. Under normal and heavy operations, distributing requests over four instances meets this SLA. As an SA, recommend an architecture that will ensure cost efficient fault-tolerant operations of your service if an Availability Zone becomes unreachable

Answer

- Deploy the service on eight servers across two Availability Zones

Question

- Your web service has a performance SLA to respond to 99% of requests in <1 second. Under normal and heavy operations, distributing requests over four instances meets this SLA.
- As an SA, recommend an architecture that will ensure cost efficient high availability of your service if an Availability Zone becomes unreachable

Scenario Question

As a Solutions Architect you're designing a highly available and reliable solution for a cluster of Amazon EC2 instances.

Following the requirements, you must ensure that any EC2 instance within the cluster recovers automatically after a system failure. The solution must ensure that the recovered instance maintains the same IP address.

How can these requirements be met?

Answer

Create an Amazon CloudWatch alarm for the StatusCheckFailed_System metric, and then configure an EC2 action to recover the instance.

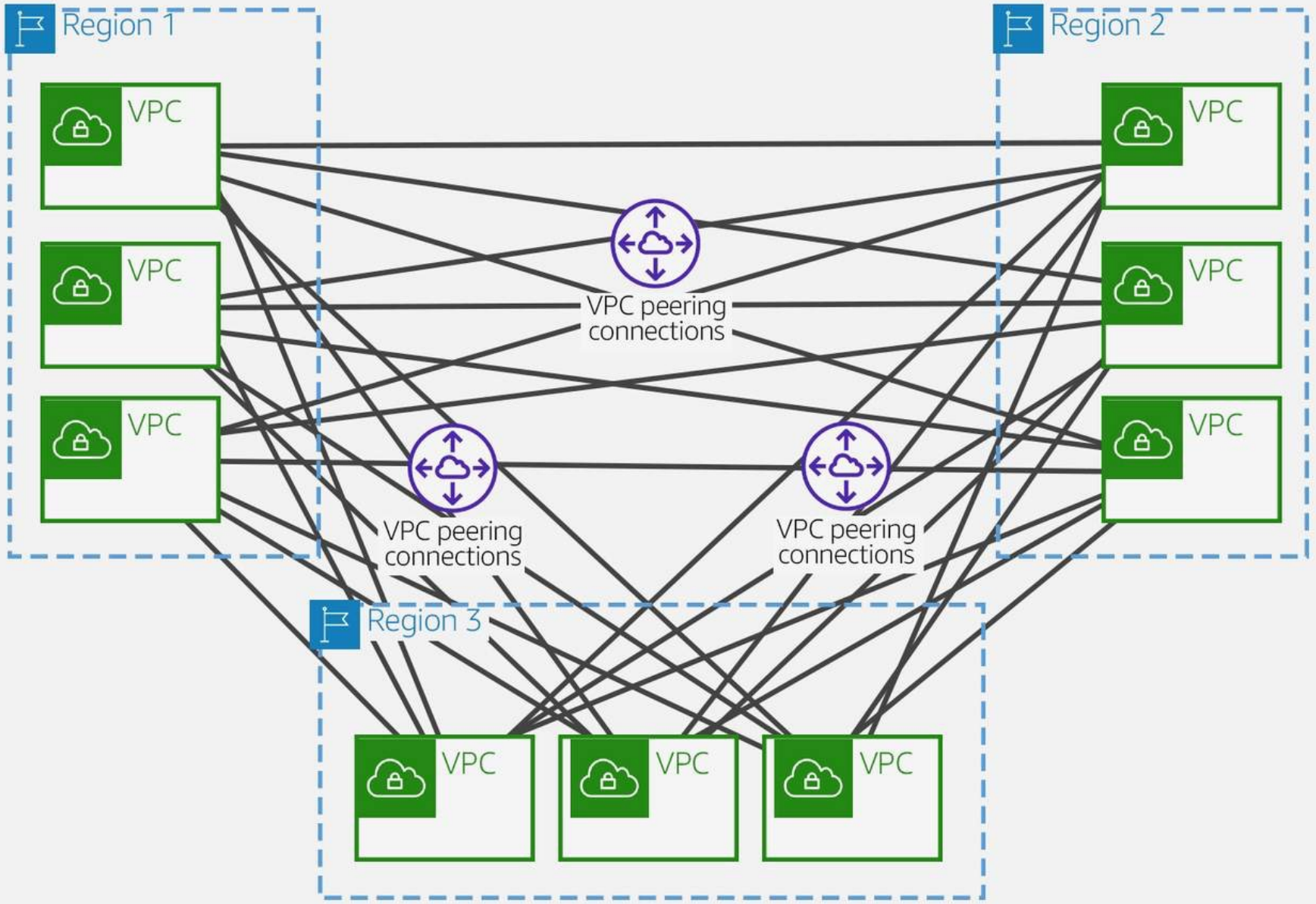
AWS have taken care of this now for us

Question

- AWS offers two types of peering connections for routing traffic between VPCs in different Regions: VPC peering and transit gateway peering. What are some benefits of TGW over VPC peering?

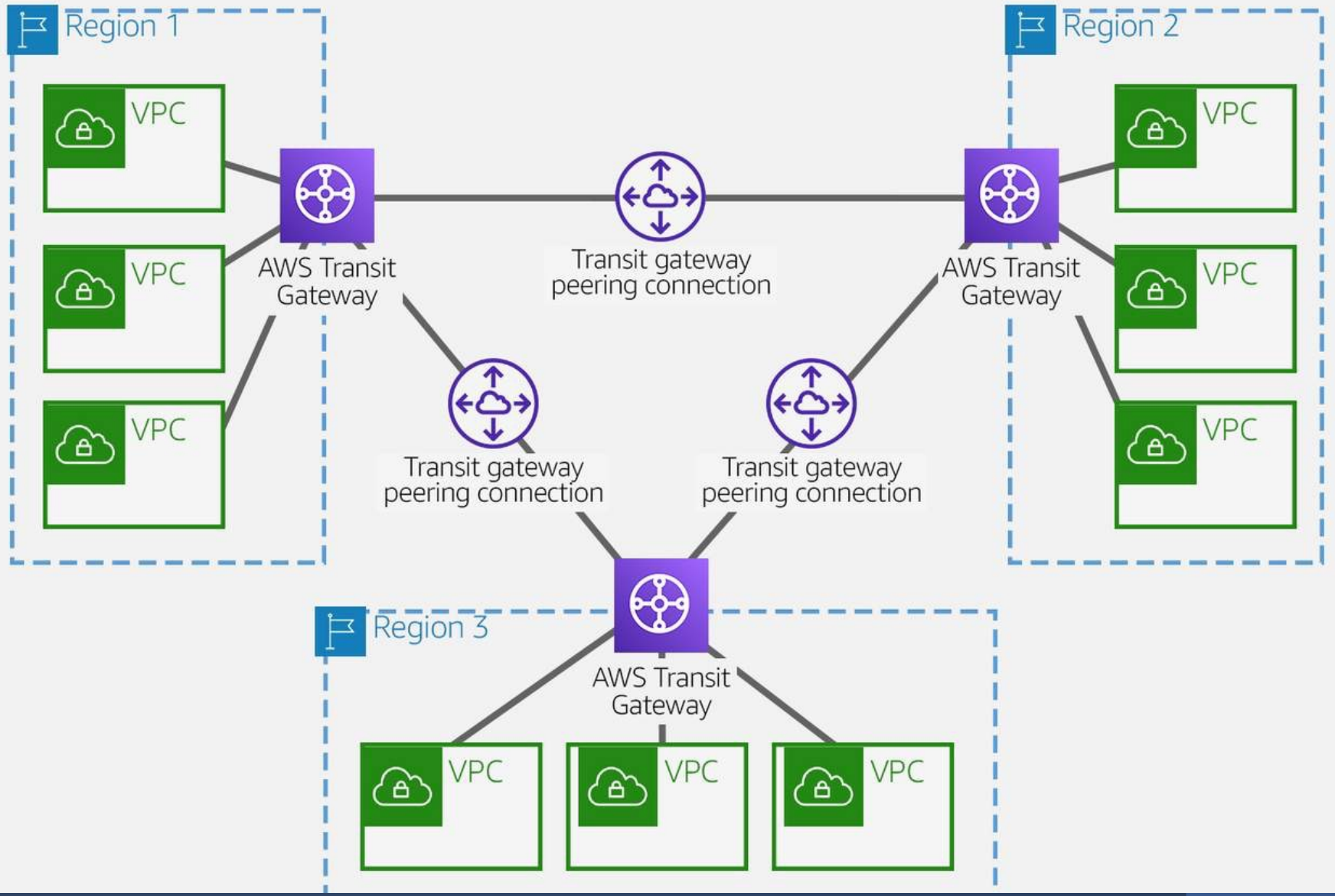
Answer

- Both peering types are one-to-one, but transit gateway peering connections have a simpler network design and more consolidated management.
- One customer in my previous project had multiple VPCs about 9 in three different Regions. To permit network traffic to route between each VPC would have required us creating 72 VPC peering connections. Each VPC would have needed 8 different routing configurations and security policies.


Region 1
VPC
VPC
VPC
Region 2
VPC
VPC
VPC
VPC peering connections
VPC peering connections
VPC peering connections
Region 3
VPC
VPC
VPC

Answer Continues

- With AWS Transit Gateway, the same environment only needed three peering connections. The transit gateway in each Region facilitated routing network traffic to all the VPCs in its Region. Because all routing can be managed by the transit gateway, the customer only needed to maintain three routing configurations, thereby simplifying management.


Region 1
VPC
VPC
VPC
AWS Transit Gateway
Transit gateway peering connection
Region 2
VPC
VPC
VPC
AWS Transit Gateway
Transit gateway peering connection
Transit gateway peering connection
Region 3
AWS Transit Gateway
VPC
VPC
VPC

Question

- Tell me about your experience on Networking (VPC)

Answer

- I have developed and implemented various customer solution architectures leveraging VPC and associated services and features including subnets, route tables, NAT Gateways, VPC peering connections, Transit Gateways, VPC Endpoints, Security Groups, NACLs and more. For example, I have created public facing subnets for our web resources with access to the internet and place our backend systems such as database and application servers in a protected private subnet. I have also implemented multiple layers of security including security groups with firewall rules and NACLs to control access to EC2 instances.
- The number of VPCs and how to divide resources among them have been a strategic design point that I usually consider. I have used VPC to group resources in many ways either by region, applications, line of business, and compliance requirements depending on an individual customer's requirements like the scope of the customer's environment and the number of distinct environments that the customer needs. For example, I have designed and created separate VPCs for Development, Testing and Production. Another use case was where we needed to separate production data from backups. I designed and created a separate AWS account with it's own VPC and replicated data into resources in that VPC. That way, our data was isolated even if the production account happen to be compromised and data is destroyed.
- For some customers who are unable to make a complete move to AWS and who as a result have some on-premises applications that can't be moved at the moment or can never be moved due to architectural or compliance reasons, I have implemented various networking solutions to connect the customer's existing data center to the AWS cloud.
- The easiest choice is to connect across the public internet which is sometimes appropriate for some of our use cases. However, it's not secured, and the performance of the link is usually not consistent. To secure the link, I have implemented an IPSec VPN. While a VPN is usually excellent in securing the link, it traverses the internet, making its bandwidth to be limited and the performance fluctuate. For scenarios where my customer require guaranteed bandwidth and high security, I usually consider and recommend AWS Direct Connect.

End here

Only continue if asked to give an implementation experience.

- I have been fortunate to be in a team where Direct connect implementation decisions where being made. One of the first and possibly most influential decisions we had to make was the form we wanted our connection to take. AWS provides its customers with three choices: A dedicated connection collocated at a Direct Connect location; Contracting with a Direct Connect Partner or Connecting directly to a Direct Connect node. Because Direct Connect relies on a physical connection to our AWS environment, we also needed to choose the bandwidth we wanted that connection to support from three types of Ethernet connections that AWS support i.e. either one gigabit per second or 10 gigabits per second or 100 gigabits per second. In that specific case we used a single 10-gigabit connection. As we discussed options regarding public and private virtual interface, the next choice we needed to make was the Ethernet frame size that our circuit will support. Direct Connect virtual interfaces support a default Ethernet frame size of 1522 bytes and a jumbo Ethernet frame size of 9023 bytes. We needed to ensure that all the equipment we will use to connect our on-premises location to our AWS environment supports the Ethernet frame size we wanted to implement. For that specific case, we used frame size 1522. After considering all the prerequisite options for preparing our connection to Direct Connect, we were ready to create a standalone connection in our AWS account

Question

Tell me about your experience on VPC

Answer

- I have developed different network architecture solutions for various environments depending on the customer's requirements including designing our VPC implementation based on our expansion requirements, looking ahead at least two years. When designing our Amazon VPC, I consider the number of IP addresses required and the connectivity type with the data center before choosing the CIDR block within the permissible size of the block ranges- between a /16 netmask and a /28 netmask.
- In my design, I usually isolating our VPC environments such as using separate Amazon VPCs for development, production, and staging environments. If we do decide to keep all of those environments in one Amazon VPC, I make sure we provide as much isolation as possible, using network ACLs, subnets, and other Amazon VPC resources.
- From an implementation standpoint, I have created both public and private subnets to accommodate resources that are internet facing and resources that are expected to be isolated from the internet. I have also leveraged NAT Gateways to make it easy for resources within a private subnet to connect to the Internet but prevents the internet from initiating a connection with those resources.
- For use cases where resources in one of our VPC's needs to route traffic privately to resources in another VPC, I have created and configured VPC Peeing.
- For enterprise setups, I have configured Transit Gateway which acts a central hub for multiple VPCs and on-premises network.
- For environments where my customer needs to connect their existing data center with the AWS cloud, I have implemented an IPSec VPN to secure the link between on-premises network and our AWS VPC. For use cases where guaranteed bandwidth and high security are required, I have used AWS Direct Connect and selected either the 1 gigabytes or 10 gigabytes or 100 gigabytes dedicated option.
- Where resources in our VPC needs to connect to resources outside of the VPC, I have recommended and configured VPC endpoints.
- As part of security around our network, I configured SGs and only opens the ports that are specifically needed while making sure protocols like SSH and RDP security group rules are only open to specific networks. Where we explicitly need to deny a particular source network from traversing our network space such as in an event of DDOS, I've used NACL to achieve this.
- To log IP information that traverses our VPC, I enabled VPC flow logs to deliver logs to either a designated CloudWatch logs or S3 bucket.
    - You can end here!
    - If you want to shine, continue below.

Answer continues

- **Situation:** When I joined one my previous employers, their dev and prod resources were hosted and running in a single VPC in AWS. In addition, almost all their databases were public facing. Even their firewalls were not configured properly as they were overly open. Also, each tier of their application was hosted in a single AZ even for prod resources.
- So, I recommended to my manager for us to move the development workloads to a different AWS account. That will enable us group the workloads based on business purpose and data classification, promote innovation, limit the scope of impact from adverse events and to enhance automation. I also advised that for production workloads, that we move databases and other resources to private subnets for security reasons.
- **Task(Target)**:Following my recommendations to my manager and subsequent discussions with leadership during our leadership monthly updates, I was asked to develop a new architecture for both prod, DR and lower environment workloads.
- **Actions**: In developing my architecture, I moved all resources that do not need to be accessed directly from the internet to private subnets. In the application, only the web hosts were in public subnets. I placed the application and RDS instances in private subnets. To achieve application availability, I made sure every tie used 2 availability zones. Another major concern I raised was on security groups and I addressed that in my architecture solution in several ways:
- The web hosts will allow internet traffic on port 80 and 443 for web requests.
- Application hosts will only allow application port traffic connections from the web hosts
- The RDS instances will only allow database connections from port 3306 from the application hosts
- **Results**: After implementing the new architecture, I remember specifically how management and the entire team celebrated the subsequent CGA reports.

Question

Tell me about your experience on VPC

Answer

- I have developed network designs for Multi-tier architectures, Multi-VPC architectures and Hybrid networks using various AWS networking service offerings such as VPC endpoints, VPC peering, AWS Direct Connect, AWS Site-to-Site VPN, AWS Transit Gateway.
- When I joined one of my previous employer, there was a need to standup multiple environments for various application migrations. As one of the SAs on the team at the time, I needed to develop SDDs that captures the current state architecture, functional and non-functional requirements and the end-state architecture including network architecture diagrams and other infrastructure components.
- In developing my network architecture, I previewed both public and private subnets, and placed web resources such as web instances in public subnets and application and database instances in private subnets. To enable instances in our private subnets to connect to the internet, I previewed a NAT gateway in my network architecture. To ensure application resources in our VPC were able to connect to our existing monitoring and security tools that were hosted in another VPC to protect the company's proprietary business data and processes, I added a VPC peering functionality in my architecture design. On network firewalls, I previewed SGs and NACLs.
- When I completed the SDD, it went through our Architecture Review Council –approval process and was and was signed-off with minimal or no architectural adjustments. After getting all the required authorizations for the SDD, I worked with the Cloud Engineer on the project to develop the LLD and eventually built out the environments –dev, stage, prod & DR. As the various environments were sequentially handed over to the business team, everyone was happy and that is how I got moved to other migration projects within the company.

Question

- Tell me about your experience on VPC

Answer

- I have designed and implemented various customer solution architectures leveraging VPC and associated services and features including subnets, route tables, NAT Gateways, VPC peering connections, Transit Gateways, VPC Endpoints, Security Groups, NACLs and more. For example, I have created public facing subnets for our web resources with access to the internet and place our backend systems such as database and application servers in a protected private subnet. I have also implemented multiple layers of security including security groups with firewall rules and NACLs to control access to EC2 instances.
- In my current assignment, our organization has company services distributed across four VPCs and a single VPC dedicated to centralized IT services and logging. To facilitate data sharing, we have constructed a fully mesh network design using VPC peering to connect each VPC to every other VPC in the organization. Each VPC have a one-to-one connection with each VPC it is approved to communicate with. This is because each VPC peering connection is nontransitive in nature and does not allow network traffic to pass from one peering connection to another. For example, VPC 1 is peered with VPC 2, and VPC 2 is peered with VPC 4. We cannot route packets from VPC 1 to VPC 4 through VPC 2. To route packets directly between VPC 1 and VPC 4, we have created a separate VPC peering connection between them.
- In another assignment, our company was implementing a centralized logging and analytics service for the Amazon Web Services (AWS) hosted systems. Over time, the company's AWS environment had grown into an environment with eight virtual private clouds (VPCs) across three Regions. According projections at the time, this is likely the largest number of VPCs necessary for the business going forward. Two solutions were under consideration for this new service. The first solution proposes linking each VPC within a Region to a dedicated transit gateway. Each transit gateway would be connected to an isolated VPC that hosts only the new logging and analytics service. The other solution proposes the use of VPC peering to connect each of the eight VPCs to an isolated VPC hosting the new logging and analytics service. We needed to adopt a solution that will provide the best balance of cost and performance; so, I recommended VPC Peering. I determined that VPC peering connection offers the best balance of cost and performance base on the requirements. The first solution would have required three transit gateways that will peer with the new service VPC. Each transit gateway would have incurred a VPC connection and data transmission charge. On the other hand, VPC peering connections only incur a data transmission charge. VPC peering connections, unlike transit gateways, have no aggregate bandwidth restriction. Lastly, deploying a transit gateway adds a hop between the data source and the data destination.

Answer Continues

- Also, in one my previous engagements, I was working for a large regional charity that uses Amazon Web Services (AWS) to host all the organization's systems and services. The organization expanded their reach at the time by merging with three smaller charities located in a nearby city. All three charities also used AWS to host their systems and services. As the mergers were finalized, we quickly integrated services by establishing peering connections with the other charities' virtual private clouds (VPCs).
- When the merger was complete, the offices of the three charities in the nearby city needed to be consolidated into a single office as a cost-savings measure. The IT director had asked me to use the new office as an opportunity to simplify the new organization's network and centralize management as much as possible. To achieve this, I recommended AWS Transit Gateway TO route all the organization's traffic to and from each VPC through a centrally managed service. I also leverage AWS Transit Gateway Network Manager to get a single global view of our private network. I was able to define the resources we wanted to monitor on the AWS Transit Gateway Network Manager dashboard.

Question

Tell me about your experience on EC2

Answer

- I have extensive experience on EC2 and associated features including Amazon EC2 Instance Types, EC2 Cost Model or purchasing options, Auto scaling, load balancing, EC2 security, EBS volume types and back and recovery of the EC2 persistent storage.
- I remember when I joined one of my previous employers, we had a workshop and there was a colleague who sat next to me, and I noticed she spent most of her time during the workshop taking snapshots of EBS volumes. So, during our lunch break while we were networking, I approached her to check what the kind of project she was currently working. During our conversation she was manually snapshotting over 300 EBS volumes every week. I used that opportunity to introduced Data Lifecycle Manager to her to automate the snapshotting and retention of snapshots based on their RTO and RPO. I recall the excitement when she realized how seamless DLM was taking snapshots of their volumes without any manual intervention.
- Also, when I joined Ventech, they were considering moving from stand-alone EC2 instances to an Auto Scaling group managed instances which can automatically adjust the amount of compute capacity to accommodate circumstances such as changes in demand for their application. This was critical to them because they understood that less manual intervention will lead to lower costs and their compute costs were significant exceed the budget at the time. As a result, I was tasked to lead this effort based on my experience on AS. To address this, I leveraged Dynamic Auto Scaling policy. In the configuration, I created a CloudWatch alarm based on performance information for our EC2 instances or a load balancer. Whenever a performance threshold was breached, a CloudWatch alarm will trigger the Auto Scaling event which will either scale out or scale in EC2 instances in the environment.
- One concern that the developers had in the AS solution was the possibility of an EC2 instance premature being registered to our load balancer. To address their concern, I recommended and configured an Auto Scaling feature called lifecycle hooks which will keep the instance in a pending wait state for custom actions to complete before the instance enters “In-Service”. Also, to ensure that AS was removing instances from service with care, I leverage connection draining which will ensure the load balancer allows existing, in-flight requests made to an instance to complete, and will not send any new requests to the instance that is marked for deregistration.
- After implementing the solution and all those associated features, Auto Scaling was able to manage our workload dynamically and we were able to focus on other issues and functionalities.

Question

Tell me about your experience on EC2

Response

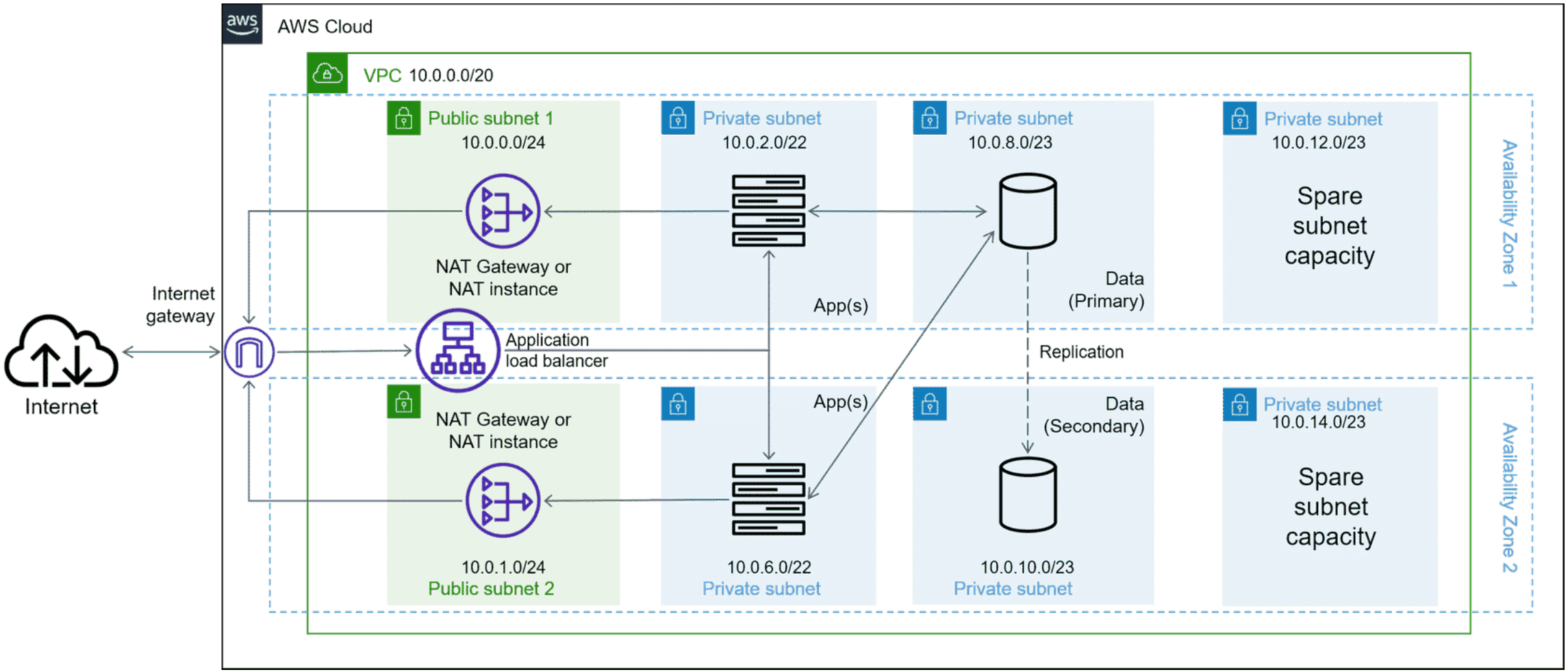
- I have extensive experience on EC2 including recommending the right instance type that is optimized for the characteristics of our workload; I also leveraged auto scaling to maintain application availability and to dynamically scale our Amazon EC2 capacity up or down automatically, according to conditions that we defined. I have also optimized cost by recommending and using the appropriate instances and resources for our systems; taking advantage of flexible and cost-effective pricing options to acquire EC2 instances in a way that best fits our needs, managing the demand and supply with the right amount of resources using auto scaling.

Question

- Tell me about your experience designing a 3-Tier Architecture

Answer

- I was a solution architect for a company that was targeting to migrate an e-commerce web application which was developed in Java and was monolithic. It was a 2-tier solution, meaning a separate web and database tier. The web front end and application were deployed to a pair of load balanced web/app servers. They were running in a public subnet accessed by customers through a firewall with port 80 and 443 open. The database was MySQL running on a single server with no redundancy. It was located in a private subnet behind a firewall which only allowed traffic on port 3306 from the web/app server via a hard-coded IP address rule. The functionality was good; but it wasn't performing well. During certain hours, it was really slow to respond to users. Also, they were looking to have a low touch environment; at least from a day-to-day perspective, that will allow them to focus their energy on implementing new functionality.
- After I reviewed the customer's current architecture, and with a solid understanding of the customer's challenges, I had a pretty good idea of which services and features would address those pain points, so I developed an amazing solution to present to the customer – something that covers all the bases: security, availability, cost-optimization and so on. *The solution I proposed supported AWS best practices aligned to the well architected framework.*
- I evolved the original 2-tier to a 3-tier application with web, app and DB tier, and the database functionality was migrated to Amazon RDS using MySQL-engine. This greatly reduced the management overhead of the database and provided automatic failover and backup to S3.
- To address the scaling concerns, I made both the web and application tier to use elastic load balancing and auto-scaling. Instances were to be added and removed based upon the load at every given time.
- Additionally, S3 was deployed to host static content, which was to be served via Amazon CloudFront for a more robust customer experience across geographical locations.
- Application availability was achieved not only through autoscaling, but by having every tie use 2 availability zones.
- Another major concern was security. I addressed this in several ways.
- In the application, only the web hosts were in public subnet. The application and RDS instances are were placed in private subnets
- Security groups were configured with best practices into consideration. For example:
    - The web hosts will allow internet traffic on port 80 and 443 for web requests.
    - Application hosts will only allow application port traffic connections from the web proxy
    - The RDS instances will only allow database connections from port 3306 from the application hosts
    - Bastion hosts will allow management connection only from JJ Tech Inc internal network. And all of the other hosts will only allow management connections to the bastion host.
- Implemented Network Access Control Lists (NACL) rules to filter traffic into subnets as an additional layer of network security
- S3 buckets were implemented with security features enabled.
- Standard IAM policies with associated groups and roles configured along the principles of least privilege.
- In addition, monitoring and logging; alerts and notifications for critical events were configured.
- The new architecture was scalable, performant efficient, reliable, secured and cost optimized, and the customer was very happy.


AWS Cloud
VPC 10.0.0.0/20
Public subnet 1
10.0.0.0/24
Private subnet
10.0.2.0/22
Private subnet
10.0.8.0/23
Private subnet
10.0.12.0/23
Spare subnet capacity
Availability Zone 1
NAT Gateway or NAT instance
App(s)
Data (Primary)
Internet gateway
Internet
Application load balancer
Replication
NAT Gateway or NAT instance
App(s)
Data (Secondary)
Private subnet
10.0.14.0/23
Spare subnet capacity
Availability Zone 2
10.0.1.0/24
Public subnet 2
10.0.6.0/22
Private subnet
10.0.10.0/23
Private subnet

Scenario Question

- You work for a specialty retail organization. They are building out their AWS VPC for running a few applications. They store sensitive customer information in two different encrypted S3 buckets. The applications running in the VPC access, store and process sensitive customer information by reading from and writing to both the S3 buckets. The company is also using a hybrid approach and has several workloads running on-premises. The on-premises datacenter is connected to their AWS VPC using Direct Connect. You have proposed that an S3 VPC Endpoint be created to access the two S3 buckets from the VPC so that sensitive customer data is not exposed to the internet. What type of VPC endpoint would you recommend?

Answer

- Interface Endpoint

Scenario Question

A company operating a website on AWS requires high levels of scalability, availability, and performance. The company is running a Ruby on Rails application on Amazon EC2.

It has a data tier on MySQL 5.6 on Amazon EC2 using 16 TB of Amazon EBS storage Amazon CloudFront is used to cache application content.

The Operations team is reporting continuous and unexpected growth of EBS volumes assigned to the MySQL database. The Solutions Architect has been asked to design a highly scalable, highly available, and high-performing solution.

Which solution is the MOST cost-effective at scale?

Answer

- Ensure that EC2 instances are right-sized and behind an Elastic Load Balancing load balancer. Implement Auto Scaling with EC2 instances. Ensure that the reserved instances are purchased for fixed capacity and that Auto Scaling instances run on demand. Migrate to an Amazon Aurora MySQL Multi-AZ cluster. Ensure that multi-AZ architectures are implemented.

Scenario Question

- For your production web farm, you have configured an auto scaling group behind a Network Load Balancer. Your auto-scaling group is defined to have a core number of reserved instances and to scale with spot instances. Because of differences in spot pricing across AZs, sometimes you end up with many more instances in one AZ over another. During times of peak load, you notice that AZs with fewer instances are averaging 70% CPU utilization while the AZ with more instances average barely above 10% CPU utilization. What is the most likely cause of this behavior?

Answer

- Cross-zone load balancing is disabled on the Network Load Balancer.