AWS Interview Questions

AWS Interview Questions and Answers
Prepared by Lana Begunova for Educational Purposes

About the Presentation

The presentation walks you through some of the most popular questions faced in an AWS interview.

Cloud computing has become the norm among enterprises that want more flexibility, greater efficiencies, lower costs, and improved disaster recovery.

AWS is the dominant cloud service provider, holding 40% of the market share. From $3.1 billion in 2013 to over $62 billion in 2021, the revenue growth of AWS has been steady, to say the least.

If you're moving into the field as an AWS Solution Architect Associate and preparing for an AWS Solution Architect job interview, then this is the presentation for you.

1. Define and explain the 3 basic types of cloud services and the AWS products that are built based on them.

AWS has three basic types of cloud services:

- Compute
- Storage - store your data somewhere
- Networking - connect other services to your application

These basic services exclude monitoring, analytics, because they are considered optional/advanced.
Can choose a non-cloud service/product for monitoring and analytics, so they are not considered as basic.
Basic services are compute, storage and networking.

Compute domain helps in the following aspects:

- To run any application
- Control and manage server functions such as scaling and deployment
- Run event-initiated stateless applications (i.e. Lambda)

- EC2 - the main Compute product, a major share of the Compute resource
- Elastic Beanstalk - PaaS (Platform as a Service)
- Lambda - FaaS (Function as a Service)
- Auto Scaling
- Lightsail

Networking domain helps in the following aspects:

- To control and manage the connectivity requirements for various AWS services
- You can select your own IP address range as well as accelerate the delivery of your content

The products that are built based on the Storage service:

- VPC - can't imagine working without VPC in the cloud environment, especially in AWS cloud environment.
- CloudFront - edge caching service which helps customers read the application with low latency.
- Route53 - domain resolution, DNS

2. What is the relation between Availability Zone and Region?

- AWS Region is a separate geographic area.
  E.g., us-west-1 (North California), asia-south (Mumbai)

- Each AWS Region has multiple isolated locations knows as Availability Zones.
  All availability zones inside one Region are isolated from one another in terms of failure.

Some of the services will replicate themselves within the Availability Zone. AZs can replicate within AZs.
Regions do not typically replicate within themselves.

3. What is auto scaling?

- Auto Scaling in AWS allows you to configure and automatically provision and launch new instances whenever the demand increases/decreases.
- Auto Scaling allows automatic adjustment (increase or decrease) of resource capacity as per the needs.
- Businesses need not worry continuously about managing the capacity of resources.
- Auto Scaling is one of the most appealing reasons to choose AWS.

4. What is geo targeting and how do you setup geo targeting in CloudFront?

CloudFront is caching and it caches content globally in the Amazon caching servers worldwide. The whole point is to provide users worldwide with access to the data from the nearest possible server. Based on language or what's popular in the ares, you can customize the content.

- Geo targeting is a concept where you can show personalised content to your audience based on their geographic location without changing the URL.

  The URL stays the same, we can change the content partially, not entirely, otherwise it would be dynamic. We can make small changes to the content - file, picture, or a particular link - in a website and show customize content to users who are in different parts of the globe.

- AWS allows you to send customised content through CloudFront (*based on what's popular or in demand*).

- Amazon CloudFront connects with other members of the AWS family of services to deliver content to end users at high speed and with low latency.

- CloudFront detects the country where your viewers are located and forwards the country code to your origin server so that you can personalise content for that viewer without changing the URL. *Once the server receives the specific country code, it changes the content and sends it to the caching server, where it gets cached forever. The user gets to view the content, which is customised for them, for the country they are in.*

- CloudFront detects the user's country and passes their their country code to you in the `CloudFront-Viewer-Country` header.

5. What are the steps involved in a CloudFormation solution?

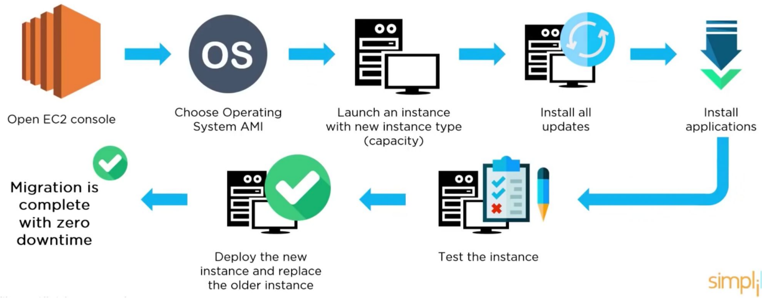
You can back up an environment within CloudFormation template. If there is a template, you can simply run it, and it provisions the environment. But there is a lot more that goes into it.

Steps to moving towards infrastructure as a code:

1. Create or use existing CloudFormation template using JSON/YAML format.
2. Save your code template locally or in S3 bucket (*as a repository for your code*).
3. Use AWS CloudFormation to create a stack on your template.
    - *Call the CloudFormation, call the file in the S3 bucket and create a stack. Now CloudFormation uses/reads the file, understands services that are being called, understands the order and how they are connected to each other.*
    - *CloudFormation is an intelligent service. It understands the relationships among different services, based on the code. It sets an order for itself and then provisions the services one after another.*
    - *Let's say Service B has a dependency on Service A, and the dependent service has to be provisioned. CloudFormation is intelligent enough to provision resource A first and then only resource B. If you invert the order, resource B gets provisioned first, and because it does not have a dependency the chances are that CloudFormation is intelligent enough to roll back if something is not healthy or not provisioned properly. Firstly, CloudFormation provisions the resources that are depended on by other services and only then provisions the dependent services.*
4. AWS CloudFormation constructs and configures your stack resources that you've specified in your template.

6. How do you upgrade or downgrade a system with near zero downtime?

We can upgrade or downgrade a system with near zero downtime using the following steps of migration:

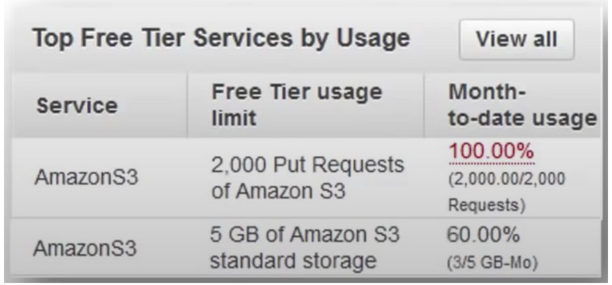

Open EC2 console
OS
Choose Operating System AMI
Launch an instance with new instance type (capacity)
Install all updates
Install applications
Test the instance
Deploy the new instance and replace the older instance
Migration is complete with zero downtime
simpli

- Everyone is moving towards zero downtime or near-zero downtime and wants to have their application highly available. You can upgrade an EC2 instance to a better EC2 instance by changing the instance type, stopping and starting. But stopping and starting causes a downtime. This is a wrong answer.
- Upgrading a system with zero downtime includes launching another system parallelly, with a bigger EC2 instance type with a bigger capacity and install all that's needed. If using an AMI from the old machine well, you don't have to go through installing all the updates and installing all the application from the AMI. Once you launched a bigger instance locally, test the application to verify it's working. Don't put it in production yet. If the application works as expected, you can swap your server behind Route53. All you have to do is go to Route 53, update the information with the new IP address of the new server and that's going to send traffic to the new server now. So, the cutover is handled. Or, if you're using static IP you can remove the static IP from the old machine and assign it to the new machine. Or, if you are using a new Elastic card, you can remove the new card from the old machine and attach the new card to the new machine. That way we can get near-zero downtime.

7. What are all the tools and techniques you can use in AWS to identify and correct if you are paying more than you should be?

1. Top Free Tier Services Table

- This is a dashboard of the Billing and Cost Management console.
- This table shows the free tier usage limit for your top five most-used free tier services.



Top Free Tier Services by Usage

Service	Free Tier usage limit	Month-to-date usage
AmazonS3	2,000 Put Requests of Amazon S3	100.00% (2,000.00/2,000 Requests)
AmazonS3	5 GB of Amazon S3 standard storage	60.00% (3/5 GB-Mo)

How do you get the visibility of your AWS resources running? One way is to check the billing. There's a place where you can check the top services being utilized. Can be free or paid services. It's the dashboard of the cost management console. That table shows the top 5 most used services. By looking at it, you can detect you're using a lot of storage, a lot of EC2, why is storage high? You can find out and justify that. If you are storing things that you should be storing and then clean it up. Why is compute capacity so high? Why is data transfer so high? When you think on that level, you are able to dig in and clean up unnecessary resources and save on your bill.

2. Cost Explorer

- This allows you to view and analyse costs.
- You can view costs for the last 13 months.
- You can also get cost forecast for the coming 3 months.

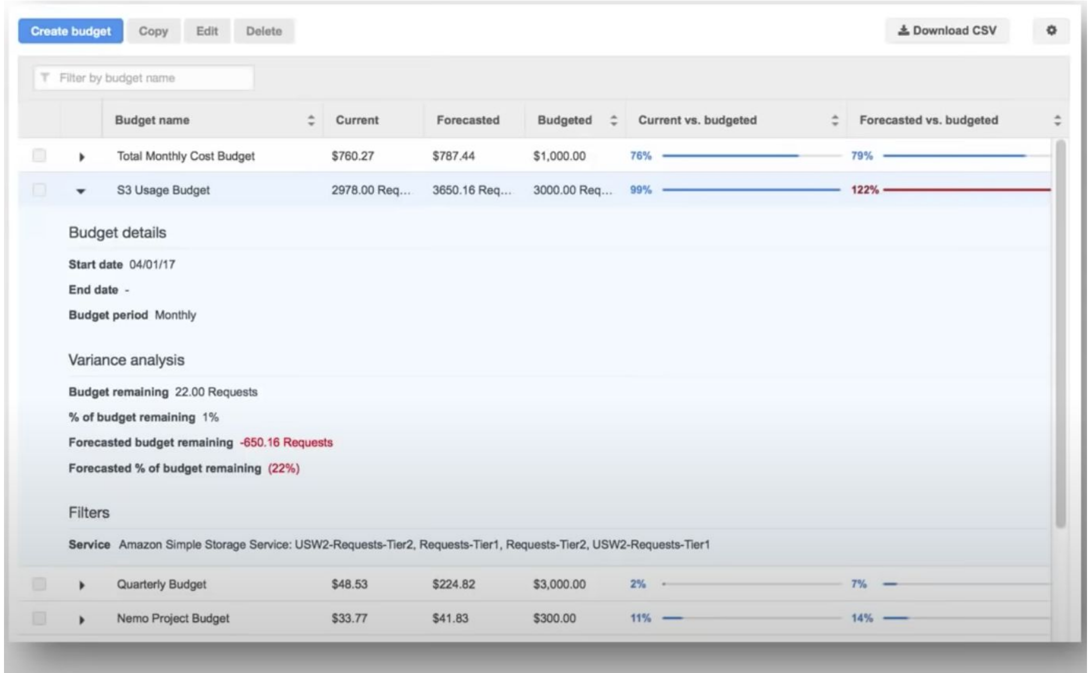
How much will we be using is the pattern is like this. That will help and will give you visibility on how much you have spent, how much you will be spending if the trend continues.

3. AWS Budgets

- Here you can plan your service usage, service costs and instance reservations.
- You can view the following:
    - Is your current plan meeting your budget?
    - Usage details

Budgets is another excellent way to control the costs. You can set up a budget and indicate how much you are willing to spend for this application, this team, etc, this month, this particular resource, etc. You can put a budget mark, and any time it nears the mark, you get an alarm stating you're about to reach the allocated budget amount. You know how big the bill is going to be for that month, you can take steps to control bill amount for that particular month.

AWS Budgets Dashboard Screenshot



Budget name	Current	Forecasted	Budgeted	Current vs. budgeted	Forecasted vs. budgeted
Total Monthly Cost Budget	$760.27	$787.44	$1,000.00	76%	79%
S3 Usage Budget	2978.00 Req...	3650.16 Req...	3000.00 Req...	99%	122%
Quarterly Budget	$48.53	$224.82	$3,000.00	2%	7%
Nemo Project Budget	$33.77	$41.83	$300.00	11%	14%

Budget details

Start date 04/01/17

End date -

Budget period Monthly

Variance analysis

Budget remaining 22.00 Requests

% of budget remaining 1%

Forecasted budget remaining -650.16 Requests

Forecasted % of budget remaining (22%)

Filters

Service Amazon Simple Storage Service: USW2-Requests-Tier2, Requests-Tier1, Requests-Tier2, USW2-Requests-Tier1

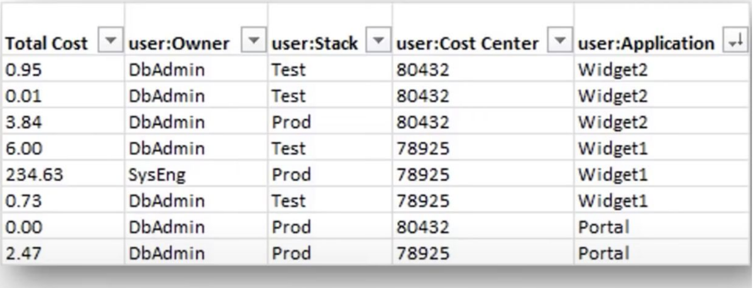
Here is a screenshot of the AWS Budgets Dashboard

4. Cost Allocation Tags

- You can assign a label to every AWS resource.
- Each tag has a *key* and a *value*.
- You can organize your resources and cost allocation tags to keep track of your AWS costs..

Helps in identifying which team or resource has spent more in that particular month. Instead of looking at the bill as one list with no specifics in it, and looking at it as an expenditure list, you can break it down and tag the expenditure to the teams with cost allocation tags. The dev team has spent this much, the production team has spent this much, the training team has spent more than the dev and production teams. Why is that? You'll be able to thin along those levels only if you have cost allocation tags.

These are the tags you put when you create a resource. For production, you create a production tag and you associate those resources to it. At a later point, when you pull up your bill, it's going to show the detail of who is the owner, which group, and how much they have used in the last month. And you can move forward with your investigation and encourage or stop users from using more services.with the help of cost allocation tags.



Total Cost	user:Owner	user:Stack	user:Cost Center	user:Application
0.95	DbAdmin	Test	80432	Widget2
0.01	DbAdmin	Test	80432	Widget2
3.84	DbAdmin	Prod	80432	Widget2
6.00	DbAdmin	Test	78925	Widget1
234.63	SysEng	Prod	78925	Widget1
0.73	DbAdmin	Test	78925	Widget1
0.00	DbAdmin	Prod	80432	Portal
2.47	DbAdmin	Prod	78925	Portal

Here is a screenshot of the Cost Allocation Tags

8. Are there any other alternative tools to log into the Cloud environment other than Console?

In other words, other than GUI, how would you use the AWS resource and how familiar you are with those tools and technologies.

1
Putty
2
AWS
CLI
AWS CLI for Linux
3
AWS
CLI
AWS CLI for Windows
4
AWS
CLI
C:\>cmd
AWS CLI for Windows CMD
5
SDK
AWS SDK
6
eclipse
Eclipse

You can configure PuTTY to access the AWS resources, such as log into EC2 instance. EC2 instance does not always has to be logged into through the console. You can use PuTTY to login like a jump box, proxy machine or a gateway machine adn from there you can access the rest of the resources.

Can install AWS CLI for Linux, Windows, and Mac and from there, from the local machine, you can access, run AWS commands and access, provision, monitor AWS resources.

Can access AWS programmatically using SDK and Eclipse.

9. What services can be used to create a centralized logging solution?

- Log management helps organizations to track a relationship between operational, security and change management events.
- It also helps you to understand the infrastructure.
- We can create a centralized logging solution using the following:

Amazon CloudWatch Logs

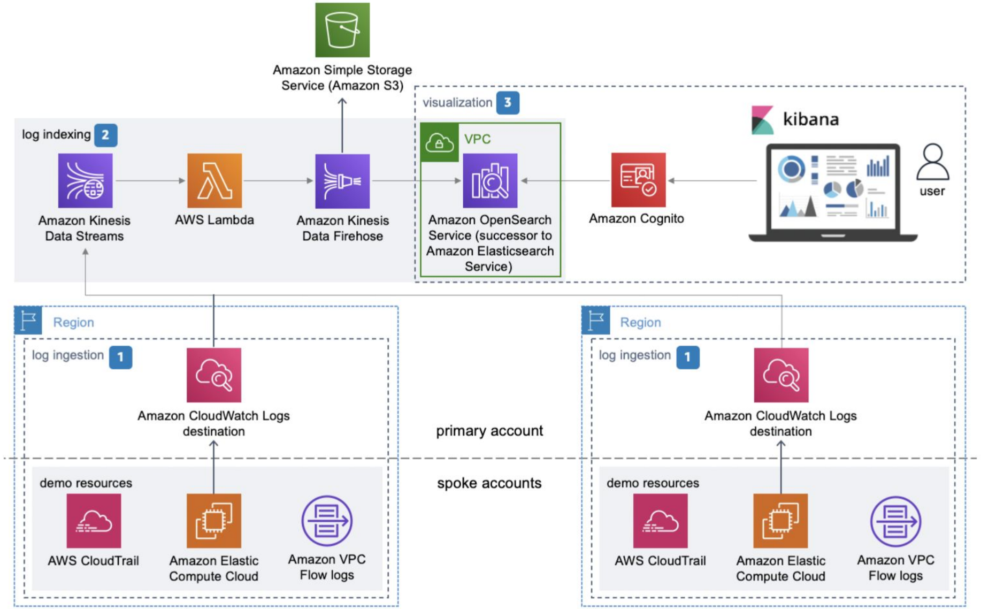
Amazon Kinesis Firehose

Amazon S3

Amazon Elastic Search

You may come across this question, if you work for a security company or a client who focuses more on security and want to use AWS native services.
Can use CloudWatch Logs and store them in S3 and then use Elastic Search to visualize them.
Use Kinesis to move the data from S3 to Elastic Search.
Log management helps organizations to track the relationship between operational and security changes the the event that got triggered based on those logs. Instead of logging into an instance/environment and checking the resources physically, you can come to a fair conclusion by merely looking at the logs. Every time there is a change, the system will screen and it gets tracked in the CloudWatch. Then CloudWatch pushes it to S3. Kinesis pushes data from S3 to Elastic Search. You can do a time-based filter and get a fair understanding of what's been going on in the environment in the past hour or whatever time window you want to look at. Helps with understanding the infrastructure as a whole. All the infrastructure logs get saved in one place, so it's easy for me to look at from an instractructure perspective.

Here's a diagram showing the centralized logging architecture you can deploy.


Amazon Simple Storage Service (Amazon S3)
log indexing 2
Amazon Kinesis Data Streams
AWS Lambda
Amazon Kinesis Data Firehose
visualization 3
VPC
Amazon OpenSearch Service (successor to Amazon Elasticsearch Service)
Amazon Cognito
kibana
user
Region
log ingestion 1
Amazon CloudWatch Logs destination
primary account
spoke accounts
demo resources
AWS CloudTrail
Amazon Elastic Compute Cloud
Amazon VPC Flow logs
Region
log ingestion 1
Amazon CloudWatch Logs destination
demo resources
AWS CloudTrail
Amazon Elastic Compute Cloud
Amazon VPC Flow logs

https://aws.amazon.com/solutions/implementations/centralized-logging

The diagram shows some of the services and how they connect to each other. It could be logs that belong to one account or multiple accounts. Doesn't matter, those three services are going to work fairly good. They are going to inject or suck logs from the other accounts, put them in one place and help us monitor.
You have CloudWatch here which tracks the metrics.
You can also use Cloud Trail if your want to log API calls as well, and push them to an S3 bucket.
There are different types of logs. Flow logs, instance application logs are getting captured from the same or different VPC account, from the same or different accounts. And all of them are analyzed with Elastic Search, using the Kibana client.

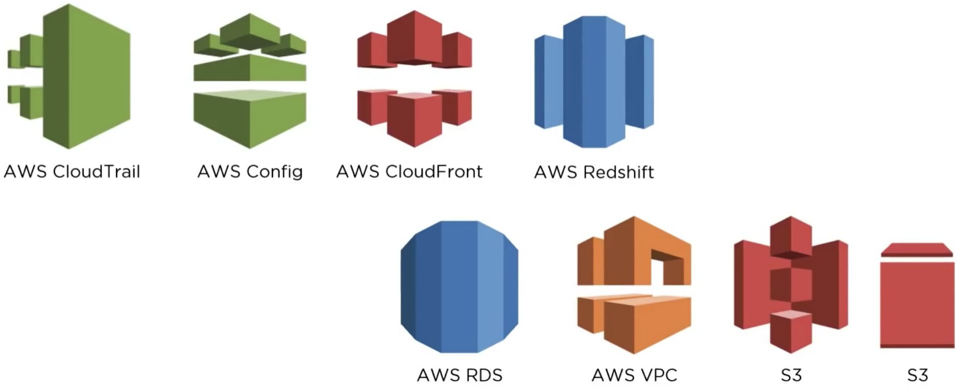
- Step 1 - primary template deploys an Amazon Elastic Search (ES) domain along with two AZs of VPC network.
- Step 2 - two instances with proxy serve as an additional layer of security to restrict access to Amazon ES dashboard.
- Step 3 - a custom Lambda function is used to load the data from CloudWatch to an ES domain.
- Step 4 - only those user requests from approved IP addresses will be allowed access to the Kibana UI using customer-defined credentials.

Step 1 deploys the ECS cluster.
Step 2 restricts access to the ECS cluster, because it's valid data. You don't want anyone to put their hands and access their data. So, restrict access to the ECS dashboard.
Step 3 - you can also use lambda to push the data from CloudWatch to the Elastic Search domain. Kibana is the graphical tool that helps us to visualize the logs instead of looking at logs as statements or a bunch of characters or files. Kibana helps is analyze the logs in a graphical, chart or bar diagram format.

10. What are the native AWS Security logging capabilities?

This question tests your knowledge of AWS security products., especially logging, monitoring, event/incident management.

- **Every service in AWS provides metrics or lodg files to provide insight on how that service is operating.**
- **The following provide the AWS service-specific log recommendations:**



Most services have in-built logging. S3, CloudFront, EBS, VPC have their own logging.

In addition, there are account level logins, such as CloudTrail and AWS config services.

There is a variety of logging options available in the AWS - CloudTrail, config, CloudFront, RedShift logging, RDS logging, VPC flow logs, S3 object logging, S3 access logging, etc.

We are going to look at two services in specifcL: CloudTrail and Config Service

- **AWS CLoud Trail provides a history of AWS API calls for every account.**
- **You can perform security analysis, resource change tracking and compliance auditing of your AWS environment.**
- **It delivers log files to a designated S3 bucket every 5 minutes.**
- **It can be configured to send notifications via AWS SNS when new logs are delivered.**

The CloudTrail provides a high level history of the API calls for all the accounts. With that we can actually perform a very good security analysis of an account. These logs are delivered to you, you can configure it to be delivered to S3 for long-time archivals, for example. Based in a particular event, it can also send an email notification to you saying, "Hey, just got this error, though I'd let you know."

- AWS Config provides an AWS inventory which includes configuration history, configuration change notification and relationships between AWS resources.
- It provides a timeline of resource configuration change for specific services.
- Ir records the cumulative changes if many changes are made within a short period of time.
- It can also be configured to send notifications via AWS SNS when new logs are delivered.

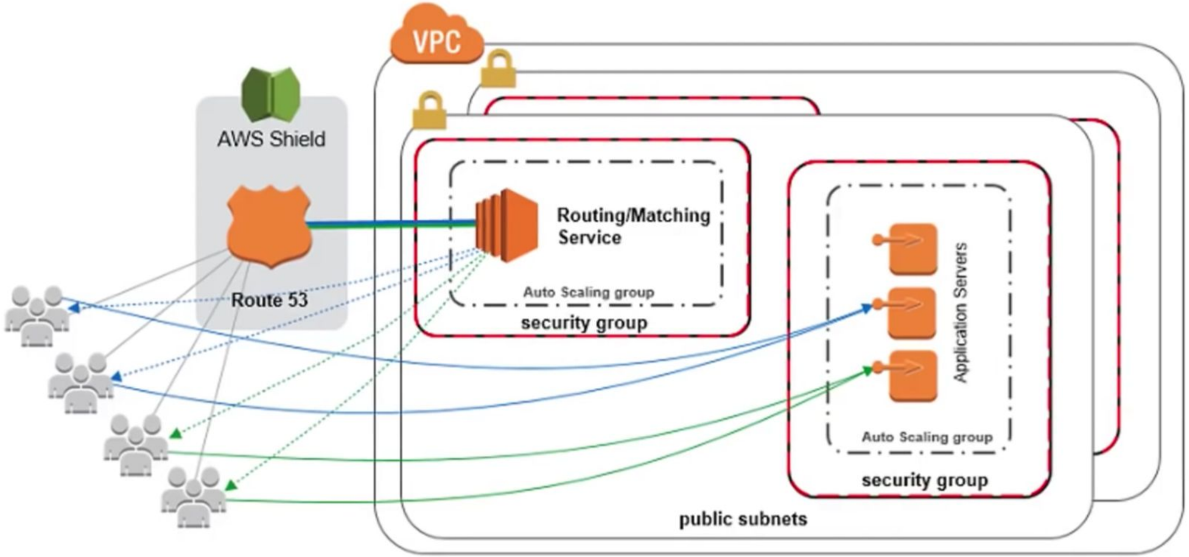
Config service helps us understand the configuration changes that happened in our environment. You can also set up notifications based on the configuration changes. It records the cumulative changes that are made in a short period of time. If you want to go through the lifetime of a particular resource, what are the things that happened, what are the things it went through? - they can be looked at using AWS Config.

11. What is DDoS attack and what services can minimize DDoS attacks?

If your role includes taking care of the cloud security as well, then the other question you can be asked is the native services that Amazon provides to mitigate DDoS. Not all companies would go with Amazon native services. But there are some companies what want to stick with the AWS native services just to save them from the headache of managing the other softwares or bringing in another third-party tool into managing DDoS, they simply want to stick with the proprietary native Amazon services. A lot of companies use Amazon service to prevent DDoS (Denial of Service). Denial of Service is a user trying to or maliciously making attempt to access a website or an application. The user would create multiple sessions and occupy all the sessions and they would not legitimate users access the servers. They are in turn denying the service for the user.

These users, instead of making one connection, are making multiple connections. There are cheap software programs available that would trigger connections from different computers in the internet with different Mac addresses so everything looks legitimate for the server and it would accept those connections and it would keep the sessions open. The actual users won't be able to use them. That is denying the service for the actual users - Denial of Service.
Distributed Denial of Service is generating attacks from multiple places, from a distributed environment.

We can minimize DDoS attacks using the below architecture where a TCP or UDP based application



Source: https://aws.amazon.com/answers/networking/aws-ddos-attack-mitigation/

A DDoS attack is an attempt to male a website or an application unavailable to other genuine end users. This is achieved by hackers using various methods that completely consume a network and its resources.

We can minimize DDoS attacks using the following services:

AWS Shield
AWS WAF
Amazon Route53
CloudFront
ELB
VPC

The native tools that help us to prevent the denial of service attacks in AWS are Cloud Shield and WAF (web access firewall). They are the major ones, they are designed to mitigate a denial of service. If your website is often bothered by denial of service then you should be using AWS Shield or WAF.

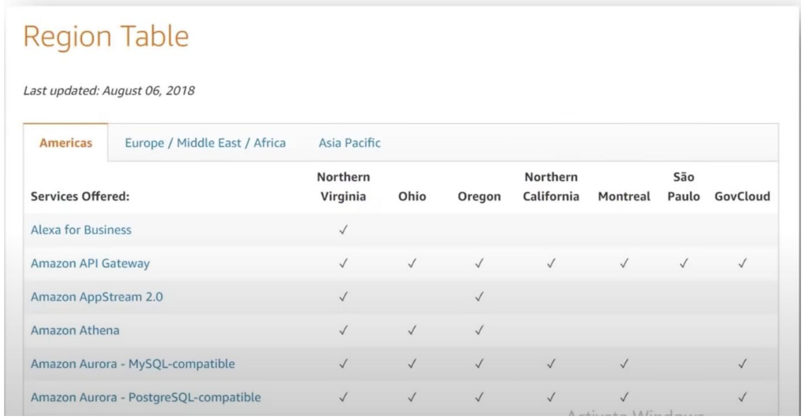
There are a couple of other tools that also handle denial of service. Denial of service is not their primary job, but you could use them for it. Route53's purpose is to provide DNS. CloudFront is to provide caching. Elastic Load Balancer (ELB) works to provide load balancing. VPC is to create and secure a virtual private environment. They also support mitigating denial of service but not to the extent you get in AWS Shield and AWS WAF. Shield and WAF are the primary ones, but the rest can also be used to mitigate distributed denial of service.

12. You are trying to provision a service in a particular region but you are not seeing the service in that region? Why? How to fix it?

This is a tricky question that tests your familiarity with the region and the services available in the region.
When you are trying to provision a service in a particular region, you're not seeing the service in that region, How do we go about fixing it, it how do we go about using the service in the cloud? Not all services are available in all regions.

- As of now, not all services are available in all regions. This is because of the high infrastructure and maintenance costs;

- Here is a snippet of the available regions for various services.



Region Table

Last updated: August 06, 2018

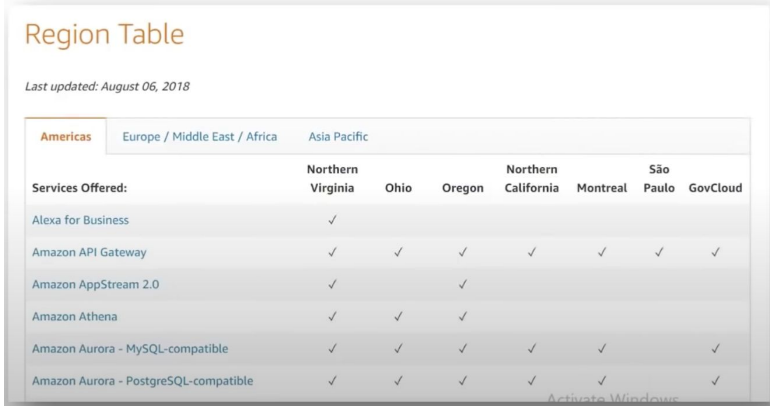
Americas | Europe / Middle East / Africa | Asia Pacific

Services Offered:	Northern Virginia	Ohio	Oregon	Northern California	Montreal	São Paulo	GovCloud
Alexa for Business	✓						
Amazon API Gateway	✓	✓	✓	✓	✓	✓	✓
Amazon AppStream 2.0	✓		✓				
Amazon Athena	✓	✓	✓				
Amazon Aurora - MySQL-compatible	✓	✓	✓	✓	✓		✓
Amazon Aurora - PostgreSQL-compatible	✓	✓	✓	✓	✓		✓

Any time Amazon announces a new service, they don't immediately publish them in all regions. They start small. When the traffic increases and when it becomes more likeable to the customers, they move the service to different regions.
Within Americas, North Virginia has the most services compared to Ohio or NorCal. Within North America itself, North Virginia is the preferred one.
Similarly, there are preferred regions within Europe, Middle East and Africa, and preferred regions within Asia Pacific.
Any time we don't see a service in a particular region, chances are that the service is not available in that region. Yet we have to check the documentation and find the nearest region that offers that service and start using the service from that region.

If you are looking for a service in Asia, e.g., Mumbai and it's not available, why not simply switch to North Virginia and start using it? You could, but that's going to add more latency to your application. That's why we need to check for application, which is check for region which is very near to the place you want to serve your customers, and find the nearest region instead of always going back to North Virginia and deploying an application there.
There is a link in AWS that you can go to and look for services available in different regions https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/.

You can go around this problem by switching your service to another region where its support is available.



Region Table

Last updated: August 06, 2018

Services Offered:	Northern Virginia	Ohio	Oregon	Northern California	Montreal	São Paulo	GovCloud
Alexa for Business	✓						
Amazon API Gateway	✓	✓	✓	✓	✓	✓	✓
Amazon AppStream 2.0	✓		✓				
Amazon Athena	✓	✓	✓				
Amazon Aurora - MySQL-compatible	✓	✓	✓	✓	✓		✓
Amazon Aurora - PostgreSQL-compatible	✓	✓	✓	✓	✓		✓

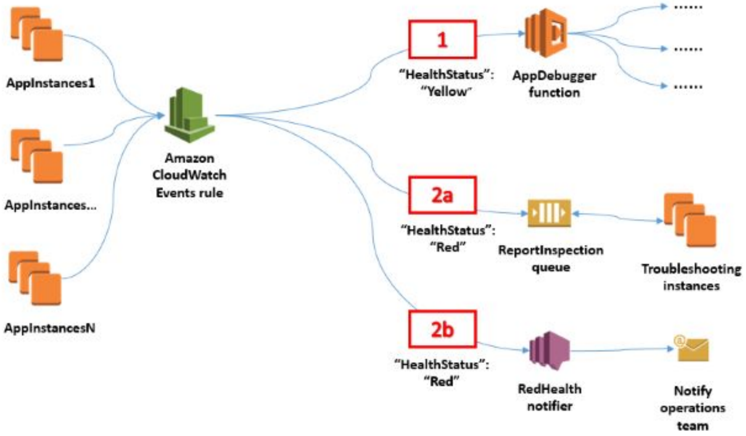
13. How do you setup a system to monitor website metrics in real-time in AWS?

With the emergence of the cloud, a lot of companies have turned down their monitoring team. Instead they want to go with the monitorings that the cloud provides. Not many people want to go through the hassle of at least new startups and new companies that are thinking of having a monitoring environment. They don't want to go with traditional knock monitoring. Instead they would like to leverage the AWS monitorings available. AWS monitors a lot of stuff, not just the availability but it also monitors failures, errors, triggers emails, etc.

- **CloudWatch events helps us to monitor application status of various AWS services and custom events.**
- **Using CloudWatch we can monitor:**
  - **State changes in Amazon EC2**
  - **Auto-scaling lifecycle events -** ***any time there are services added or a reduction in the number of servers because of decreased usage. Very informative messages can be received through CloudWatch.***
  - **Scheduled events -** ***if you want to schedule anything, CloudWatch has an event that would schedule an action, a trigger, time-based not incident-based.***
  - **AWS API calls**
  - **Console sign-in events**

Any time you have a question about monitoring, CloudWatch should come to mind because CloudWatch is meant for monitoring, collecting metrics, providing graphical representation of what's going on in a particular network at a particular point of time.

CloudWatch integrates well with a lot of other services, such as notifications for notifying the use or administrator about it. It can integrate well with Lambda. To trigger an action, any time you're designing an auto healing environment, the CloudWatch can monitor and send an email if we're integrating it with SNS (Simple Notification Service). Or, the CloudWatch can monitor and, based on what's happening, it can trigger an event in Lambda. And that would in turn run the function until the environment comes back to normal. CloudWatch integrates well with a lot of other AWS services.



- CloudWatch has a conditional statement that maps an incoming event to its target.
- A target is a resource such as Lambda or SNS.

- If HealthStatus is Yellow, a Lambda function is invoked called SampleAppDebugger.
- If HealthStatus is Red, it is published either to Amazon SQS or SNS.

CloudWatch has 3 statuses:
-Green - everything's going well.
-Yellow - the service is degraded. When Yellow, it's calling a Lambda function to debug and fix the application.
-Red - the service is not available. It immediately notifies the owner of the application about the service being down and provides the report with the metrics collected about the service.

Source: https://aws.amazon.com/blogs/security/how-to-use-amazon-cloudwatch-events-to-monitor-application-health/

14. What are the various types of virtualization in AWS and what are the differences among them?

You may be tested with questions like the various types of virtualization in AWS and differences among them, if your job role requires you to manage the servers as well. There are certain job roles on the system side which include development plus the system side, and you're responsible for the application as well as the server. Describing the 3 types is the answer for describing the differences among them.

Hardware Virtual Machine (HVM)

- Fully virtualized set of hardware - *all hardware is virtualized and all VMs (virtual machines) act separate from each other.*
- They boot by executing a master boot record in the root block device of your image.

Paravirtualization (PV)

- PV-GRUB us a special boot loader which boots Paravirtual AMIs.
- This PV-GRUB chain loads the kernel specified in the menu.

Paravirtualization on HVM (PV on HVM)

- Paravirtual drivers on HVM help operating systems leverage advantages in storage and network I/O.
- *The marriage between HVM and PV.*

15. Name some AWS services which are not region specific.

You've been taught that all services are within a region and some services are within an AZ.
E.g., EC2, EBS are within an AZ.
S3, DynamoDB are region-specific.
VPC is both AZ and region-specific, meaning subnets are AZ-specific and VPCs are region-specific.
There are services that are region-specific:

We can't have IAM for every AZ and every region, which means users have to use one username and password for one region. Any time they switch to another region, they have to use another username and password. That's more work and not a good design. Authentication has to be global. IAM is a global service, not region-specific.
Route53 is not region-specific. We can't have a Route53 for every region. So, Route53 is not region-specific, it's a global service. And its one application users access from everywhere or from every part of the world, so we can't have one URL or one DNS name for each region, if your application is global.
WAF works well with CloudFront. CloudFront is a region-based service, so the WAF is not a region-specific service. It's a global service.
CloudFront is a global service. Though you can cache content on a continent and country basis, it's still considered a global service, it's not bound to any region. When you activate CloudFront, you're activating it away from a region or AZ. So when you're activating a WAF, because it's not a region-specific service you're activating it away from AZs and regions.

IAM

IAM Users, Groups, Roles and Accounts can be used globally across all regions.

Route 53

All Route53 services are offered at AWS edge locations and are global.

Web Application Firewall

WAF which protects web applications from common web exploits is offered at AWS edge locations and is global.

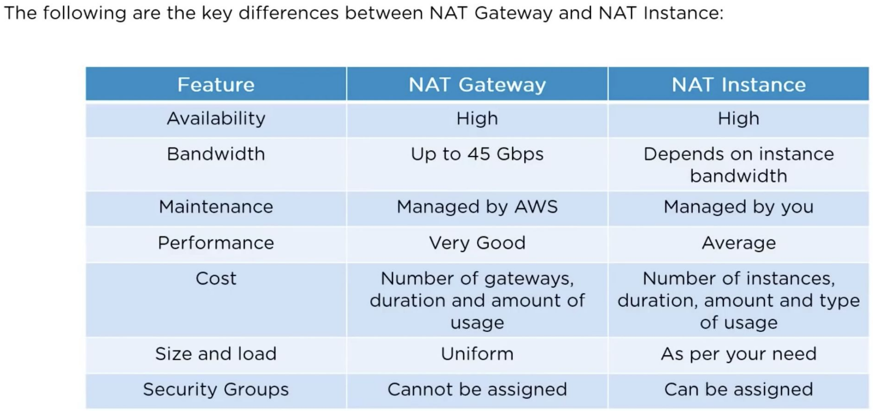
CloudFront

CloudFront is the global Content Delivery Network (CDN) service which is offered at AWS edge locations.

16. What are the differences between NAT Gateways and NAT Instances?

If your company wants to secure the environment using NAT, or is already securing their environment using NAT by using either of the methods - NAT Gateway or NAT Instances - you can expect this question.

Both NAT Gateway and NAT Instanced serve the same purpose, they are not two different services achieving different goals. Still there are differences between them. On a high level, they both achieve privising NAT'ing for the service behind it, but the difference comes when we talk about the availability of it.



The following are the key differences between NAT Gateway and NAT Instance:

Feature	NAT Gateway	NAT Instance
Availability	High	High
Bandwidth	Up to 45 Gbps	Depends on instance bandwidth
Maintenance	Managed by AWS	Managed by you
Performance	Very Good	Average
Cost	Number of gateways, duration and amount of usage	Number of instances, duration, amount and type of usage
Size and load	Uniform	As per your need
Security Groups	Cannot be assigned	Can be assigned

NAT Gateway is a managed service for Amazon, whereas NAT Instance is managed by us (maintenance).
Availability of NAT Gateway is very high, availability of NAT Instance is less compares to Gateway because it's managed by us, it's on EC2 instance which could fail. It it fails we can relaunch it, but if it's NAT Gateway, if something happens to that service Amazon would take care of reprovisioning it.
NAT Gateway traffic bandwidth can burst up to 75 Gb. NAT Instance bandwidth depends on the server. If we launch a t2,micro, it barely gets any bandwidth.
NAT Gateway performance is very high, because it's highly available and has a bigger pipe of up to 75 Gb. NAT Instance performance is average, it depends on the size of the NAT Instance that we pick.
Billing for NAT Gateway is the number of gateways that we provision and the duration for which we use the NAT Gateway.
Billing for NAT Instance is the number, duration and type of instances we use.
Security in NAT Gateway cannot be assigned, it already comes with fully-packed security.
In NAT Instance security is a bit customizable. We can go and change the security because it's a server managed by us. Can define what's allowed or not.
Size and load in NAT Gateway is uniform (it's a fixed product), but the size in the load of NAT Instance changes (can be small or big instance).

17. What is the difference between stopping and terminating an EC2 instance?

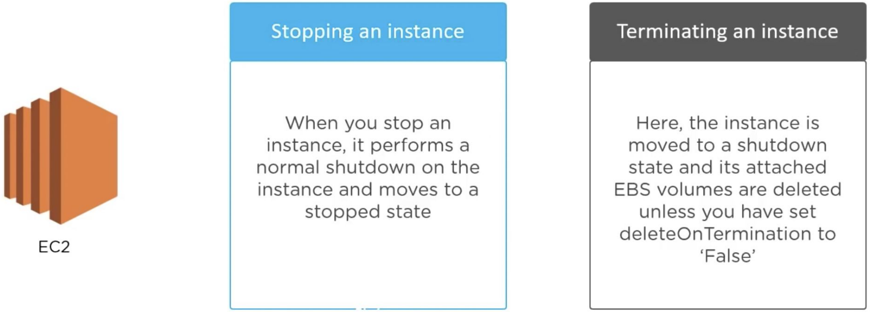
When you stop an instance it performs a normal shutdown and moves the instance to the stopped state.

When you terminate the instance, the instance is moved to the stopped state. The EBS volumes that are attached to it are deleted and removed and we'll never be able to recover them again.

That's the significant difference between stopping and terminating an instance.

If you intend to use the instance again, along with the data in it, you should only be thinking of stopping the instance.

You should be terminating the instance only if you want to get rid of that instance forever



Stopping an instance

When you stop an instance, it performs a normal shutdown on the instance and moves to a stopped state

Terminating an instance

Here, the instance is moved to a shutdown state and its attached EBS volumes are deleted unless you have set deleteOnTermination to 'False'

18. What are the different types of EC2 instances based in their cost?

There are three types of Amazon EC2 instances based on costs:

On-demand instance

Spot instance

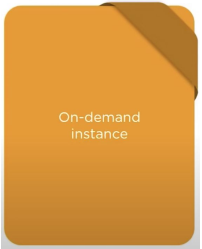
Reserved instance

All 3 types provide compute capacity and same type of hardwares.
But if you're looking at cost saving or optimizing cost in our environment, we have to be very careful about which one we're picking.
We may want to go with the on-demand instance, because we pay on a per-hour-basis. It's cheap, we can use them any time we want. And any time we don't want it, we can get rid of it by simply terminating it.

But if the requirement is to use the service for 1 year or 3 years, then we'll be wasting a lot of money buying on-demand instances, paying on an hourly basis. Instead we should be going for Reserved instance, where we can reserve the capacity for a whole year or three years and save big amount in buying reserved instances.

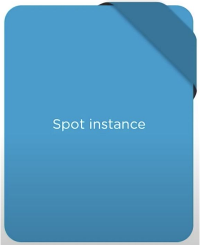
On-demand is cheap to start with, if we're only planning to use it for a short while, but if we run it for a long while then we should be going wfor Reserved instance which is more cost efficient.

Spot instance is cheaper than on-demand instance and there are different use cases for Spot instance as well.



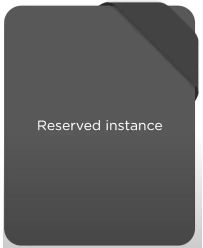
On-demand instance

- EC2 instances that are purchased at a fixed rate per hour.
- Used for applications with short-term irregular workloads that cannot be interrupted.
- Best suited for development and testing of applications.



Spot instance

- AWS allows customers to purchase unused EC2 capacity at highly reduced rates.
- Spot instances provide AWS with a flexible option to sell extra capacity.
- They are sold through a bidding process where customers bid a specific price per hour that they are willing to pay.
- The price of a Spot instance will vary based on the supply and demand in the market. *No guarantee you'll get a Spot instance at all times, that's the caveat you should remember before proposing cost savings via purchasing Spot instances. If you want a Spot instance to be available to you, then carefully watch the price history to estimate a price quote.*



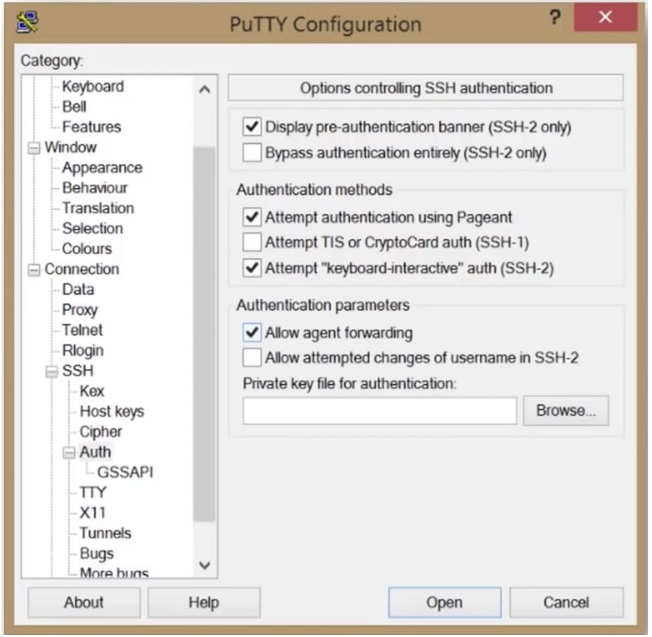
Reserved instance

- Reserved instances are mainly used for short-term and they provide cost savings for companies.
- While purchasing Reserved instances, users can opt for no upfront payment, partial payment or full upfront payment.
- Reserved instances are available in 3 types: light, medium and heavy *(based in the amount you pay).*
- *Cost benefits are based on upfront, no upfront, or a partial payment and split monthly payments afterwards.*
- *Reserve for 1 or 3 years to gain the most out of the cost benefit.*

19. How to setup SSH agent forwarding so that you don't have to copy the key every time you login?

You may be asked how to interact with the AWS environment. Are you using CLI, SDK or console?
The panelist will put a score, indicating if you are CLI-specific, console-specific, or if you've used the AWS environment through the SDK.

If you use PuTTY, every time you want to log into an EC2 instance you have to put the IP and the port number. Along with that, you have to map the key in PuTTY. This has to be done every time, e.g., in a lab environment.
But in the production environment using the same key or mapping the same key over and over is a hassle. It's considered a blocker.
You might want to cache it or permanently add it in your PuTTY session, so that you could immediately login and start using it.



Options controlling SSH authentication

☑ Display pre-authentication banner (SSH-2 only)
☐ Bypass authentication entirely (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant
☐ Attempt TIS or CryptoCard auth (SSH-1)
☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

☑ Allow agent forwarding
☐ Allow attempted changes of username in SSH-2

Private key file for authentication:

Follow the below steps to setup SSH agent forwarding:

1. **Go to your PuTTY Configuration**
2. **Got to the category SSH -> Auth**
3. **Enable SSH agent forwarding to your instance.**

This will bind ourkey to the SSH. Next time when we try to login, we don't have to always go through mapping the key and trying to login.

https://putty.en.softonic.com/mac

PuTTY is a highly configurable SSH client. The software can connect to remote servers to securely transfer data through the network.

Send and receive information by accessing distant devices through linkages.

20. What are Solaris and AIX operating systems? Are they available with AWS?

Tests your familiarity with the available AMIs, EC2s, EC2 hardwares

Your first thought may be that everything is available in AWS. I've seen Windows, Ubuntu, Red Hat, Amazon AMIs. If I don't see my operating system there, I can always go to Marketplace and try them. If Marketplace doesn't have it, I can always go to community. There are a lot of AMIs and OSs available. I'll be able to find Solaris and AIX. But this is not the case.

No, neither Solaris nor AIX are available with AWS.

Solaris uses SPARC processor architecture which is not supported in public cloud currently. Linux and Windows use x86 processors.

AIX runs only on Power CPU and not on Intel. Hence, you cannot build AIX instances in EC2, at least until AWS does not propose power machines.

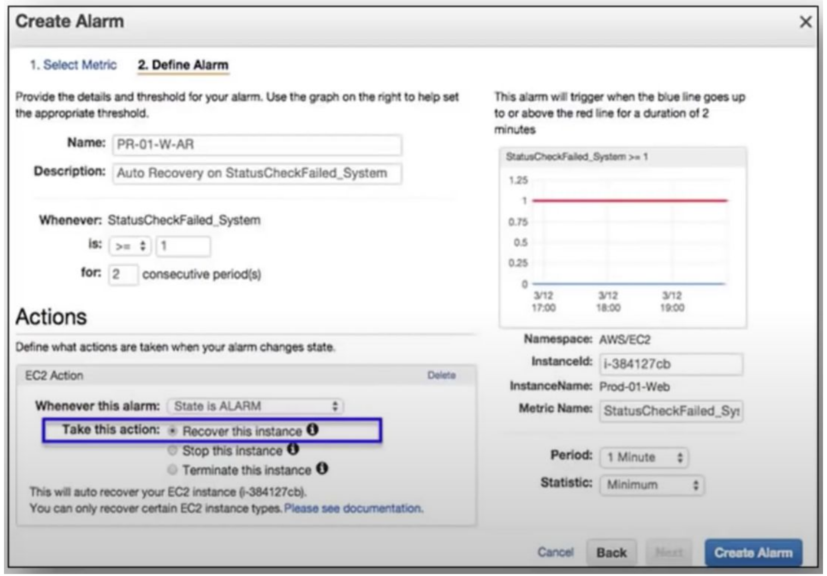
Do not confuse with HPC (high performance computing). This is different hardware, a different CPU in itself that the cloud providers do not provide yet.

21. How do you configure CloudWatch to recover an EC2 instance?

CloudWatch

- You can create an Alarm using Amazon CloudWatch
- In this Alarm, go to Define Alarm -> Actions tab
- Select the "Recover this instance" option

EC2 instances are immutable (irreparable). We don't spend time fixing bugs in an OS.
Once an EC2 crashes, it goes on panic, there are various reasons why it would fail. Don't have to worry about fixing it, can always relaunch that instance and that would fix it.
But what is it happens at 2 AM or during the weekend when nobody is in the office looking at or monitoring those instances.
You would want to automate it, not only for the nights and weekends, but as a general practice it's good to automate it.



Create Alarm

1. Select Metric   2. Define Alarm

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name: PR-01-W-AR
Description: Auto Recovery on StatusCheckFailed_System

Whenever: StatusCheckFailed_System
is: >= 1
for: 2 consecutive period(s)

Actions

Define what actions are taken when your alarm changes state.

EC2 Action   Delete

Whenever this alarm: State is ALARM
Take this action: ● Recover this instance
○ Stop this instance
○ Terminate this instance

This will auto recover your EC2 instance (i-384127cb).
You can only recover certain EC2 instance types. Please see documentation.

This alarm will trigger when the blue line goes up to or above the red line for a duration of 2 minutes

Namespace: AWS/EC2
InstanceId: i-384127cb
InstanceName: Prod-01-Web
Metric Name: StatusCheckFailed_Sy:

Period: 1 Minute
Statistic: Minimum

Can face this question - how to automate an EC2 instance once it fails? - The answer is we can use CloudWatch to recover the instance. There is an Alarm Threshold set in CloudWatch. Once the threshold is met (meaning if there's an error, a failure, EC2 instance is not responding for a certain while), we can set an alarm. Let's say CPU utilization stayed high got 5 minutes, it's not taking any connections. Or, the instance is not pinging for 2 minutes. So, it's not going to respond connection.

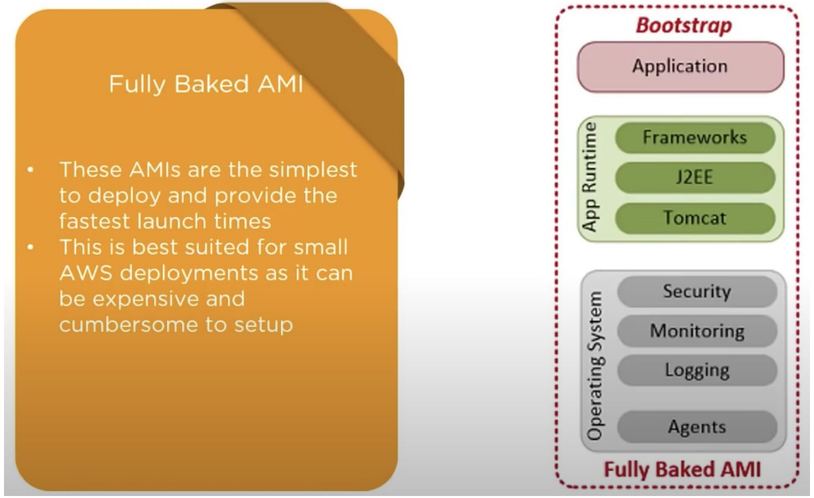
In those cases, you'd want to automatically recover that EC2 instance by rebooting the instance.
Under Actions tab, in the "Take this action" section, we have a bunch of options, like recover this instance, meaning reboot this instance. That's how we would recover.
The other 2 options are beyond the scope of the question, but let's cover them anyway.
Another option is "Stop the instance" - very useful when you want to stop instances that are having low utilization. When nobody is using the system now, you don't want them to be running and wasting the cloud expenditure. You can set an alarm that stops the EC2 instance that's having low utilization. E.g., someone was working on an instance, they left it without shutting down that instance (may have forgotten to shut down). They will only use it again the next day morning, so in between there could 12 hours that the system is running idle, nobody's using it and you're paying for it. You can identify such instances and stop them when the CPU utilization is low, meaning nobody's using it.
The other option is "Terminate the instance" - example scenario - if you want to give the system to someone temporarily and you don't want them to hand the system back to you. When they are done, they are done. We can terminate the system. You could instruct the other person to terminate the instance when they are done. And they could forget, and the instance could be running for ever. Or, you could monitor the system after the specified time is over and termite the system. Or, the best part is can automate the system termination. So you assign a system to somebody and then turn on this CloudWatch action to terminate the instance when the CPU is low for 2 hours or 30 minutes, meaning they already left.

22. What are the common and different types for AMI designs?

When working as a system-side architect, even sysop side, you could face this question.
There are a lot of AMI designs. The question is the common ones and the difference between them.

The common ones Fully Bakes AMI, Just Enough OS AMI (JeOS), and Hybrid AMI.

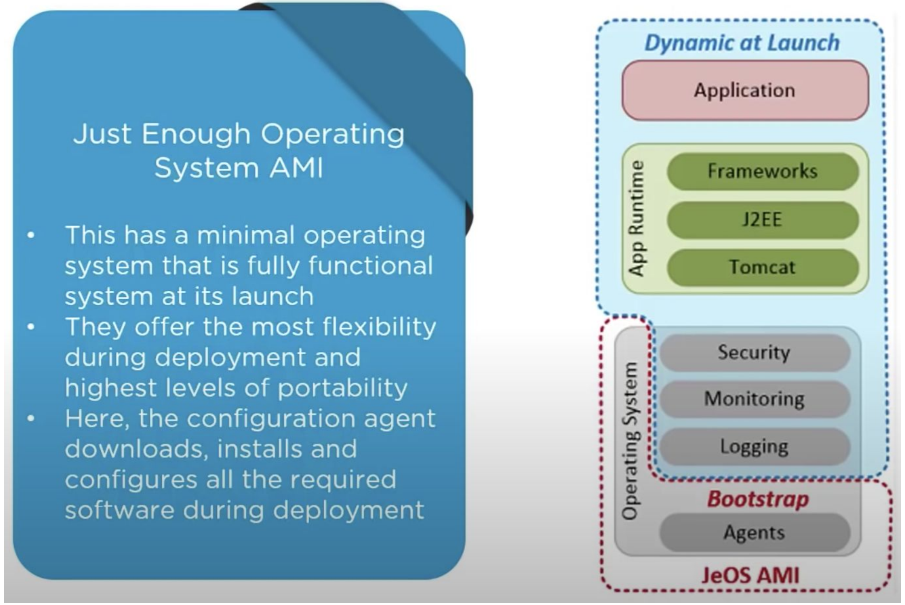
Fully Baked AMI - ready-to-use simplest AMI.
Can be a but expensive. And cumbersome, because you have to do a lot of work beforehand you could use the AMI.
A lot of planning and thought process will go into it.
The AMI is ready to use. You hand it over to somebody and it’s ready to use. Or, if you want to reuse the AMI, it’s ready for you to use.



Fully Baked AMI

- These AMIs are the simplest to deploy and provide the fastest launch times
- This is best suited for small AWS deployments as it can be expensive and cumbersome to setup

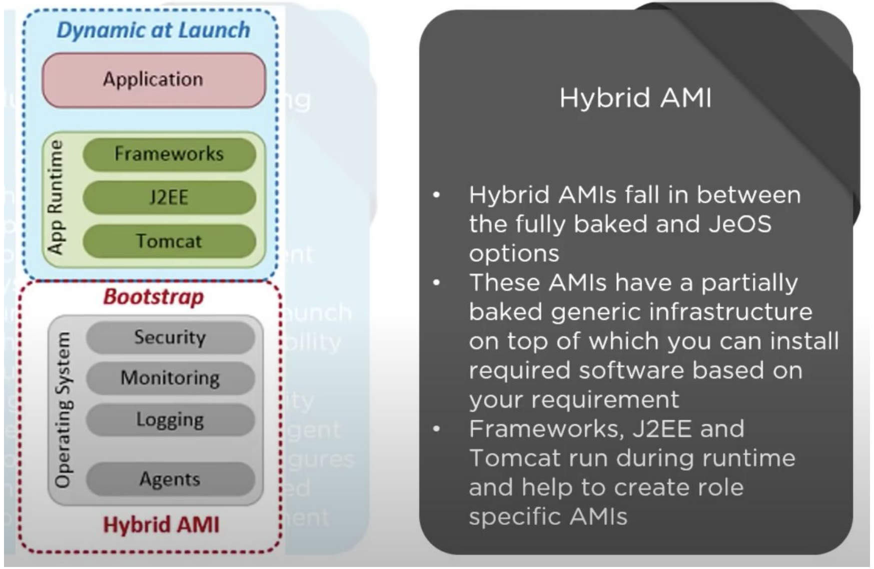
Just Enough OS AMI - covers part of the OS.
All bootstraps are already packed properly and the security logging, monitoring and the other stuff are configured at the time of deployment or at the time you'd be using it.
Not much thought process goes in here. The only focus is on choosing the OS and the OS-specific agents and bootstraps that go into the OS. That's all we worry about.
The advantage of this is that it's flexible, meaning you can choose to install additional softwares at the time of deploying but that's going to require an additional expertise from the person using the AMI. So that's another overhead there. But the advantage is that it's kind of flexible, I can change the configurations at the time of deployment.



Just Enough Operating System AMI

- This has a minimal operating system that is fully functional system at its launch
- They offer the most flexibility during deployment and highest levels of portability
- Here, the configuration agent downloads, installs and configures all the required software during deployment

Hybrid AMI - falls between the Full Bakes AMI and Just Enough OS options. These AMIs have some of the Features of Fully Baked and some of the JeOS AMIs.
The Security, Monitoring, Logging are packed in that AMI. The Runtime environments are installed at the time of Deployment. This is where the strict company policies will go into the AMI. The company policies like you got to log this, you have to monitor that, these are the ports that generally get open in all the systems, etc.
They strictly go into an AMI and sit in an AMI format, and during deployment you have the flexibility of choosing the different runtime and the application that sits in an EC2 instance.



Hybrid AMI

- Hybrid AMIs fall in between the fully baked and JeOS options
- These AMIs have a partially baked generic infrastructure on top of which you can install required software based on your requirement
- Frameworks, J2EE and Tomcat run during runtime and help to create role specific AMIs

23. How can you recover/login to an EC2 instance to which you lost the key?

We know that when the key is lost we cannot recover it. There are some organizations that integrate their EC2 instance with an AD, that's different. You can go and reset password in the AD and you'll be able to login with the new password. But here the specific tricky question is you are using a key to login and how you recover if you've lost the key. Generally the company would have made a backup of the key, so we can pick from the backup. But here the specific question is we've lost the key, literally no backups of the key at all. We can't login to the instance without the key present with us.

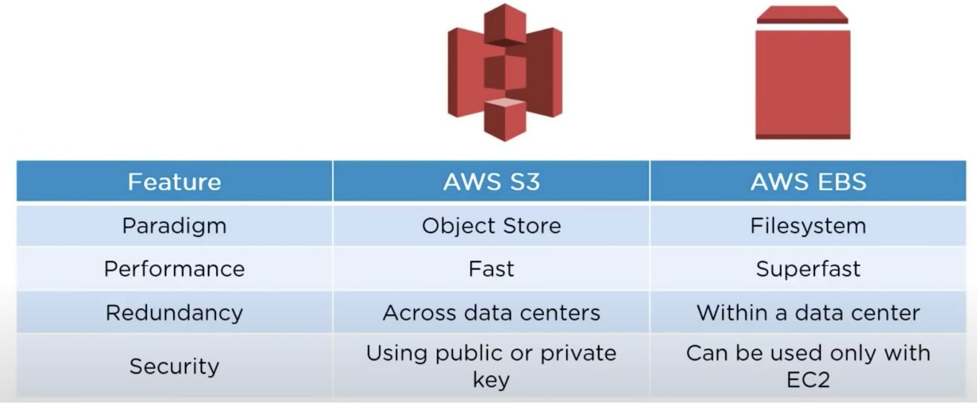
Make the instance use another key and use that key to login. Once the key is lost, it's lost forever. We won't be able to recover it. You can't raise the ticket with Amazon - not possible, they aren't going help, it's beyond scope. So make the instance use another key. It's only the key that is the problem. You still have valid data in it, you got to recover the data. Focus on the key part alone. Change the key and it will allow us to do it.

Follow the below steps to recover or login to an EC2 instance to which you have lost the key:

Step 1: Verify that the EC2Config service is running - *If you want, beforehand you can install the EC2 Config in that service. Or, you can make the EC2 Config run through the console with just a couple of button clicks and that will make it easy to configure run in that EC2 instance.*
Step 2: Detach the root volume from the instance - *It's going to require a stop and start to detach.*
Step 3: Attach the volume to a temporary instance - *Attach to another instance as a temporary volume. Or, it can be a temporary instance that you've launched only to fix this issue. Then login to that instance and to that particular volume.*
Step 4: Modify the configuration file - *To use the new key and then move the root volume back to its original position.*
Step 5: Restart the original instance - *Now the insurance is going to have the new key. And you have the new key with which you can login.*

24. What are some key differences between AWS S3 and EBS?

***The general perception is that S3 and EBS can be used interchangeably.
EBS uses S3, but they can't be interchangeably used.***



Feature	AWS S3	AWS EBS
Paradigm	Object Store	Filesystem
Performance	Fast	Superfast
Redundancy	Across data centers	Within a data center
Security	Using public or private key	Can be used only with EC2

S3 is an object store, meaning you can't install anything in it. You can store drive files but you can't actually install in it. It's not a file system.
But EBS is a file system. You can install applications and they are going to run.

Performance-wise, when accessing S3 from the instance you have to go out through the internet access S3. S3 is an external service, you have to go outside of your VPC to access S3. S3 does not come under a VPC. But EBS comes under a VPC, it's on the same VPC. you'll be able to use it kind of locally. Compared to S3, EBS is very local, that way it's faster.

Redundancy-wise, S3 is replicated. The data in S3 is replicated across the data centers. But EBS is replicated within the data center, meaning S3 is replicated across AZs. EBS is within an AZ. That way the EBS redundancy is a bit less in EBS. Redundancy is higher in S3.

Security-wise, S3 can be made private as well as public, meaning anybody can access S3 from anywhere in the internet. That's possible with S3. But EBS can only be accessed when attached to an EC2 instance, just one instance can access it. Whereas S3 is publicly directly accessible.

25. How do you allow access to a user to a certain bucket?

Question related to S3 security.
This user doesn't have access to S3 at all, but needs to be given access to a certain bucket. The same thing applies to the servers as well.
A person may be new to the team, and you don't want them to access the production servers. they are a part of the production group, but you want to specifically restrict access to the production server until they've matured enough to understand the process, dos and don'ts before they put their hands on the production server .

We will follow the following four steps to allow access to a certain bucket:

- Step 1: Categorize your instances - *critical, normal - we put a tag on them to categorize. Tags, such as medium critical, high critical, or not critical at all, still in production. etc.*
- Step 2: Define how authorized users can (or can't) manage specific servers - *Pick the users who should or shouldn't be given access to a certain server. Based on tags, allow or disallow the server access. Define that this user is allowed or denied to use resources with this tag, etc. in this case, the user is not allowed to access a server with a critical tag. Same goes for the bucket.*
- Step 3: Lock down your tags
- Step 4: Attach your policies to IAM users

26. How can you monitor S3 cross region replication to ensure consistency without actually checking the bucket?

If an organization is excessively using S3 for their data storage, because of the benefit it provides, the cost and durability, you might get asked this question.
Organizations would replicate their data from one region to another for additional data durability, data redundancy, disaster recovery purposes. If the whole region is down, you still have the data available somewhere else and you can pick and use it.
Some organizations will store data in different regions for compliance reasons, to provide low latency access to the users who are local to that region.
When companies do region replication, how to ensure there's consistency, that the replication is not failing and the data gets transferred for sure, and there are logs for that replication. This is smth the companies would use, when they are excessively using S3 and fully relying on the replication in running their business.

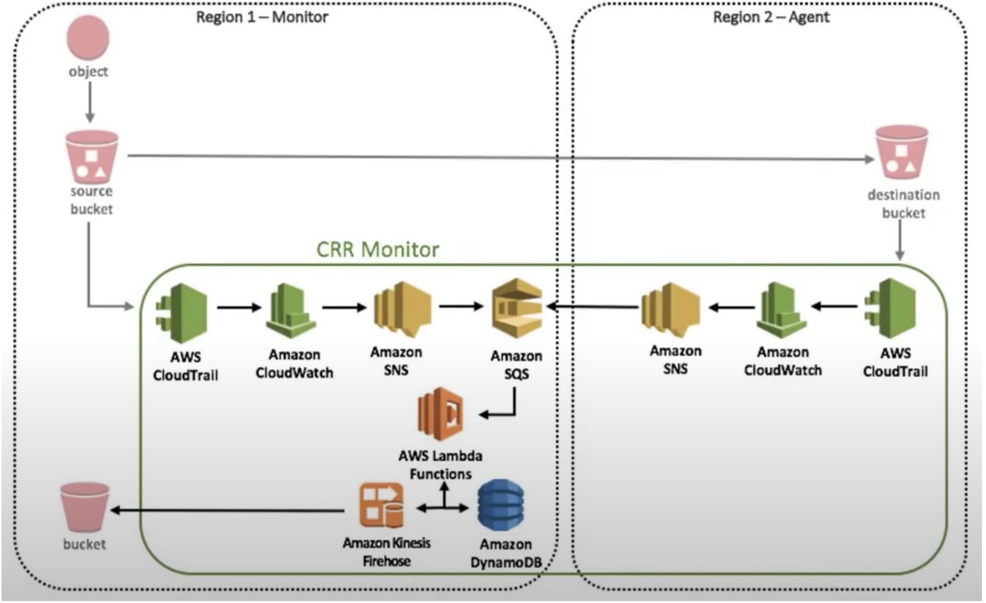
Cross-Region Replication Monitor (CRR Monitor) application is used to monitor the replication status of your Amazon S3 objects.

It's a set of tools that we could use together to make sure that the cloud region-level replication is happening properly.

CRR monitors your environment. CloudWatch that makes sure the data is moved, no data is failing. There's CloudWatch in the other end to make sure the data is moving. Then we have the logs generated through CloudTrail, that is written in DynamoDB.

If there's is failure, you get notified through an SMS or email using SNS service.

That's how we can leverage these tools and set up a cross region replication monitor that monitors your data replication.



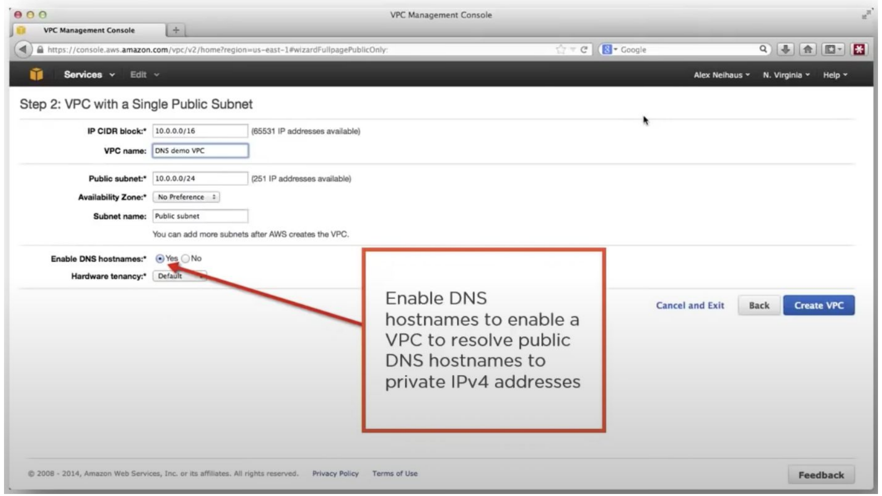
27. VPC is not resolving the server through DNS. What might be the issue and how can you fix it?

We can use DNS to resolve the IP address externally from the internet. But by default the servers will not connect to other servers using a custom DNS name. That does not do that by default. There are additional things that as an administrator or an architect, you have to do. You can access the server through the IP but not through the DNS name.

To enable a VPC to resolve public IPv4 DNS hostnames to private IPv4 addresses when queried from instances in the peer VPC, you must modify the peering connection.

Enable hostname resolution before, so they actually resolve. This is for the custom DNS, not default DNS that comes along.

Let's say I want to connect to server1.amazon.com, by default it's not allowed. But if enable this option then I'll be able to connect to server1.amazon.com.



Step 2: VPC with a Single Public Subnet

IP CIDR block:* 10.0.0.0/16 (65531 IP addresses available)

VPC name: DNS demo VPC

Public subnet:* 10.0.0.0/24 (251 IP addresses available)

Availability Zone:* No Preference

Subnet name: Public subnet

You can add more subnets after AWS creates the VPC.

Enable DNS hostnames:* ● Yes ○ No

Hardware tenancy:* Default

Enable DNS hostnames to enable a VPC to resolve public DNS hostnames to private IPv4 addresses

Cancel and Exit
Back
Create VPC
© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Feedback

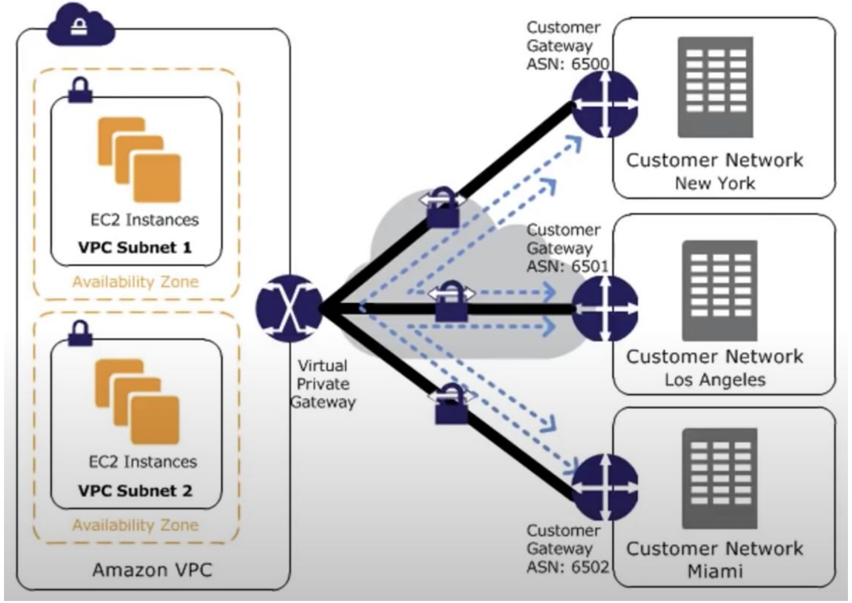
28. How do you connect multiple sites to a VPC?

If the company has VPCs in different regions and they have a head office in a central place and rest of them are branch services and they connect to the head office for access or saving data, accessing certain files or data, or storing data, they would mimic the hub and spoke topology, where you have the VPC which is in a centrally accessible region, then you'd have local VPCs or branch offices in different other regions. And they get connected to the VPC in the central location, how do you connect the multiple sited and make communication happen between them? By default it does not do that. VPCs need to be paired between them in order to access the resources.

If you have multiple VPN connections, you can provide secure communication between sites using the AWS VPN CloudHub.

You can connect multiple sites to a VPC shown in the diagram.

The different remote offices are connecting to the VPC and they're talking. But they can't connect or talk to each other. But the requirement is the traffic needs or they should be able to talk to each other, but they should not have direct connection between them. They will have to come and hit the VPC and then reach the other customer network in another city - that's the requirement. That's possible with some architecting in the cloud. That's using VPN CloudHub. The dotted lines allow customers or corporate networks to talk to each other through the VPC. Again, by default it does not happen. Cloudhub is the architecture that we should be using to make this happen. The advantage is, as a central HQ office or data center, which is in the VPC, you or the VPC have control of who talks to who, what traffic can be routed to the other head office. That centralized control is on the VPC.



EC2 Instances
VPC Subnet 1
Availability Zone
EC2 Instances
VPC Subnet 2
Availability Zone
Amazon VPC
Virtual Private Gateway
Customer Gateway ASN: 6500
Customer Network New York
Customer Gateway ASN: 6501
Customer Network Los Angeles
Customer Gateway ASN: 6502
Customer Network Miami

29. Name and explain some security products and features available in VPC.

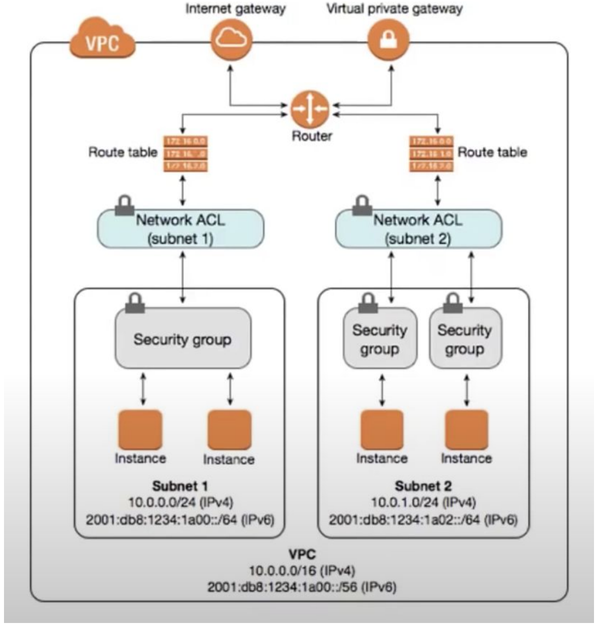
VPC itself is a security service, it provides security service to the application but how do secure the VPC itself. There are products that can secure the VPC, or VPC delivers those products to secure the application.

- Security groups - Act as a firewall for associated Amazon EC2 instances, controlling both inbound and outbound traffic at the instance level

- Network access control lists (ACLs) — Act as a firewall for associated subnets, controlling both inbound and outbound traffic at the subnet level

- Flow logs - Capture information about the IP traffic going to and from network interfaces in your VPC

Flow logs are used in later analysis as in what's the traffic pattern and behavior.



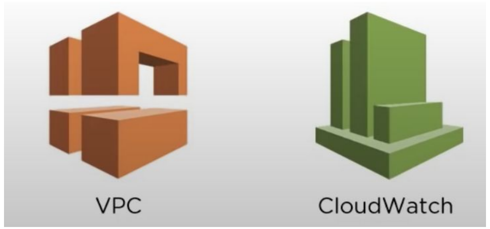
30. How do you monitor Amazon VPC?

VPC is an important service. Most services, except Lambda, S3, DynamoDB and a few other services, sit in a VPC for security reasons. How do you gain some visibility on your VPC?

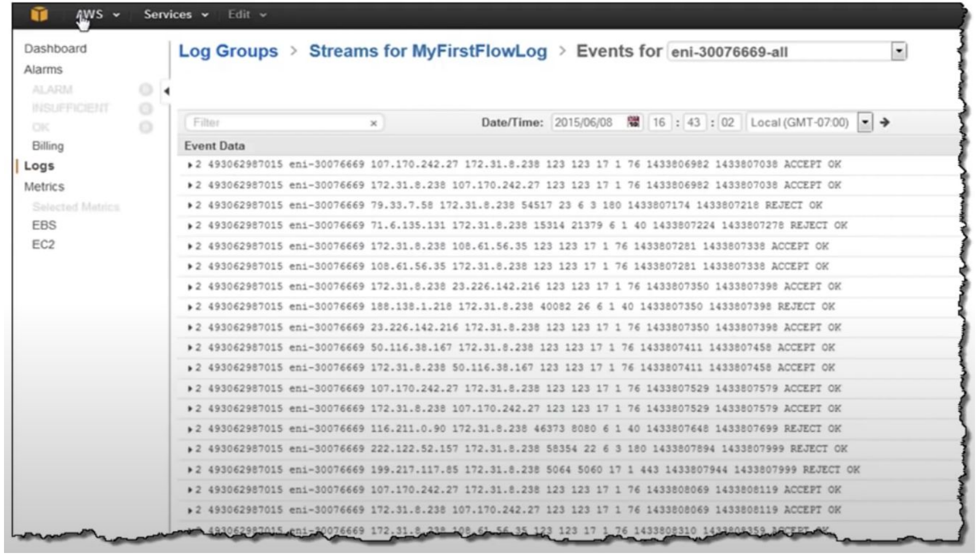
You can monitor Amazon VPC using the following:

□ CloudWatch and CloudWatch Logs

□ VPC Flow Logs



VPC Flow Logs is a basic service, captures what's allowed or not allowed. E.g. which IPs are allowed. We can gather it and use for analysis.


AWS
Services
Edit
Dashboard
Alarms
ALARM
INSUFFICIENT
OK
Billing
Logs
Metrics
Selected Metrics
EBS
EC2
Log Groups > Streams for MyFirstFlowLog > Events for eni-30076669-all
Filter
Date/Time: 2015/06/08 16 : 43 : 02 Local (GMT-07:00)
Event Data

```
▸2 493062987015 eni-30076669 107.170.242.27 172.31.8.238 123 123 17 1 76 1433806982 1433807038 ACCEPT OK
▸2 493062987015 eni-30076669 172.31.8.238 107.170.242.27 123 123 17 1 76 1433806982 1433807038 ACCEPT OK
▸2 493062987015 eni-30076669 79.33.7.58 172.31.8.238 54517 23 6 3 180 1433807174 1433807218 REJECT OK
▸2 493062987015 eni-30076669 71.6.135.131 172.31.8.238 15314 21379 6 1 40 1433807224 1433807278 REJECT OK
▸2 493062987015 eni-30076669 172.31.8.238 108.61.56.35 123 123 17 1 76 1433807281 1433807338 ACCEPT OK
▸2 493062987015 eni-30076669 108.61.56.35 172.31.8.238 123 123 17 1 76 1433807281 1433807338 ACCEPT OK
▸2 493062987015 eni-30076669 172.31.8.238 23.226.142.216 123 123 17 1 76 1433807350 1433807398 ACCEPT OK
▸2 493062987015 eni-30076669 188.138.1.218 172.31.8.238 40082 26 6 1 40 1433807350 1433807398 REJECT OK
▸2 493062987015 eni-30076669 23.226.142.216 172.31.8.238 123 123 17 1 76 1433807350 1433807398 ACCEPT OK
▸2 493062987015 eni-30076669 50.116.38.167 172.31.8.238 123 123 17 1 76 1433807411 1433807458 ACCEPT OK
▸2 493062987015 eni-30076669 172.31.8.238 50.116.38.167 123 123 17 1 76 1433807411 1433807458 ACCEPT OK
▸2 493062987015 eni-30076669 107.170.242.27 172.31.8.238 123 123 17 1 76 1433807529 1433807579 ACCEPT OK
▸2 493062987015 eni-30076669 172.31.8.238 107.170.242.27 123 123 17 1 76 1433807529 1433807579 ACCEPT OK
▸2 493062987015 eni-30076669 116.211.0.90 172.31.8.238 46373 8080 6 1 40 1433807648 1433807699 REJECT OK
▸2 493062987015 eni-30076669 222.122.52.157 172.31.8.238 58354 22 6 3 180 1433807894 1433807999 REJECT OK
▸2 493062987015 eni-30076669 199.217.117.85 172.31.8.238 5064 5060 17 1 443 1433807944 1433807999 REJECT OK
▸2 493062987015 eni-30076669 107.170.242.27 172.31.8.238 123 123 17 1 76 1433808069 1433808119 ACCEPT OK
▸2 493062987015 eni-30076669 172.31.8.238 107.170.242.27 123 123 17 1 76 1433808069 1433808119 ACCEPT OK
▸2 493062987015 eni-30076669 172.31.8.238 108.61.56.35 123 123 17 1 76 1433808310 1433808359 ACCEPT OK
```

And another one is CloudWatch and CloudWatch logs. The data transfers that happen. The Flows Logs is who gets allowed or denied, that kind of detail. And CloudWatch fives info about the data transfer, how much data is getting transferred. We can pick unusual data transfers if there is a certain hike in a graph. There's a certain hike that happens at 12 on a regular basis and you weren't expecting it, there's smth suspicious it could valid backups or it could be malicious activity as well. That's what you know by looking at CloudWatch logs and dashboard.