**What is AWS (Amazon Web Services)? Did got chance to work on Amazon tools?**

AWS provides a set of flexible services designed to enable companies to create and deliver products with greater speed and reliability using AWS and DevOps practices. These services simplify commissioning and infrastructure management, application code deployment, automated software release process and monitoring of the application and infrastructure performance. Amazon used tools like AWS CodeCommit, AWS CodeDeploy, AWS CodePipeline etc. that helps to make DevOps easier.

**Name the several layers of Cloud Computing.**

Here is the list of layers of the cloud computing

* SaaS: Software as a Service (SaaS), it provides users access directly to the cloud application without installing anything on the system.
* IaaS: Infrastructure as a service, it provides the infrastructure in terms of hardware like memory, processor speed etc.
* PaaS: Platform as a service, it provides cloud application platform for the developers

**What are the advantages of using cloud computing?**

The advantages of using cloud computing are

* Data backup and storage of data
* Powerful server capabilities
* SaaS (Software as a service)
* Information technology sandboxing capabilities
* Increase in productivity
* Cost effective & Time saving

**How user can gain from utility computing?**

* Utility computing allows the user to pay only for what they are using. It is a plug-in managed by an organization which decides what type of services should be deployed from the cloud.
* Most organizations prefer hybrid strategy.

**What are easiest ways to build a small cloud?**

VMfest is one of the options for making IaaS cloud from VirtualBox VMs in no time. If you want a lightweight PaaS there is Dokku which is basically a bash script that makes PaaS out of Dokku containers.

**List down the three basic clouds in cloud computing?**

* Professional cloud
* Personal cloud
* Performance cloud

**As an infrastructure as a service what are the resources that are provided by it?**

IAAS (Infrastructure as a Service) provides virtual and physical resources that are used to build a cloud. It deals with the complexities of deploying and maintaining of the services provided by this layer. Here the infrastructure is the servers, storage and other hardware systems.

**What are the business benefits involved in cloud architecture?**

The benefits involved in cloud architecture is

* Zero infrastructure investment
* Just in time infrastructure.
* More efficient resource utilization

**What are the of cloud architecture that separates it from traditional one?**

The characteristics that makes cloud architecture above traditional architecture is

* According to the demand cloud architecture provides the hardware requirement
* Cloud architecture can scale the resource on demand
* Cloud architecture can manage and handling dynamic workloads without failure

**Mention platforms which are used for large scale cloud computing?**

The platforms that are used for large scale cloud computing are

* Apache Hadoop
* MapReduce

**What are the security aspects provided with cloud?**

* Identity management: It authorizes the application services
* Access control: permission must be provided to the users so that they can control the access of another user who is entering the cloud environment.
* Authentication and Authorization: Allows only the authorized and authenticated user only to access the data and applications

**What are the different deployment models for cloud computing?**

The different models are:

* Private Cloud
* Public Cloud
* Hybrid Cloud
* Community Cloud

For the description of each deployment model and service models, refer to the below given link:

<https://www.linkedin.com/pulse/3-service-4-deployment-models-cloud-computing-sankar-somepalle>

**List out different layers which define cloud architecture?**

There are five layers:

* Cloud Controller (CLC)
* Walrus
* Cluster Controller
* Storage Controller (SC)
* Node Controller (NC)

**What is the requirement of virtualization platform in implementing cloud?**

The requirement of virtualization platform in implementing cloud is to

* Manage the service level policies
* Cloud Operating System
* Virtualization platforms helps to keep the backend level and user level concepts different from each other

**Before going for cloud computing platform what are the essential things to be taken in concern by users?**

* Compliance
* Loss of data
* Data storage
* Business continuity
* Uptime
* Data integrity in cloud computing

**Mention some open source cloud computing platform databases?**

The open source cloud computing platform databases are

* MongoDB
* CouchDB
* LucidDB

**What are the security laws which are implemented to secure data in a cloud?**

The security laws which are implemented to secure data in cloud are

* Processing: Control the data that is being processed correctly and completely in an application
* File: It manages and control the data being manipulated in any of the file
* Output reconciliation:  It controls the data which has to be reconciled from input to output
* Input Validation: Control the input data
* Security and Backup: It provides security and backup it also controls the security breaches logs

**Mention the name of some large cloud providers and databases?**

* Google bigtable
* Amazon simpleDB
* Cloud based SQL

**Explain the difference between cloud and traditional datacenters?**

* The cost of the traditional data center is higher due to heating and hardware/software issues
* Cloud gets scaled when the demand increases.  Majority of the expenses are spent on the maintenance of the data centers, while that is not the case with cloud computing

**Explain what are the different modes of software as a service (SaaS)?**

* Simple multi-tenancy:  In this each user has independent resources and are different from other users, it is an efficient mode.
* Fine grain multi-tenancy:  In this type, the resources can be shared by many but the functionality remains the same.

**How important is the platform as a service?**

Platform as a service or PAAS is an important layer in cloud computing.  It provides application platform for providers.  It is responsible for providing complete virtualization of the infrastructure layer and makes it work like a single server.

**What is a cloud service?**

Cloud service is used to build cloud applications using the server in a network through internet.  It provides the facility of using the cloud application without installing it on the computer. It also reduces the maintenance and support of the application which are developed using cloud service.

**What are system integrators in Cloud Computing?**

In Cloud Computing, systems integrator provides the strategy of the complicated process used to design a cloud platform. Integrator allows to create more accurate hybrid and private cloud network, as integrators have all the knowledge about the data center creation.

**What is “EUCALYPTUS” stands for?**

“EUCALYPTUS” stands for Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems”

**Explain what is the use of “EUCALYPTUS” in cloud computing?**

“Eucalyptus” is an open source software infrastructure in cloud computing, which is used to implement clusters in cloud computing platform. It is used to build public, hybrid and private clouds. It can produce your own data center into a private cloud and allows you to use its functionality to many other organizations.

**Why API’s have in cloud services?**

Application Programming Interface (API) has the following uses:

* It eliminates the need to write fully fledged programs
* It provides the instructions to set up communication between one or more applications
* It allows easy creation of applications and links the cloud services with other systems

**How many data centers are deployed for cloud computing? What are they?**

There are two data centers in cloud computing:

* Containerized Data centers
* Low Density Data centers

**What is the difference in cloud computing and computing for mobiles?**

Mobile computing uses the same concept as cloud computing.   Cloud computing becomes active with the data with the help of internet rather than individual device. It provides users with the data which they should retrieve on demand.  In mobile, the applications run on the remote server and gives user the access for storage and manage.

**What you know about Serverless model?**

* Serverless refers to a model where the existence of servers is hidden from developers. It means you no longer should deal with capacity, deployments, scaling and fault tolerance and OS. It will essentially reduce maintenance efforts and allow developers to quickly focus on developing codes.
* Examples are Amazon AWS Lambda and Auth0 Serverless platform.

**What is AWS Certificate Manager?**



AWS Certificate Manager (ACM) handles the complexity of provisioning, deploying, and managing certificates provided by ACM (ACM Certificates) for your AWS-based websites and applications. You use ACM to request and manage the certificate and then use other AWS services to provision the ACM Certificate for your website or application. As shown by the above illustration, ACM Certificates are currently available for use with only Elastic Load Balancing and Amazon CloudFront. You cannot use ACM Certificates outside of AWS.

**What is Redshift?**

Redshift is a fast, fully managed, petabyte-scale data warehouse service that makes it simple and cost-effective to efficiently analyze all your data using your existing business intelligence tools.

**What is VPC?**

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. You can configure or create your VPC as per requirement like select region, create subnets (IP- CIDR), configure route tables, security groups, Internet gateway etc. to your AWS account by which you can launch your AWS resources, such as Amazon EC2, RDS instances etc. into your VPC. So, basically you can say that Amazon VPC is the networking layer for AWS Infrastructure.

**What is VPC peering?**

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IP addresses. And instances which is in VPC can communicate with each other as if they are within the same network.

You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account within a single region.

If you have more than one AWS account within a same region and wants to share or transfer the data, you can peer the VPCs across those accounts to create a file sharing network. You can also use a VPC peering connection to allow other VPCs to access resources you have in one of your VPCs.

A VPC peering connection can help you to facilitate the transfer of data.

**What is VPC endpoints?**

A VPC endpoint enables you to create a private connection between your VPC with another AWS service without requiring access over the Internet, through a NAT device, a VPN connection, or AWS Direct Connect. They are horizontally scaled, redundant, and highly available VPC components that allow communication between instances in your VPC and AWS services without imposing availability risks or bandwidth constraints on your network traffic.

An endpoint enables instances in your VPC to use their private IP addresses to communicate with resources in other services. Don’t require public IP addresses to your instances, and you don’t need an Internet gateway, a NAT device, or a virtual private gateway in your VPC.

**What are microservices and why they have an impact on operations?**

Microservices are a product of software architecture and programming practices. Microservices architectures typically produce smaller, but more numerous artifacts that Operations is responsible for regularly deploying and managing. For this reason, microservices have an important impact on Operations. The term that describes the responsibilities of deploying microservices is micro deployments. So, what DevOps is about is bridging the gap between microservices and micro deployments.

**What is an AMI?**

AMI stands for Amazon Machine Image. It is effectively a snapshot of theroot filesystem. AWS AMI provides the information required to launch an instance, which is a virtual server in the cloud. You specify an AMI when you launch an instance, and you can launch as many instances from the AMI as you need. You can also launch instances from as many different AMIs as you need.

An AMI includes the following:

* A template for the root volume for the instance (such as an operating system, an application server, and applications).
* Launch permissions that control which AWS accounts can use the AMI to launch instances.
* A block device mapping that specifies the volumes to attach to the instance when it’s launched
* Build a new AMI by first spinning up and instance from a trusted AMI.  Then adding packages and components as required.  Be wary of putting sensitive data onto an AMI.
* For instance, your access credentials should be added to an instance after spinning up.  With a database, mount an outside volume that holds your MySQL data after spinning up as well.

**What is IAM service?**

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources for your users. You use IAM to control who can use your AWS resources (authentication) and what resources they can use and in what ways (authorization).

**What is the relation between instance and AMI?**

An Amazon Machine Image (AMI) is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, you launch an instance, which is a copy of the AMI running as a virtual server in the cloud.

You can launch several types of instances from a single AMI. An instance type determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory capabilities.

**What is instance profile? How do you create a role?**

It’s a container for IAM role and you can pass this role information to a EC2-instance when the instance starts. This role gives access permissions to S3 buckets and other repositories where your applications are stored.

* We can create IAM instance profile by using AWS CLI
* Command to create instance profile is ---
* Aws iam create -instance-profile - -instance-profile-name

**What are the key components of AWS (Amazon Web Service)?**  
The key components of AWS are:

* **Route 53:** A DNS web service
* **Simple E-mail Service:** It allows sending e-mail using RESTFUL API call or via regular SMTP
* **Identity and Access Management:** It provides enhanced security and identity management for your AWS account
* **Simple Storage Device or (S3):** It is a storage device and the most widely used AWS service
* **Elastic Compute Cloud (EC2):** It provides on-demand computing resources for hosting applications. It is very useful in case of unpredictable workloads
* **Elastic Block Store (EBS):** It provides persistent storage volumes that attach to EC2 to allow you to persist data past the lifespan of a single EC2
* **CloudWatch:** To monitor AWS resources. It allows administrators to view and collect key Also, one can set a notification alarm in case of trouble.

**What is S3?**

S3 stands for Simple Storage Service. You can use S3 interface to store and retrieve any amount of data, at any time and from anywhere on the web. Also, we can host a website in Amazon S3. most of the companies storing the documents, images and other files to S3. For S3, the payment model is “pay as you go”.

**What Is Amazon EC2?**

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

**Explain some features of Amazon EC2?**

* Virtual computing environments, known as instances
* Preconfigured templates for your instances, known as Amazon Machine Images (AMIs), that package the bits you need for your server (including the operating system and additional software)
* Various configurations of CPU, memory, storage, and networking capacity for your instances, known as instance types
* Secure login information for your instances using key pairs (AWS stores the public key, and you store the private key in a secure place)
* Storage volumes for temporary data that’s deleted when you stop or terminate your instance, known as instance store volumes
* Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as Amazon EBS volumes
* Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as regions and Availability Zones
* A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using security groups
* Static IP addresses for dynamic cloud computing, known as Elastic IP addresses

**How you will find out the instance id from within an ec2 machine?**

* wget -q -O – http://instance-data/latest/meta-data/instance-id
* If you need programmatic access to the instance ID from within a script  
  die() { status=$1; shift; echo “FATAL: $\*”; exit $status; }  
  EC2\_INSTANCE\_id=”`wget -q -O – http://instance-data/latest/meta-data/instance-id || die \”wget instance-id has failed: $?\”`”

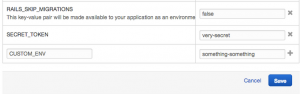
**For a transport in cloud how you can secure your dat**

To secure your data while transporting them from one place to another, check that there is no leak with the encryption key implemented with the data you are sending.

**How do you pass custom environment variable on Amazon Elastic Beanstalk (AWS EBS)?**

As a head, up to anyone who uses the .ebextensions/\*.config way: nowadays you can add, edit and remove environment variables in the Elastic Beanstalk web interface.

The variables are under Configuration? Software Configuration:



**What is DynamoDB?**

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. You can use Amazon DynamoDB to create a database table that can store and retrieve any amount of data, and serve any level of request traffic. Amazon DynamoDB automatically spreads the data and traffic for the table over a sufficient number of servers to handle the request capacity specified by the customer and the amount of data stored, while maintaining consistent and fast performance.

**What is ElastiCache?**

ElastiCache is a web service that makes it easy to set up, manage, and scale distributed in-memory cache environments in the cloud.

**What is Amazon Kinesis Firehose?**

Amazon Kinesis Firehose is a fully managed service for delivering real-time streaming data to destinations such as Amazon Simple Storage Service (Amazon S3) and Amazon Redshift.

**What Is Amazon CloudSearch and its features?**

Amazon CloudSearch is a fully managed service in the cloud that makes it easy to set up, manage, and scale a search solution for your website or application.

You can use Amazon CloudSearch to index and search both structured data and plain text. Amazon CloudSearch features:

* Full text search with language-specific text processing
* Boolean search
* Prefix searches
* Range searches
* Term boosting
* Faceting
* Highlighting
* Autocomplete Suggestions

**What is the AWS Key Management Service?**

The AWS Key Management Service (AWS KMS) is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data.

**The following components of AWS Data Pipeline work together to manage your data:**

A pipeline definition specifies the business logic of your data management. For more information, see Pipeline Definition File Syntax.

A pipeline schedules and runs tasks. You upload your pipeline definition to the pipeline, and then activate the pipeline. You can edit the pipeline definition for a running pipeline and activate the pipeline again for it to take effect. You can deactivate the pipeline, modify a data source, and then activate the pipeline again. When you are finished with your pipeline, you can delete it.

Task Runner polls for tasks and then performs those tasks. For example, Task Runner could copy log files to Amazon S3 and launch Amazon EMR clusters. Task Runner is installed and runs automatically on resources created by your pipeline definitions. You can write a custom task runner application, or you can use the Task Runner application that is provided by AWS Data Pipeline. For more information, see Task Runners.

**What is Regions and Endpoints in AWS?**

* To reduce data latency in your applications, most Amazon Web Services products allow you to select a regional endpoint to make your requests. An endpoint is a URL that is the entry point for a web service. For example, https://dynamodb.us-west-2.amazonaws.com is an entry point for the Amazon DynamoDB service.
* Some services, such as IAM, do not support regions; their endpoints therefore do not include a region. A few services, such as Amazon EC2, let you specify an endpoint that does not include a specific region, for example, https://ec2.amazonaws.com. In that case, AWS routes the endpoint to us-east-1.

**What is AWS WAF? What are the potential benefits of using WAF?**

AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to Amazon CloudFront and lets you control access to your content. Based on conditions that you specify, such as the IP addresses that requests originate from or the values of query strings, CloudFront responds to requests either with the requested content or with an HTTP 403 status code (Forbidden. You can also configure CloudFront to return a custom error page when a request is blocked.

Benefits of using WAF:

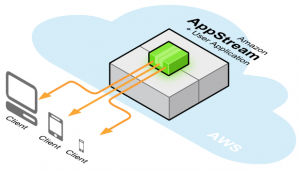
* Additional protection against web attacks using conditions that you specify. You can define conditions by using characteristics of web requests such as the IP address that the requests originate from, the values in headers, strings that appear in the requests, and the presence of malicious SQL code in the request, which is known as SQL injection.
* Rules that you can reuse for multiple web applications
* Real-time metrics and sampled web requests
* Automated administration using the AWS WAF API

**How to find your regions and Availability Zones using the Amazon EC2 CLI?**

Use the ec2-describe-regions command as follows to describe your regions.  
PROMPT> ec2-describe-regions  
REGION us-east-1 ec2.us-east-1.amazonaws.com  
REGION ap-northeast-1 ec2.ap-northeast-1.amazonaws.com  
REGION ap-southeast-1 ec2.ap-southeast-1.amazonaws.com

**What is Amazon AppStream and advantage of using AppStreaming?**

Amazon AppStream is an application streaming service that lets you stream your existing resource-intensive applications from the cloud without code modifications.



**Advantages of Streaming Your Application?**

Interactively streaming your application from the cloud provides several benefits:

* **Remove Device Constraints –** You can leverage the compute power of AWS to deliver experiences that wouldn’t normally be possible due to the GPU, CPU, memory or physical storage constraints of local devices.
* **Support Multiple Platforms –** You can write your application once and stream it to multiple device platforms. To support a new device, just write a small client to connect to your streaming application.
* **Fast and Easy Updates –** Because your streaming application is centrally managed by Amazon AppStream, updating your application is as simple as providing a new version of your streaming application to Amazon AppStream. You can immediately upgrade all of your customers without any action on their part.
* **Instant On –** Streaming your application with Amazon AppStream lets your customers start using your application or game immediately, without the delays associated with large file downloads and time-consuming installations.
* **Improve Security –** Unlike traditional boxed software and digital downloads, where your application is available for theft or reverse engineering, Amazon AppStream stores your streaming application binary securely in AWS datacenters.
* **Automatic Scaling –** You can use Amazon AppStream to specify capacity needs, and then the service automatically scales your streamed application and connects customers’ devices to it.

**What are the benefits of EBS vs. instance-store?**

* EBS backed instances can be set so that they cannot be (accidentally) terminated through the API.
* EBS backed instances can be stopped when you’re not using them and resumed when you need them again (like pausing a Virtual PC), at least with my usage patterns saving much more money than I spend on a few dozen GB of EBS storage.
* EBS backed instances don’t lose their instance storage when they crash (not a requirement for all users, but makes recovery much faster)
* You can dynamically resize EBS instance storage.
* You can transfer the EBS instance storage to a brand-new instance (useful if the hardware at Amazon you were running on gets flaky or dies, which does happen from time to time)
* It is faster to launch an EBS backed instance because the image does not have to be fetched from S3.

**Which AWS service responsible for managed email and calendaring?**

WorkMail is a managed email and calendaring service with strong security controls and support for existing desktop and mobile email clients. You can access their email, contacts, and calendars wherever you use Microsoft Outlook, your browser, or your iOS and Android mobile devices. You can integrate Amazon WorkMail with your existing corporate directory and control both the keys that encrypt your data and the location where your data is stored.

**What is auto-scaling? How does it work?**

* Horizontally Scaling
* Vertically Scaling

**Auto scaling is a feature of AWS** which allows you to configure and automatically provision and spinning up new instances without the need for your intervention. You can do this by setting thresholds and metrics to monitor. When those thresholds are crossed, a new instance of your choosing will be spun up, configured, and rolled into the load balancer pool. You’ve scaled horizontally without any operator intervention.

**Vertically Scaling:**This is an incredible feature of AWS and cloud virtualization. Spinning up a new larger instance than the one you are currently running.  Pause that instance and detach the root EBS volume from this server and discard. Then stop your live instance, detach its root volume. Note the unique device ID and attach that root volume to your new server. And the start it again. You have scaled vertically in-place.

## **Determining the ELB Option That’s Best for You**

Your mileage will vary depending on your exact situation, of course. But in general, the Classic Load Balancer is likely to be the best choice if your routing and load-balancing needs can all be handled based on IP addresses and TCP ports.

In contrast, the Application Load Balancer can address more complex load-balancing needs by managing traffic at the application level. This is especially advantageous for next-generation infrastructure, such as that based on containers, or if you are building complex web applications in which requests for certain components should be directed to one cluster, while others go to a different one.

**What automation tools can you use to spinning up the servers?**

Here below many types tools given any of the following tools can be used:

* Roll-your-own scripts, and use the AWS API tools. Such scripts could be written in bash, Perl or other language or your choice.
* Use a configuration management and provisioning tool like Ansible, puppet or its successor Opscode Chef etc.
* You might also look towards a tool like Scalr. Lastly you can go with a managed solution such as Rightscale.
* 1by default, 7 to 30 days
* yes

**How you would simulate perimeter security using amazon web services model?**

Traditional perimeter security that we’re already familiar with using firewalls and so forth is not supported in the Amazon EC2 world.

AWS supports security groups.  One can create a security group for a jump box with ssh access – only port 22 open. From there a web server group and database group are created.

The web server group allows 80 and 443 from the world, but port 22 only from the jump box group.  Further the database group allows port 3306 from the web server group and port 22 from the jump box group. Add any machines to the web server group and they can all hit the database.

No one from the world can, and no one can directly SSH to any of your boxes.

Want to further lock this configuration down?  Only allow SSH access from specific IP addresses on your network, or allow just your subnet.

**What is S3? What is it used for? Should encryption be used in S3?**

Amazon S3 is stand for Simple storage service that is storage for the Internet. It as a, “simple storage service that offers software developers a highly-scalable, reliable, and low-latency data storage infrastructure at very low costs”.

Amazon S3 provides a simple web service interface which you can use to store and retrieve any amount of data, at any time, from anywhere on the web. Using this web service, developers can easily build applications that make use of Internet storage.

You can think of it like ftp storage, where you can move files to and from there, but not mount it like a file system. AWS automatically puts your snapshots there, as well as AMIs there. Encryption should be considered for sensitive data, as S3 is a proprietary technology developed by Amazon themselves, and as yet unproven vis-a-vis a security standpoint.

Encryption should be considered for sensitive data, as S3 is a proprietary technology developed by Amazon themselves, and yet to be proven from a security standpoint.

**How is buffer used in amazon web services?**

Buffer is used to make the system more resilient to burst of traffic or load by synchronizing different components. The components always receive and process the requests in an unbalanced way. Buffer keeps the balance between different components and makes them work at the same speed to provide faster services.

**What is the function of Amazon Elastic Compute Cloud?**

Amazon Elastic compute cloud also known as Amazon EC2 is an Amazon web service that provides scalable resources and makes the computing easier for developers. The main functions of Amazon EC2 are:

* It provides easy configurable options and allow user to configure the capacity.
* It provides the complete control of computing resources and let the user run the computing environment according to his requirements.
* It provides a fast way to run the instances and quickly book the system hence reducing the overall time.
* It provides scalability to the resources and changes its environment according to the requirement of the user.
* It provides varieties of tools to the developers to build failure resilient applications.

**What is ECS?**

* Amazon EC2 Container Service (ECS) is a highly scalable container management service and high performance that supports the Docker containers and allows you to easily run applications on a cluster managed by Amazon EC2 instances.
* The EC2 service is inseparable from the concept of Amazon Machine Image - AMI. The May is Indeed the image of a virtual machine That Will Be Executed. EC2 based on XEN virtualization, that's why it is quite easy to move XEN servers to EC2.

**What are the security for amazon EC2?**

There are several best practices for secure Amazon EC2. A few of them are given below:

* Use AWS Identity and Access Management (IAM) to control access to your AWS resources.
* Restrict access by only allowing trusted hosts or networks to access ports on your instance.
* Review the rules in your security groups regularly, and ensure that you apply the principle of least
* Privilege – only open permissions that you require.
* Disable password-based logins for instances launched from your AMI. Passwords can be found or cracked, and are a security risk.

**What are the different components used in AWS?**

The components that are used in AWS are:

* Amazon S3: it is used to retrieve input data sets that are involved in making a cloud architecture and used to store the output data sets that is the result of the input.
* Amazon SQS: it is used for buffering requests that is received by the controller of the Amazon. It is the component that is used for communication between different controllers.
* Amazon Simple DB: it is used to store intermediate status log and the tasks that are performed by the user
* Amazon EC2: it is used to run a large distributed processing on the Hadoop cluster. It provides automatic parallelization and job scheduling.

**Mention what are the differences between Amazon S3 and EC2?**

**S3:** Amazon S3 is just a storage service, typically used to store large binary files. Amazon also has other storage and database services, like RDS for relational databases and DynamoDB for NoSQL.

**EC2:** An EC2 instance is like a remote computer running Windows or Linux and on which you can install whatever software you want, including a Web server running PHP code and a database server.

**Is it possible to use AWS as a web host? What are the way of using AWS as a web host?**

Yes, it is completely possible to host websites on AWS in 2 ways

* Easy – S3 (Simple Storage Solution) is a bucket storage solution that lets you serve static content e.g. images but has recently been upgraded so you can use it to host flat .html files and your site will get served by a default Apache installation with very little configuration on your part (but also little control).
* Trickier – You can use EC2 (Elastic Compute Cloud) and create a virtual Linux instance then install Apache/Nginx (or whatever) on that to give you complete control over serving whatever/however you want. You use SecurityGroups to enable/disable ports for individual machines or groups of them.

**How step you follow to make 10,000 files as public in S3?**

I will generate a bucket policy which gives access to all the files in the bucket. The bucket policy can be added to a bucket through AWS console.  
{  
“Id”: “…”,  
“Statement”: [ {  
“Sid”: “…”,  
“Action”: [  
“s3:GetObject”  
],  
“Effect”: “Allow”,  
“Resource”: “arn:aws:s3:::bucket/\*”,  
“Principal”: {  
“AWS”: [ “\*” ]  
}  
} ]  
}

**How to delete files recursively from an S3 bucket?**

* aws s3 rm –recursive s3://your\_bucket\_name/foo/
* Or delete everything under the bucket:  
  aws s3 rm –recursive s3://your\_bucket\_name
* If what you want is to actually delete the bucket, there is one-step shortcut:  
  aws s3 rb –force s3://your\_bucket\_name

**Write down the command you will use to copy all files from one S3 bucket to another with s3cmd?**

s3cmd sync s3://from/this/bucket/ s3://to/this/bucket/

**How do you see how much disk space is using by S3 bucket?**

s3cmd can show you this by running s3cmd du, optionally passing the bucket name as an argument.

**How to access/ping a server located on AWS?**

Using UI:  
**In your security group:**

* Click the inbound tab
* Create a custom ICMP rule
* Select echo request
* Use range 0.0.0.0/0 for everyone or lock it down to specific IPs
* Apply the changes and you’ll be able to ping.

**Using cmd: To do this on the command line you can run:**

* ec2-authorize -P icmp -t -1:-1 -s 0.0.0.0/0

**What happens when I reboot an EC2 instance?**

Rebooting an instance is like rebooting a PC. The hard disk isn’t affected. You don’t return to the image’s original state, but the contents of the hard disks are those before the reboot.  
Rebooting isn’t associated with billing. Billing starts when you instantiate an image and stops when you terminate it. Rebooting in between hasn’t any effect.

**What is the difference between Amazon SNS and Amazon SQS?**

* Amazon SNS allows applications to send time-critical messages to multiple subscribers through a “push” mechanism, eliminating the need to periodically check or “poll” for updates.
* Amazon SQS is a message queue service used by distributed applications to exchange messages through a polling model, and can be used to decouple sending and receiving components—without requiring each component to be concurrently available.

**How you will change the root EBS device of my amazon EC2 instance?**

* Stop the instance.
* Detach the root EBS volume.
* Attach the alternate EBS volume (as the root e.g. /dev/sda1)
* Start the instance.
* This presupposes that your alternate EBS volume is bootable, of course – it has to contain the bootable OS image.

**Can I vertically scale an Amazon instance?  How?**

Yes.  This is an incredible feature of AWS and cloud virtualization.  Spin up a new larger instance than the one you are currently running.  Pause that instance and detach the root EBS volume from this server and discard.  Then stop your live instance, detach its root volume.  Note the unique device ID and attach that root volume to your new server.   And the start it again.  Voila you have scaled vertically in-place

**What is the difference between NACL and SG?**

* NACL are applicable at the subnet level, if any instance in the subnet associated with NACL must follow the NACL rules whereas security groups are applicable at the instance level.
* In NACL we can set both allow and deny rules for instance, whereas in SG we can only set allow rules for instance, by default everything is denied.
* SG evaluates everything before allowing the traffic. But it’s not the same case while coming to NACL. It first check the deny rules and the check the allow rules.

**What is EBS (Elastic Block Storage)?  What type of performance can you expect?  How do you back it up?  How do you improve performance?**

* EBS is a virtualized SAN or storage area network. Elastic Block Store (Amazon EBS) provides persistence block level storage volumes for use with EC2 instances. EBS volumes are highly available and reliable storage volumes that can be attached to any running instance that is in the same Availability Zone.
* **Performance that we can expect:**Performance on EBS can exhibit variability. That is it can go above the SLA performance level, then drop below it. The SLA provides you with an average disk I/O rate you can expect. This can frustrate some folks especially performance experts who expect reliable and consistent disk throughput on a server. Traditional physically hosted servers behave that way. Virtual AWS instances do not.

**Amazon EBS offering high availability & durability. And it offers the consistent & low-latency performance needed to run your workloads.**

* **EBS Magnetic volumes:**You can create EBS Magnetic volumes from 1 GB to 1 TB in size
* **EBS General Purpose SSD (gp2):** You can create **EBS General Purpose SSD (1G–16TB )**
* **Provisioned IOPS SSD (io1**)**:**Highest-performance SSD volume designed for mission-critical applications (4 GB – 16 TB)
* **Cold HDD (sc1):**Lowest cost HDD volume designed for less frequently accessed workloads (**500 GB – 16 TB**)
* **Amazon EBS Encryption:** You can use encrypted EBS volumes to meet a wide range of data-at-rest encryption requirements for regulated/audited data and applications.
* **Amazon EBS Snapshots:**You can create point-in-time snapshots of EBS volumes, which are persisted to Amazon S3. Snapshots protect data for long-term durability, and they can be used as the starting point for new EBS volumes. The same snapshot can be used to instantiate as many volumes as you wish. These snapshots can be copied across AWS regions.
* Performance metrics, such as bandwidth, throughput, latency, and average queue length, are available through the AWS Management Console. These**metrics, provided by AmazonCloudWatch,** allow you to monitor the performance of your volumes to make sure that you are providing enough performance for your applications without paying for resources you don’t need.

**What is cfn-init?**

It is a script that reads the template metadata from AWS::CloudFormation::init key and used to

1. Fetch data from cloud formation
2. Install packages
3. Write files to disk
4. Enable/disable, start/stop service

If you use cfn-init to update an existing file it creates a backup file with .bak extension.

**I have some private servers on my premises, also I have distributed some of my workload on the public cloud, what is this architecture called?**

Hybrid Cloud

**Explanation:** This type of architecture would be a hybrid cloud. Why? Because we are using both, the public cloud, and you’re on premises servers i.e. the private cloud. To make this hybrid architecture easy to use, wouldn’t it be better if your private and public cloud were all on the same network(virtually). This is established by including your public cloud servers in a virtual private cloud, and connecting this virtual cloud with your on-premise servers using a VPN (Virtual Private Network).

**What does the following command do with respect to the Amazon EC2 security groups?**

**ec2-create-group Create Security Group**

Creates a new security group for use with your account.

**Explanation:**A Security group is just like a firewall, it controls the traffic in and out of your instance. In AWS terms, the inbound and outbound traffic. The command mentioned is pretty straight forward, it says create security group, and does the same. Moving along, once your security group is created, you can add different rules in it. For example, you have an RDS instance, to access it, you must add the public IP address of the machine from which you want access the instance in its security group.

**You have a video trans-coding application. The videos are processed according to a queue. If the processing of a video is interrupted in one instance, it is resumed in another instance. Currently there is a huge back-log of videos which needs to be processed, for this you need to add more instances, but you need these instances only until your backlog is reduced. Which of these would be an efficient way to do it?**

You should be using an **On-Demand** instance for the same. Why? First, the workload has to be processed now, meaning it is urgent, secondly you don’t need them once your backlog is cleared, therefore Reserved Instance is out of the picture, and since the work is urgent, you cannot stop the work on your instance just because the spot price spiked, therefore Spot Instances shall also not be used. Hence On-Demand instances shall be the right choice in this case.

**You have a distributed application that periodically processes large volumes of data across multiple Amazon EC2 Instances. The application is designed to recover gracefully from Amazon EC2 instance failures. You are required to accomplish this task in the most cost-effective way.**

**Which of the following will meet your requirements?**

Spot Instances

**Explanation:** Since the work we are addressing here is not continuous, a reserved instance shall be idle at times, same goes with On-Demand instances. Also, it does not make sense to launch an On-Demand instance whenever work comes up, since it is expensive. Hence Spot Instances will be the right fit because of their low rates and no long-term commitments.

**How is stopping and terminating an instance different from each other?**

Starting, stopping and terminating are the three states in an EC2 instance, let’s discuss them in detail:

* **Stopping and Starting** an instance: When an instance is stopped, the instance performs a normal shutdown and then transitions to a stopped state. All of its Amazon EBS volumes remain attached, and you can start the instance again at a later time. You are not charged for additional instance hours while the instance is in a stopped state.
* **Terminating** an instance: When an instance is terminated, the instance performs a normal shutdown, then the attached Amazon EBS volumes are deleted unless the volume’s *deleteOnTermination* attribute is set to false. The instance itself is also deleted, and you can’t start the instance again at a later time.

**If I want my instance to run on a single-tenant hardware, which value do I have to set the instance’s tenancy attribute to?**

Dedicated

**Explanation:** The Instance tenancy attribute should be set to Dedicated Instance. The rest of the values are invalid.

**When will you incur costs with an Elastic IP address (EIP)?**

When it is allocated and associated with a stopped instance.

**Explanation:** You are not charged, if only one Elastic IP address is attached with your running instance. But you do get charged in the following conditions:

* When you use more than one Elastic IPs with your instance.
* When your Elastic IP is attached to a stopped instance.
* When your Elastic IP is not attached to any instance.

**How is a Spot instance different from an On-Demand instance or Reserved Instance?**

First of all, let’s understand that Spot Instance, On-Demand instance and Reserved Instances are all models for pricing. Moving along, spot instances provide the ability for customers to purchase compute capacity with no upfront commitment, at hourly rates usually lower than the On-Demand rate in each region. Spot instances are just like bidding, the bidding price is called Spot Price. The Spot Price fluctuates based on supply and demand for instances, but customers will never pay more than the maximum price they have specified. If the Spot Price moves higher than a customer’s maximum price, the customer’s EC2 instance will be shut down automatically. But the reverse is not true, if the Spot prices come down again, your EC2 instance will not be launched automatically, one has to do that manually.  In Spot and On-demand instance, there is no commitment for the duration from the user side, however in reserved instances one has to stick to the time period that he has chosen.

**Are the Reserved Instances available for Multi-AZ Deployments?**

Available for all instance types

**Explanation:** Reserved Instances is a pricing model, which is available for all instance types in EC2.

**How to use the processor state control feature available on the c4.8xlarge instance?**

The processor state control consists of 2 states:

* The C state – Sleep state varying from c0 to c6. C6 being the deepest sleep state for a processor
* The P state – Performance state p0 being the highest and p15 being the lowest possible frequency.

Now, why the C state and P state. Processors have cores, these cores need thermal headroom to boost their performance. Now since all the cores are on the processor the temperature should be kept at an optimal state so that all the cores can perform at the highest performance.

Now how will these states help in that? If a core is put into sleep state it will reduce the overall temperature of the processor and hence other cores can perform better. Now the same can be synchronized with other cores, so that the processor can boost as many cores it can by timely putting other cores to sleep, and thus get an overall performance boost.

Concluding, the C and P state can be customized in some EC2 instances like the c4.8xlarge instance and thus you can customize the processor according to your workload.

**What kind of network performance parameters can you expect when you launch instances in cluster placement group?**

The network performance depends on the instance type and network performance specification, if launched in a placement group you can expect up to

* 10 Gbps in a single-flow,
* 20 Gbps in multiflow i.e., full duplex
* Network traffic outside the placement group will be limited to 5 Gbps (full duplex).

**To deploy a 4-node cluster of Hadoop in AWS which instance type can be used?**

First let’s understand what happens in a Hadoop cluster, the Hadoop cluster follows a master slave concept. The master machine processes all the data, slave machines store the data and act as data nodes. Since all the storage happens at the slave, a higher capacity hard disk would be recommended and since master does all the processing, a higher RAM and a much better CPU is required. Therefore, you can select the configuration of your machine depending on your workload. For e.g. – In this case c4.8xlarge will be preferred for master machine whereas for slave machine we can select i2.large instance. If you don’t want to deal with configuring your instance and installing Hadoop cluster manually, you can straight away launch an Amazon EMR (Elastic Map Reduce) instance which automatically configures the servers for you. You dump your data to be processed in S3, EMR picks it from there, processes it, and dumps it back into S3.

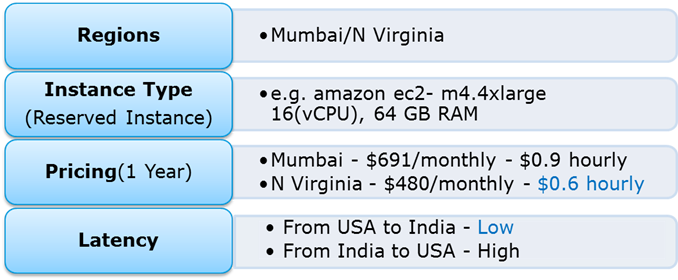
**Where do you think an AMI fits, when you are designing an architecture for a solution?**

AMIs (Amazon Machine Images) are like templates of virtual machines and an instance is derived from an AMI. AWS offers pre-baked AMIs which you can choose while you are launching an instance, some AMIs are not free, therefore can be bought from the AWS Marketplace. You can also choose to create your own custom AMI which would help you save space on AWS. For example, if you don’t need a set of software on your installation, you can customize your AMI to do that. This makes it cost efficient, since you are removing the unwanted things.

**How do you choose an Availability Zone?**

Let’s understand this through an example, consider there’s a company which has user base in India as well as in the US.

Let us see how we will choose the region for this use case:



So, with reference to the above figure the regions to choose between are, Mumbai and North Virginia. Now let us first compare the pricing, you have hourly prices, which can be converted to your per month figure. Here North Virginia emerges as a winner. But, pricing cannot be the only parameter to consider. Performance should also be kept in mind hence, let’s look at latency as well. Latency basically is the time that a server takes to respond to your requests i.e. the response time. North Virginia wins again!

So, concluding, North Virginia should be chosen for this use case.

**Is one Elastic IP address enough for every instance that I have running?**

Depends! Every instance comes with its own private and public address. The private address is associated exclusively with the instance and is returned to Amazon EC2 only when it is stopped or terminated. Similarly, the public address is associated exclusively with the instance until it is stopped or terminated. However, this can be replaced by the Elastic IP address, which stays with the instance if the user doesn’t manually detach it. But what if you are hosting multiple websites on your EC2 server, in that case you may require more than one Elastic IP address.

**What are the best practices for Security in Amazon EC2?**

There are several best practices to secure Amazon EC2. A few of them are given below:

* Use AWS Identity and Access Management (IAM) to control access to your AWS resources.
* Restrict access by only allowing trusted hosts or networks to access ports on your instance.
* Review the rules in your security groups regularly, and ensure that you apply the principle of least
* Privilege – only open up permissions that you require.
* Disable password-based logins for instances launched from your AMI. Passwords can be found or cracked, and are a security risk.

**You need to configure an Amazon S3 bucket to serve static assets for your public-facing web application. Which method will ensure that all objects uploaded to the bucket are set to public read?**

Configure the bucket policy to set all objects to public read.

**Explanation:** Rather than making changes to every object, it’s better to set the policy for the whole bucket. IAM is used to give more granular permissions, since this is a website, all objects would be public by default.

**A customer wants to leverage Amazon Simple Storage Service (S3) and Amazon Glacier as part of their backup and archive infrastructure. The customer plans to use third-party software to support this integration. Which approach will limit the access of the third-party software to only the Amazon S3 bucket named “company-backup”?**

A custom IAM user policy limited to the Amazon S3 API in “company-backup”.

**Explanation:** Taking queue from the previous questions, this use case involves more granular permissions, hence IAM would be used here.

**Can S3 be used with EC2 instances, if yes, how?**

Yes, it can be used for instances with root devices backed by local instance storage. By using Amazon S3, developers have access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of web sites. In order to execute systems in the Amazon EC2 environment, developers use the tools provided to load their Amazon Machine Images (AMIs) into Amazon S3 and to move them between Amazon S3 and Amazon EC2.

Another use case could be for websites hosted on EC2 to load their static content from S3.

**A customer implemented AWS Storage Gateway with a gateway-cached volume at their head office. An event takes the link between the main and branch office offline. Which methods will enable the branch office to access their data?**

Launch a new AWS Storage Gateway instance AMI in Amazon EC2, and restore from a gateway snapshot.

**Explanation:** The fastest way to do it would be launching a new storage gateway instance. Why? Since time is the key factor which drives every business, troubleshooting this problem will take more time. Rather than we can just restore the previous working state of the storage gateway on a new instance.

**When you need to move data over long distances using the internet, for instance across countries or continents to your Amazon S3 bucket, which method or service will you use?**

Amazon Transfer Acceleration

**Explanation:** You would not use Snowball, because for now, the snowball service does not support cross region data transfer, and since, we are transferring across countries, Snowball cannot be used. Transfer Acceleration shall be the right choice here as it throttles your data transfer with the use of optimized network paths and Amazon’s content delivery network up to 300% compared to normal data transfer speed.

**How can you speed up data transfer in Snowball?**

The data transfer can be increased in the following way:

* By performing multiple copy operations at one time i.e. if the workstation is powerful enough, you can initiate multiple cp commands each from different terminals, on the same Snowball device.
* Copying from multiple workstations to the same snowball.
* Transferring large files or by creating a batch of small file, this will reduce the encryption overhead.
* Eliminating unnecessary hops i.e. make a setup where the source machine(s) and the snowball are the only machines active on the switch being used, this can hugely improve performance.

**If you want to launch Amazon Elastic Compute Cloud (EC2) instances and assign each instance a predetermined private IP address you should:**

Launch the instances in the Amazon Virtual Private Cloud (VPC).

**Explanation:** The best way of connecting to your cloud resources (for ex- ec2 instances) from your own data center (for ex.- private cloud) is a VPC. Once you connect your datacenter to the VPC in which your instances are present, each instance is assigned a private IP address which can be accessed from your datacenter. Hence, you can access your public cloud resources, as if they were on your own network.

**Can I connect my corporate datacenter to the Amazon Cloud?**

Yes, you can do this by establishing a VPN (Virtual Private Network) connection between your company’s network and your VPC (Virtual Private Cloud), this will allow you to interact with your EC2 instances as if they were within your existing network.

**Is it possible to change the private IP addresses of an EC2 while it is running/stopped in a VPC?**

Primary private IP address is attached with the instance throughout its lifetime and cannot be changed, however secondary private addresses can be unassigned, assigned or moved between interfaces or instances at any point.

**Why do you make subnets?**

To efficiently utilize networks that have a large no. of hosts.

**Explanation:** If there is a network which has a large no. of hosts, managing all these hosts can be a tedious job. Therefore, we divide this network into subnets (sub-networks) so that managing these hosts becomes simpler.

**Which of the following is true?**

You can attach multiple subnets to a route table

**Explanation:** Route Tables are used to route network packets, therefore in a subnet having multiple route tables will lead to confusion as to where the packet has to go. Therefore, there is only one route table in a subnet, and since a route table can have any no. of records or information, hence attaching multiple subnets to a route table is possible.

**In CloudFront what happens when content is NOT present at an Edge location and a request is made to it?**

CloudFront delivers the content directly from the origin server and stores it in the cache of the edge location

**Explanation:** CloudFront is a content delivery system, which caches data to the nearest edge location from the user, to reduce latency. If data is not present at an edge location, the first time the data may get transferred from the original server, but from the next time, it will be served from the cached edge.

**If I’m using Amazon CloudFront, can I use Direct Connect to transfer objects from my own data center?**

Yes. Amazon CloudFront supports custom origins including origins from outside of AWS. With AWS Direct Connect, you will be charged with the respective data transfer rates.

**If my AWS Direct Connect fails, will I lose my connectivity?**

If a backup AWS Direct connect has been configured, in the event of a failure it will switch over to the second one. It is recommended to enable Bidirectional Forwarding Detection (BFD) when configuring your connections to ensure faster detection and failover. On the other hand, if you have configured a backup IPsec VPN connection instead, all VPC traffic will failover to the backup VPN connection automatically. Traffic to/from public resources such as Amazon S3 will be routed over the Internet. If you do not have a backup AWS Direct Connect link or a IPsec VPN link, then Amazon VPC traffic will be dropped in the event of a failure.

**If I launch a standby RDS instance, will it be in the same Availability Zone as my primary?**

No

**Explanation:** No, since the purpose of having a standby instance is to avoid an infrastructure failure (if it happens), therefore the standby instance is stored in a different availability zone, which is a physically different independent infrastructure.

**When would I prefer Provisioned IOPS over Standard RDS storage?**

If you have batch-oriented workloads

**Explanation:**  Provisioned IOPS deliver high IO rates but on the other hand it is expensive as well. Batch processing workloads do not require manual intervention they enable full utilization of systems, therefore a provisioned IOPS will be preferred for batch oriented workload.

**How is Amazon RDS, DynamoDB and Redshift different?**

* Amazon RDS is a database management service for relational databases, it manages patching, upgrading, backing up of data etc. of databases for you without your intervention. RDS is a Db management service for structured data only.
* DynamoDB, on the other hand, is a NoSQL database service, NoSQL deals with unstructured data.
* Redshift, is an entirely different service, it is a data warehouse product and is used in data analysis.

**If I am running my DB Instance as a Multi-AZ deployment, can I use the standby DB Instance for read or write operations along with primary DB instance?**

No

**Explanation:** No,Standby DB instance cannot be used with primary DB instance in parallel, as the former is solely used for standby purposes, it cannot be used unless the primary instance goes down.

**Your company’s branch offices are all over the world, they use a software with a multi-regional deployment on AWS, they use MySQL 5.6 for data persistence.**

**The task is to run an hourly batch process and read data from every region to compute cross-regional reports which will be distributed to all the branches. This should be done in the shortest time possible. How will you build the DB architecture in order to meet the requirements?**

For each regional deployment, use RDS MySQL with a master in the region and a read replica in the HQ region

**Explanation:** For this we will take an RDS instance as a master, because it will manage our database for us and since we should read from every region, we’ll put a read replica of this instance in every region where the data has to be read from. It is not correct since putting a read replica would be more efficient than putting a snapshot, a read replica can be promoted if needed to an independent DB instance, but with a Db snapshot it becomes mandatory to launch a separate DB Instance.

**Can I run more than one DB instance for Amazon RDS for free?**

Yes. You can run more than one Single-AZ Micro database instance, that too for free! However, any use exceeding 750 instance hours, across all Amazon RDS Single-AZ Micro DB instances, across all eligible database engines and regions, will be billed at standard Amazon RDS prices. For example: if you run two Single-AZ Micro DB instances for 400 hours each in a single month, you will accumulate 800 instance hours of usage, of which 750 hours will be free. You will be billed for the remaining 50 hours at the standard Amazon RDS price.

**Which AWS services will you use to collect and process e-commerce data for near real-time analysis?**

Amazon DynamoDB and Amazon Redshift

**Explanation:** DynamoDB is a fully managed NoSQL database service. DynamoDB, therefore can be fed any type of unstructured data, which can be data from e-commerce websites as well, and later, an analysis can be done on them using Amazon Redshift. We are not using Elastic MapReduce, since a near real time analyses is needed.

**Can I retrieve only a specific element of the data, if I have a nested JSON data in DynamoDB?**

Yes. When using the GetItem, BatchGetItem, Query or Scan APIs, you can define a Projection Expression to determine which attributes should be retrieved from the table. Those attributes can include scalars, sets, or elements of a JSON document.

**A company is deploying a new two-tier web application in AWS. The company has limited staff and requires high availability, and the application requires complex queries and table joins. Which configuration provides the solution for the company’s requirements?**

Amazon DynamoDB

**Explanation:** DynamoDB can scale more than RDS or any other relational database service, therefore DynamoDB would be the apt choice.

**What happens to my backups and DB Snapshots if I delete my DB Instance?**

When you delete a DB instance, you have an option of creating a final DB snapshot, if you do that you can restore your database from that snapshot. RDS retains this user-created DB snapshot along with all other manually created DB snapshots after the instance is deleted, also automated backups are deleted and only manually created DB Snapshots are retained.

**Which of the following use cases are suitable for Amazon DynamoDB? Choose 2 answers**

Storing metadata for Amazon S3 objects and Running relational joins and complex updates.

**Explanation:** If all your JSON data have the same fields ex: [id, name, age] then it would be better to store it in a relational database, the metadata on the other hand is unstructured, also running relational joins or complex updates would work on DynamoDB as well.

**How can I load my data to Amazon Redshift from different data sources like Amazon RDS, Amazon DynamoDB and Amazon EC2?**

You can load the data in the following two ways:

* You can use the COPY command to load data in parallel directly to Amazon Redshift from Amazon EMR, Amazon DynamoDB, or any SSH-enabled host.
* AWS Data Pipeline provides a high performance, reliable, fault tolerant solution to load data from a variety of AWS data sources. You can use AWS Data Pipeline to specify the data source, desired data transformations, and then execute a pre-written import script to load your data into Amazon Redshift.

**Your application must retrieve data from your user’s mobile every 5 minutes and the data is stored in DynamoDB, later every day at a time the data is extracted into S3 on a per user basis and then your application is later used to visualize the data to the user. You are asked to optimize the architecture of the backend system to lower cost, what would you recommend?**

Introduce Amazon ElastiCache to cache reads from the Amazon DynamoDB table and reduce provisioned read throughput.

**Explanation:** Since our work requires the data to be extracted and analyzed, to optimize this process a person would use provisioned IO, but since it is expensive, using a ElastiCache memoryinsread to cache the results in the memory can reduce the provisioned read throughput and hence reduce cost without affecting the performance.

**You are running a website on EC2 instances deployed across multiple Availability Zones with a Multi-AZ RDS MySQL Extra Large DB Instance. The site performs a high number of small reads and writes per second and relies on an eventual consistency model. After comprehensive tests, you discover that there is read contention on RDS MySQL. Which are the best approaches to meet these requirements? (Choose 2 answers)**

Deploy ElastiCache in-memory cache running in each availability zone

Increase the RDS MySQL Instance size and Implement provisioned IOPS

**Explanation:**Since it does a lot of read writes, provisioned IO may become expensive. But we need high performance as well, therefore the data can be cached using ElastiCache which can be used for frequently reading the data. As for RDS since read contention is happening, the instance size should be increased and provisioned IO should be introduced to increase the performance.

**A startup is running a pilot deployment of around 100 sensors to measure street noise and air quality in urban areas for 3 months. It was noted that every month around 4GB of sensor data is generated. The company uses a load balanced auto scaled layer of EC2 instances and a RDS database with 500 GB standard storage. The pilot was a success and now they want to deploy at least 100K sensors which need to be supported by the backend. You need to store the data for at least 2 years to analyze it. Which setup of the following would you prefer?**

Replace the RDS instance with a 6 node Redshift cluster with 96TB of storage

**Explanation:** A Redshift cluster would be preferred because it easy to scale, also the work would be done in parallel through the nodes, therefore is perfect for a bigger workload like our use case. Since each month 4 GB of data is generated, therefore in 2 years, it should be around 96 GB. And since the servers will be increased to 100K in number, 96 GB will approximately become 96TB. Hence option C is the right answer.

**Suppose you have an application where you have to render images and also do some general computing. From the following services which service will best fit your need?**

Application Load Balancer

**Explanation:** You will choose an application load balancer, since it supports path based routing, which means it can take decisions based on the URL, therefore if your task needs image rendering it will route it to a different instance, and for general computing it will route it to a different instance.

**What is the difference between Scalability and Elasticity?**

Scalability is the ability of a system to increase its hardware resources to handle the increase in demand. It can be done by increasing the hardware specifications or increasing the processing nodes.

Elasticity is the ability of a system to handle increase in the workload by adding additional hardware resources when the demand increases (same as scaling) but also rolling back the scaled resources, when the resources are no longer needed. This is particularly helpful in Cloud environments, where a pay per use model is followed.

**How will you change the instance type for instances which are running in your application tier and are using Auto Scaling. Where will you change it from the following areas?**

Auto Scaling launch configuration

**Explanation:** Auto scaling tags configuration, is used to attach metadata to your instances, to change the instance type you have to use auto scaling launch configuration.

**You have a content management system running on an Amazon EC2 instance that is approaching 100% CPU utilization. Which option will reduce load on the Amazon EC2 instance?**

Create a load balancer, and register the Amazon EC2 instance with it

**Explanation:** Creating alone an auto-scaling group will not solve the issue, until you attach a load balancer to it. Once you attach a load balancer to an auto-scaling group, it will efficiently distribute the load among all the instances. Option B – CloudFront is a CDN, it is a data transfer tool therefore will not help reduce load on the EC2 instance. Similarly, the other option – Launch configuration is a template for configuration which has no connection with reducing loads.

**When should I use a Classic Load Balancer and when should I use an Application load balancer?**

A Classic Load Balancer is ideal for simple load balancing of traffic across multiple EC2 instances, while an Application Load Balancer is ideal for microservices or container-based architectures where there is a need to route traffic to multiple services or load balance across multiple ports on the same EC2 instance.

**What does Connection draining do?**

 Re-routes traffic from instances which are to be updated or failed a health check.

**Explanation:** Connection draining is a service under ELB which constantly monitors the health of the instances. If any instance fails a health check or if any instance has to be patched with a software update, it  pulls all the traffic from that instance and re-routes them to other instances.

**When an instance is unhealthy, it is terminated and replaced with a new one, which of the following services does that?**

Fault Tolerance

**Explanation:** When ELB detects that an instance is unhealthy, it starts routing incoming traffic to other healthy instances in the region. If all the instances in a region becomes unhealthy, and if you have instances in some other availability zone/region, your traffic is directed to them. Once your instances become healthy again, they are re-routed back to the original instances.

**What are lifecycle hooks used for in Auto-Scaling?**

 They are used to put an additional wait time to a scale in or scale out event.

**Explanation:** Lifecycle hooks are used for putting wait time before any lifecycle action i.e., launching or terminating an instance happens. The purpose of this wait time, can be anything from extracting log files before terminating an instance or installing the necessary software’s in an instance before launching it.

**A user has setup an Auto Scaling group. Due to some issue, the group has failed to launch a single instance for more than 24 hours. What will happen to Auto Scaling in this condition?**

Auto Scaling will suspend the scaling process

**Explanation:** Auto Scaling allows you to suspend and then resume one or more of the Auto Scaling processes in your Auto Scaling group. This can be very useful when you want to investigate a configuration problem or other issue with your web application, and then make changes to your application, without triggering the Auto Scaling process.

**You have an EC2 Security Group with several running EC2 instances. You changed the Security Group rules to allow inbound traffic on a new port and protocol, and then launched several new instances in the same Security Group. The new rules apply:**

Immediately to all instances in the security group.

**Explanation:** Any rule specified in an EC2 Security Group applies immediately to all the instances, irrespective of when they are launched before or after adding a rule.

**To create a mirror image of your environment in another region for disaster recovery, which of the following AWS resources do not need to be recreated in the second region? ( Choose 2 answers )**

Route 53 Record Sets and Elastic IP Addresses (EIP)

**Explanation:** Elastic IPs and Route 53 record sets are common assets therefore there is no need to replicate them, since Elastic IPs and Route 53 are valid across regions

**A customer wants to capture all client connection information from his load balancer at an interval of 5 minutes, which of the following options should he choose for his application?**

Enable AWS CloudTrail for the load-balancer.

**Explanation:** AWS CloudTrail provides inexpensive logging information for load balancer and other AWS resources. This logging information can be used for analyses and other administrative work, therefore is perfect for this use case.

**A customer wants to track access to their Amazon Simple Storage Service (S3) buckets and also use this information for their internal security and access audits. Which of the following will meet the Customer requirement?**

Enable AWS CloudTrail to audit all Amazon S3 bucket access.

**Explanation:** AWS CloudTrail has been designed for logging and tracking API calls. Also, this service is available for storage, therefore should be used in this use case.

**Which of the following are true regarding AWS CloudTrail? (Choose 2 answers)**

CloudTrail is enabled on a per-region and service basis

Logs can be delivered to a single Amazon S3 bucket for aggregation.

**Explanation:** CloudTrail is not enabled for all the services and is also not available for all the regions. Therefore option B is correct, also the logs can be delivered to your S3 bucket, hence C is also correct.

**What happens if CloudTrail is turned on for my account but my Amazon S3 bucket is not configured with the correct policy?**

CloudTrail files are delivered according to S3 bucket policies. If the bucket is not configured or is misconfigured, CloudTrail might not be able to deliver the log files.

**How do I transfer my existing domain name registration to Amazon Route 53 without disrupting my existing web traffic?**

You will need to get a list of the DNS record data for your domain name first, it is generally available in the form of a “zone file” that you can get from your existing DNS provider. Once you receive the DNS record data, you can use Route 53’s Management Console or simple web-services interface to create a hosted zone that will store your DNS records for your domain name and follow its transfer process. It also includes steps such as updating the nameservers for your domain name to the ones associated with your hosted zone. For completing the process, you have to contact the registrar with whom you registered your domain name and follow the transfer process. As soon as your registrar propagates the new name server delegations, your DNS queries will start to get answered.

**Which of the following services you would not use to deploy an app?**

Lambda

**Explanation:** Lambda is used for running server-less applications. It can be used to deploy functions triggered by events. When we say server-less, we mean without you worrying about the computing resources running in the background. It is not designed for creating applications which are publicly accessed.

**How does Elastic Beanstalk apply updates?**

By having a duplicate ready with updates before swapping.

**Explanation:** Elastic Beanstalk prepares a duplicate copy of the instance, before updating the original instance, and routes your traffic to the duplicate instance, so that, in case your updated application fails, it will switch back to the original instance, and there will be no downtime experienced by the users who are using your application.

**How is AWS Elastic Beanstalk different than AWS OpsWorks?**

AWS Elastic Beanstalk is an application management platform while OpsWorks is a configuration management platform. BeanStalk is an easy to use service which is used for deploying and scaling web applications developed with Java, .Net, PHP, Node.js, Python, Ruby, Go and Docker. Customers upload their code and Elastic Beanstalk automatically handles the deployment. The application will be ready to use without any infrastructure or resource configuration.

In contrast, AWS OpsWorks is an integrated configuration management platform for IT administrators or DevOps engineers who want a high degree of customization and control over operations.

**What happens if my application stops responding to requests in beanstalk?**

AWS Beanstalk applications have a system in place for avoiding failures in the underlying infrastructure. If an Amazon EC2 instance fails for any reason, Beanstalk will use Auto Scaling to automatically launch a new instance. Beanstalk can also detect if your application is not responding on the custom link, even though the infrastructure appears healthy, it will be logged as an environmental event ( ex: a bad version was deployed) so you can take an appropriate action.

**How is AWS OpsWorks different than AWS CloudFormation?**

OpsWorks and CloudFormation both support application modelling, deployment, configuration, management and related activities. Both support a wide variety of architectural patterns, from simple web applications to highly complex applications. AWS OpsWorks and AWS CloudFormation differ in abstraction level and areas of focus.

AWS CloudFormation is a building block service which enables customer to manage almost any AWS resource via JSON-based domain specific language. It provides foundational capabilities for the full breadth of AWS, without prescribing a particular model for development and operations. Customers define templates and use them to provision and manage AWS resources, operating systems and application code.

In contrast, AWS OpsWorks is a higher-level service that focuses on providing highly productive and reliable DevOps experiences for IT administrators and ops-minded developers. To do this, AWS OpsWorks employs a configuration management model based on concepts such as stacks and layers, and provides integrated experiences for key activities like deployment, monitoring, auto-scaling, and automation. Compared to AWS CloudFormation, AWS OpsWorks supports a narrower range of application-oriented AWS resource types including Amazon EC2 instances, Amazon EBS volumes, Elastic IPs, and Amazon CloudWatch metrics.

**I created a key in Oregon region to encrypt my data in North Virginia region for security purposes. I added two users to the key and an external AWS account. I wanted to encrypt an object in S3, so when I tried, the key that I just created was not listed.  What could be the reason?**

The Key should be in the same region.

**Explanation:** The key created and the data to be encrypted should be in the same region. Hence the approach taken here to secure the data is incorrect.

**A company needs to monitor the read and write IOPS for their AWS MySQL RDS instance and send real-time alerts to their operations team. Which AWS services can accomplish this?**

Amazon CloudWatch

**Explanation:** Amazon CloudWatch is a cloud monitoring tool and hence this is the right service for the mentioned use case. The other options listed here are used for other purposes for example route 53 is used for DNS services, therefore CloudWatch will be the apt choice.

**What happens when one of the resources in a stack cannot be created successfully in AWS OpsWorks?**

**Instance profile can or policies**

When an event like this occurs, the “automatic rollback on error” feature is enabled, which causes all the AWS resources which were created successfully till the point where the error occurred to be deleted. This is helpful since it does not leave behind any erroneous data, it ensures the fact that stacks are either created fully or not created at all. It is useful in events where you may accidentally exceed your limit of the no. of Elastic IP addresses or maybe you may not have access to an EC2 AMI that you are trying to run etc.

**What automation tools can you use to spinning up the servers?**

Any of the following tools can be used:

* Roll-your-own scripts, and use the AWS API tools.  Such scripts could be written in bash, Perl or other language of your choice.
* Use a configuration management and provisioning tool like puppet or its successor Opscode Chef.  You can also use a tool like Scalr.
* Use a managed solution such as Rightscale.

**What is OpenStack?**

* OpenStack is often called Cloud Operating System, and that is not far from the truth. It is the complete environment for deploying IaaS which gives you possibility of making your own cloud similar to AWS. It is highly modular and consists of many sub-projects so you can pick and choose which functionality you need. OpenStack distribution are available from Red Hat, Mirantis, HPE, Oracle, Canonical and many others. It is completely open source project but some vendors make proprietary distributions.

**Classify Cloud Platforms anategory?**

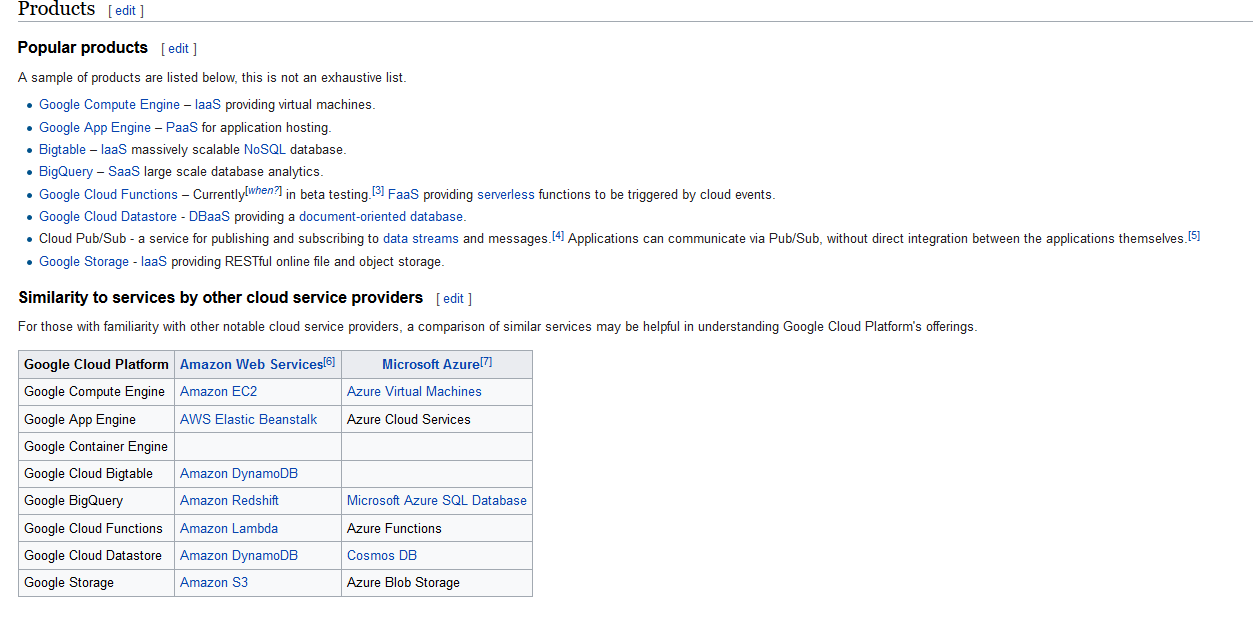
* Cloud Computing software can be classified as Software as a Service or SaaS, Infrastructure as a Service or IaaS and Platform as a Service or PaaS.
* SaaS is peace of software that runs over network on remote server and has only user interface exposed to users, usually in web browser. For example, salesforce.com.
* Infrastructure as a service is a cloud environment that exposes VM to user to use as entire OS or container where you could install anything you would install on your server.
* Example for this would-be OpenStack, AWS, Eucalyptus.

PaaS allows users to deploy their own application on the preinstalled platform, usually framework of application server and suite of developer tools. Examples for this would be OpenShHeroku.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Google Cloud\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Benefits of google cloud:

* Future proof infrastructure
* Powerful data & analytics
* Serverless, fully managed computing
* Customer friendly pricing
* Security at scale
* Data center innovation

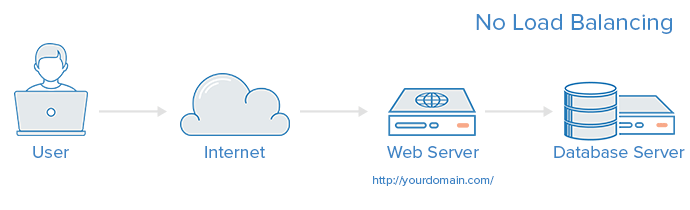


**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AWS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Load balancers**

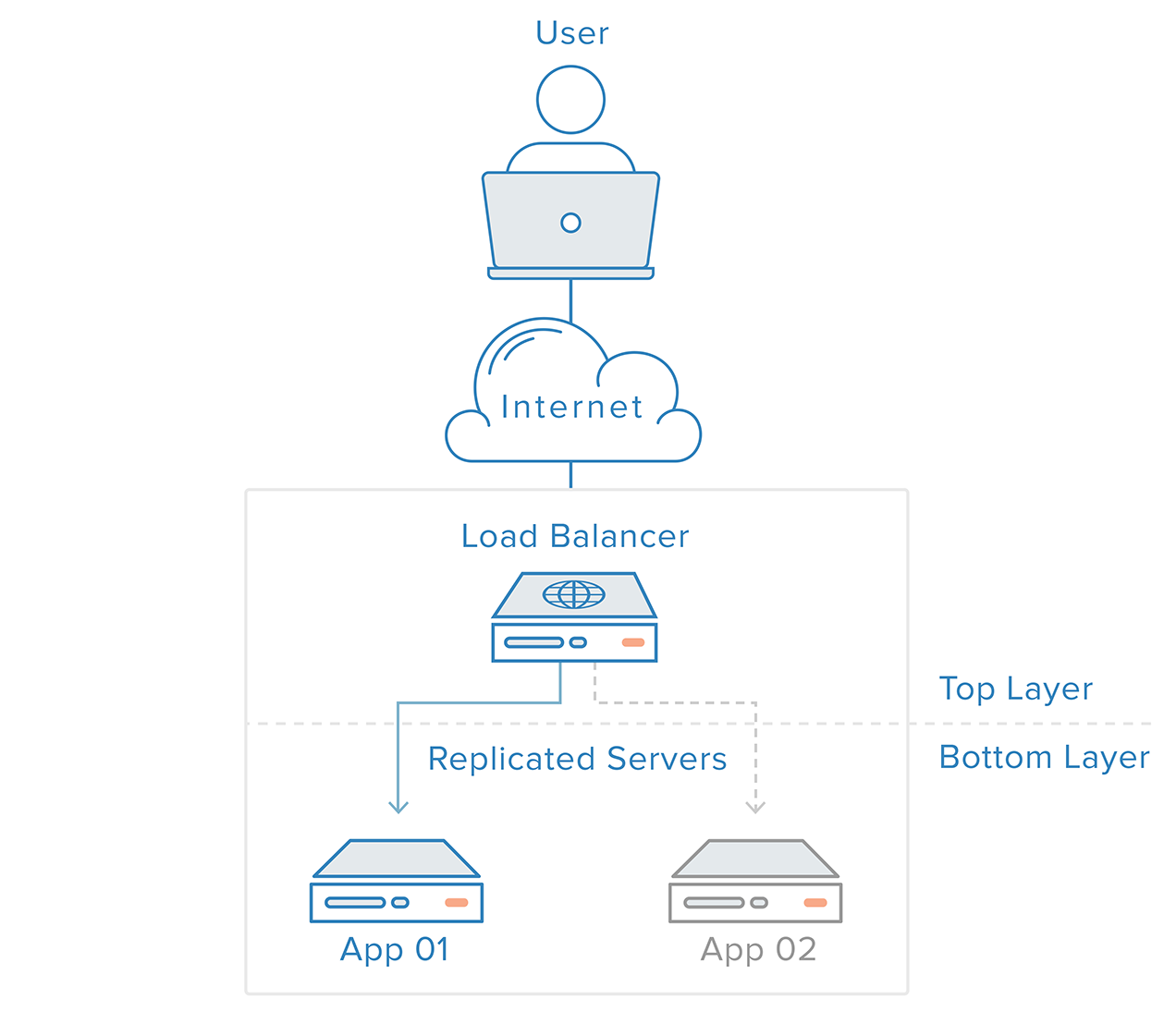
Load balancing is a key component of highly-available infrastructures commonly used to improve the performance and reliability of web sites, applications, databases and other services by distributing the workload across multiple servers.

A web infrastructure with no load balancing might look something like the following:



In this example, the user connects directly to the web server, at yourdomain.com. If this single web server goes down, the user will no longer be able to access the website. In addition, if many users try to access the server simultaneously and it is unable to handle the load, they may experience slow load times or may be unable to connect at all.

This single point of failure can be mitigated by introducing a load balancer and at least one additional web server on the backend. Typically, all of the backend servers will supply identical content so that users receive consistent content regardless of which server responds.



In the example illustrated above, the user accesses the load balancer, which forwards the user's request to a backend server, which then responds directly to the user's request. In this scenario, the single point of failure is now the load balancer itself. This can be mitigated by introducing a second load balancer, but before we discuss that, let's explore how load balancers work.

## **What kind of traffic can load balancers handle?**

Load balancer administrators create forwarding rules for four main types of traffic:

* **HTTP** — Standard HTTP balancing directs requests based on standard HTTP mechanisms. The Load Balancer sets the X-Forwarded-For, X-Forwarded-Proto, and X-Forwarded-Port headers to give the backends information about the original request.
* **HTTPS** — HTTPS balancing functions the same as HTTP balancing, with the addition of encryption. Encryption is handled in one of two ways: either with **SSL passthrough** which maintains encryption all the way to the backend or with **SSL termination** which places the decryption burden on the load balancer but sends the traffic unencrypted to the back end.
* **TCP** — For applications that do not use HTTP or HTTPS, TCP traffic can also be balanced. For example, traffic to a database cluster could be spread across all of the servers.
* **UDP** — More recently, some load balancers have added support for load balancing core internet protocols like DNS and syslogd that use UDP.

These forwarding rules will define the protocol and port on the load balancer itself and map them to the protocol and port the load balancer will use to route the traffic to on the backend.

## **How does the load balancer choose the backend server?**

Load balancers choose which server to forward a request to based on a combination of two factors. They will first ensure that any server they can choose is actually responding appropriately to requests and then use a pre-configured rule to select from among that healthy pool.

### Health Checks

Load balancers should only forward traffic to "healthy" backend servers. To monitor the health of a backend server, health checks regularly attempt to connect to backend servers using the protocol and port defined by the forwarding rules to ensure that servers are listening. If a server fails a health check, and therefore is unable to serve requests, it is automatically removed from the pool, and traffic will not be forwarded to it until it responds to the health checks again.

### Load Balancing Algorithms

The load balancing algorithm that is used determines which of the healthy servers on the backend will be selected. A few of the commonly used algorithms are:

**Round Robin** — Round Robin means servers will be selected sequentially. The load balancer will select the first server on its list for the first request, then move down the list in order, starting over at the top when it reaches the end.

**Least Connections** — Least Connections means the load balancer will select the server with the least connections and is recommended when traffic results in longer sessions.

**Source** — With the Source algorithm, the load balancer will select which server to use based on a hash of the source IP of the request, such as the visitor's IP address. This method ensures that a particular user will consistently connect to the same server.

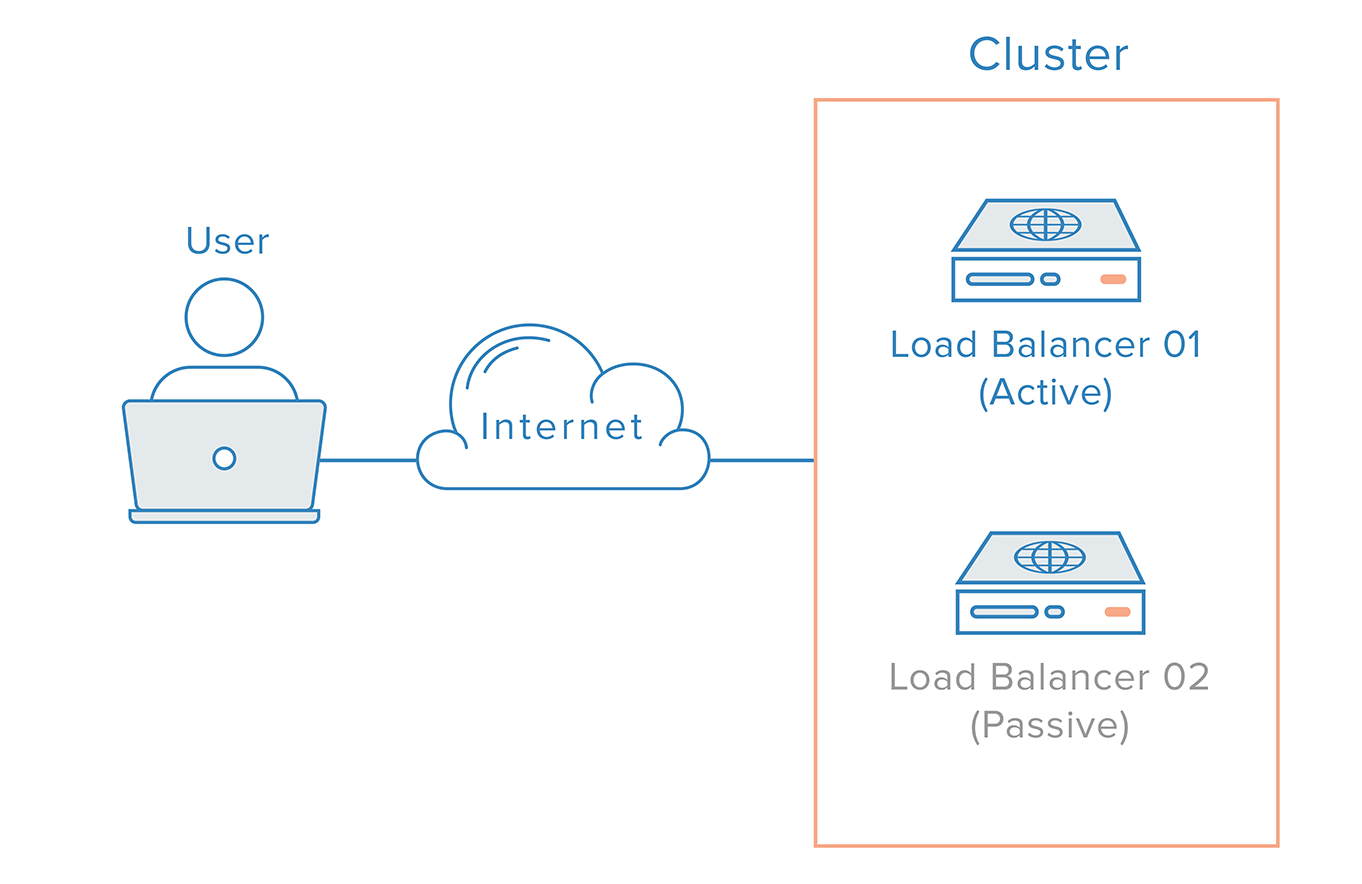
The algorithms available to administrators vary depending on the specific load balancing technology in use.

### How do load balancers handle state?

Some applications require that a user continues to connect to the same backend server. A Source algorithm creates an affinity based on client IP information. Another way to achieve this at the web application level is through **sticky sessions**, where the load balancer sets a cookie and all of the requests from that session are directed to the same physical server.

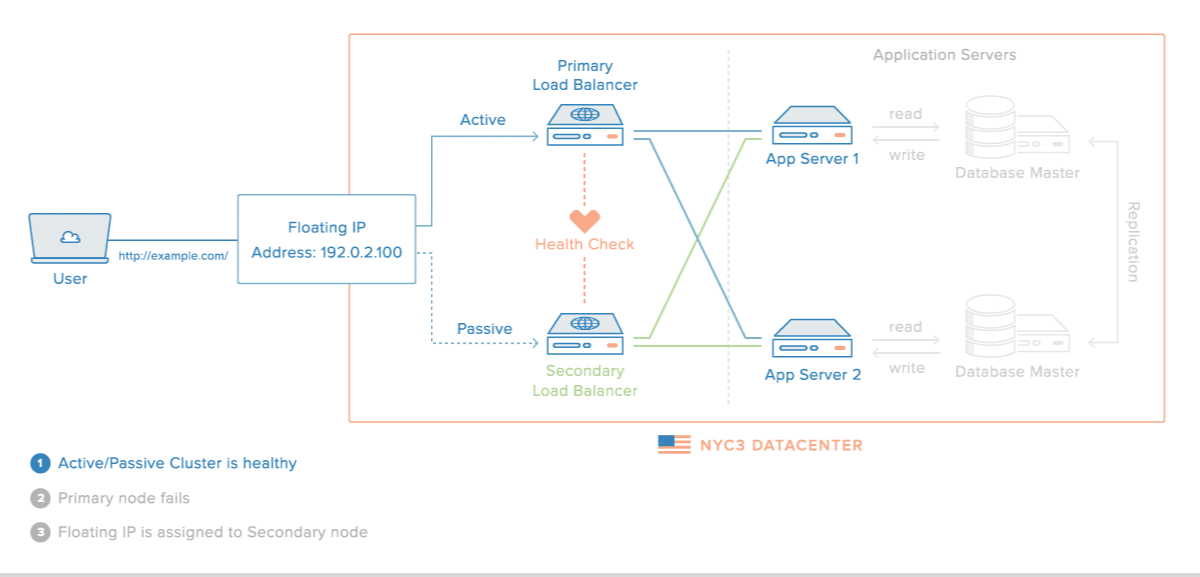
## **Redundant Load Balancers**

To remove the load balancer as a single point of failure, a second load balancer can be connected to the first to form a cluster, where each one monitors the others’ health. Each one is equally capable of failure detection and recovery.



In the event the main load balancer fails, DNS must take users to the to the second load balancer. Because DNS changes can take a considerable amount of time to be propagated on the Internet and to make this failover automatic, many administrators will use systems that allow for flexible IP address remapping, such as floating IPs. On demand IP address remapping eliminates the propagation and caching issues inherent in DNS changes by providing a static IP address that can be easily remapped when needed. The domain name can remain associated with the same IP address, while the IP address itself is moved between servers.

This is how a highly available infrastructure using Floating IPs might look:



<https://f5.com/glossary/load-balancer>

<https://en.wikipedia.org/wiki/Load_balancing_(computing)>

**AIP Management and Gateways**

* Application program interface (**API**) is a set of routines, protocols, and tools for building software applications. An **API** specifies how software components should interact. Additionally, **APIs** are used when programming graphical user interface (GUI) components.
* **API management** is the process of creating and publishing web APIs, enforcing their usage policies, controlling access, nurturing the subscriber community, collecting and analyzing usage statistics, and reporting on performance.
* **Gateway**: a server that act as an API front-end, receives API requests, enforces throttling and security policies, passes requests to the back-end service and then passes the response back to the requester. A gateway often includes a transformation engine to orchestrate and modify the requests and responses on the fly. A gateway can also provide functionality such collecting analytics data and providing caching. The gateway can provide functionality to support authentication, authorization, security, audit and regulatory compliance.

**REST and SOAP API’s**

**Amazon API Gateway:**

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale.

With a few clicks in the AWS Management Console, you can create an API that acts as a “front door” for applications to access data, business logic, or functionality from your back-end services, such as workloads running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda, or any Web application.

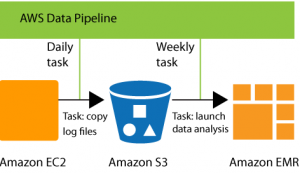
Amazon API Gateway handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, authorization and access control, monitoring, and API version management.

Amazon API Gateway has no minimum fees or startup costs. You pay only for the API calls you receive and the amount of data transferred out.

**Benefits:**

* Low-cost efficient
* Performance at any scale
* Easily monitor API activity
* Streamline API Development
* Flexible Security Controls
* Run your API’s without servers
* Create RESTful endpoints for existing servers

**Data Pipeline in AWS**



AWS Data Pipeline is a web service that helps you reliably process and move data between different AWS compute and storage services, as well as on-premise data sources, at specified intervals.

With AWS Data Pipeline, you can regularly access your data where it’s stored, transform and process it at scale, and efficiently transfer the results to AWS services such as Amazon S3, Amazon RDS, Amazon DynamoDB, and Amazon EMR.

AWS Data Pipeline helps you easily create complex data processing workloads that are fault tolerant, repeatable, and highly available.

You don’t have to worry about ensuring resource availability, managing inter-task dependencies, retrying transient failures or timeouts in individual tasks, or creating a failure notification system.

AWS Data Pipeline also allows you to move and process data that was previously locked up in on-premise data silos.

# Processor State Control for Your EC2 Instance

C-states control the sleep levels that a core can enter when it is idle. C-states are numbered starting with C0 (the shallowest state where the core is totally awake and executing instructions) and go to C6 (the deepest idle state where a core is powered off).

P-states control the desired performance (in CPU frequency) from a core. P-states are numbered starting from P0 (the highest performance setting where the core is allowed to use Intel Turbo Boost Technology to increase frequency if possible), and they go from P1 (the P-state that requests the maximum baseline frequency) to P15 (the lowest possible frequency).

The following instance types provide the ability for an operating system to control processor C-states and P-states:

* c4.8xlarge
* d2.8xlarge
* f1.16xlarge
* g3.16xlarge
* i3.16xlarge
* m4.10xlarge
* m4.16xlarge
* p2.16xlarge
* r4.8xlarge
* r4.16xlarge
* x1.16xlarge
* x1.32xlarge

You might want to change the C-state or P-state settings to increase processor performance consistency, reduce latency, or tune your instance for a specific workload.

The default C-state and P-state settings provide maximum performance, which is optimal for most workloads.

However, if your application would benefit from reduced latency at the cost of higher single- or dual-core frequencies, or from consistent performance at lower frequencies as opposed to bursty Turbo Boost frequencies, consider experimenting with the C-state or P-state settings that are available to these instances.

# Placement Groups

A placement group is a logical grouping of instances within a single Availability Zone. Placement groups are recommended for applications that benefit from low network latency, high network throughput, or both. To provide the lowest latency, and the highest packet-per-second network performance for your placement group, choose an instance type that supports enhanced networking. For more information, see Enhanced Networking.

First, you create a placement group and then you launch multiple instances into the placement group. We recommend that you launch the number of instances that you need in the placement group in a single launch request and that you use the same instance type for all instances in the placement group. If you try to add more instances to the placement group later, or if you try to launch more than one instance type in the placement group, you increase your chances of getting an insufficient capacity error.

There is no charge for creating a placement group.

If you stop an instance in a placement group and then start it again, it still runs in the placement group. However, the start fails if there isn't enough capacity for the instance.

If you receive a capacity error when launching an instance in a placement group that already has running instances, stop and start all of the instances in the placement group, and try the launch again. Restarting the instances may migrate them to hardware that has capacity for all the requested instances.

## **Placement Group Limitations**

Placement groups have the following limitations:

* A placement group can't span multiple Availability Zones.
* The name you specify for a placement group must be unique within your AWS account.
* The following are the only instance types that you can use when you launch an instance into a placement group:
  + General purpose: m4.large | m4.xlarge | m4.2xlarge | m4.4xlarge | m4.10xlarge | m4.16xlarge
  + Compute optimized: c4.large | c4.xlarge | c4.2xlarge | c4.4xlarge | c4.8xlarge | c3.large | c3.xlarge | c3.2xlarge | c3.4xlarge | c3.8xlarge | cc2.8xlarge
  + Memory optimized: cr1.8xlarge | r3.large | r3.xlarge | r3.2xlarge | r3.4xlarge | r3.8xlarge | r4.large | r4.xlarge | r4.2xlarge | r4.4xlarge | r4.8xlarge | r4.16xlarge | x1.16xlarge | x1.32xlarge
  + Storage optimized: d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge | hi1.4xlarge | hs1.8xlarge | i2.xlarge | i2.2xlarge | i2.4xlarge | i2.8xlarge | i3.large | i3.xlarge | i3.2xlarge | i3.4xlarge | i3.8xlarge | i3.16xlarge
  + Accelerated computing: cg1.4xlarge | f1.2xlarge | f1.16xlarge | g2.2xlarge | g2.8xlarge | g3.4xlarge | g3.8xlarge | g3.16xlarge | p2.xlarge | p2.8xlarge | p2.16xlarge
* The maximum network throughput speed of traffic between two instances in a placement group is limited by the slower of the two instances. For applications with high-throughput requirements, choose an instance type with 10 Gbps or 20 Gbps network connectivity.
* Although launching multiple instance types into a placement group is possible, this reduces the likelihood that the required capacity will be available for your launch to succeed. We recommend using the same instance type for all instances in a placement group.
* You can't merge placement groups. Instead, you must terminate the instances in one placement group, and then relaunch those instances into the other placement group.
* A placement group can span peered VPCs; however, you will not get full-bisection bandwidth between instances in peered VPCs.
* You can't move an existing instance into a placement group. You can create an AMI from your existing instance, and then launch a new instance from the AMI into a placement group.
* Reserved Instances provide a capacity reservation for EC2 instances in an Availability Zone. The capacity reservation can be used by instances in a placement group that are assigned to the same Availability Zone. However, it is not possible to explicitly reserve capacity for a placement group.
* To ensure that network traffic remains within the placement group, members of the placement group must address each other via their private IPv4 addresses or IPv6 addresses (if applicable). If members address each other using their public IPv4 addresses, throughput drops to 5 Gbps or less.
* Network traffic to and from resources outside the placement group is limited to 5 Gbps.

## **Launching Instances into a Placement Group**

We suggest that you create an AMI specifically for the instances that you'll launch into a placement group.

**To launch instances into a placement group using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Create an AMI for your instances.
   1. From the Amazon EC2 dashboard, choose **Launch Instance**. After you complete the wizard, choose **Launch**.
   2. Connect to your instance. (For more information, see Connect to Your Linux Instance.)
   3. Install software and applications on the instance, copy data, or attach additional Amazon EBS volumes.
   4. (Optional) If your instance type supports enhanced networking, ensure that this feature is enabled by following the procedures in Enhanced Networking on Linux.
   5. In the navigation pane, choose **Instances**, select your instance, choose **Actions**, **Image**, **Create Image**. Provide the information requested by the **Create Image** dialog box, and then choose **Create Image**.
   6. (Optional) You can terminate this instance if you have no further use for it.
3. Create a placement group.
   1. In the navigation pane, choose **Placement Groups**.
   2. Choose **Create Placement Group**.
   3. In the **Create Placement Group** dialog box, provide a name for the placement group that is unique in the AWS account you're using, and then choose **Create**.

When the status of the placement group is available, you can launch instances into the placement group.

1. Launch instances into your placement group.
   1. In the navigation pane, choose **Instances**.
   2. Choose **Launch Instance**. Complete the wizard as directed, taking care to do the following:

* On the **Choose an Amazon Machine Image (AMI)** page, select the **My AMIs** tab, and then select the AMI that you created.
* On the **Choose an Instance Type** page, select an instance type that can be launched into a placement group.
* On the **Configure Instance Details** page, enter the total number of instances that you'll need in this placement group, as you might not be able to add instances to the placement group later on.
* On the **Configure Instance Details** page, select the placement group that you created from **Placement group**. If you do not see the **Placement group** list on this page, verify that you have selected an instance type that can be launched into a placement group, as this option is not available otherwise.

**To launch instances into a placement group using the command line**

1. Create an AMI for your instances using one of the following commands:
   * create-image (AWS CLI)
   * New-EC2Image (AWS Tools for Windows PowerShell)
2. Create a placement group using one of the following commands:
   * create-placement-group (AWS CLI)
   * New-EC2PlacementGroup (AWS Tools for Windows PowerShell)
3. Launch instances into your placement group using one of the following options:
   * --placement with run-instances (AWS CLI)
   * -PlacementGroup with New-EC2Instance (AWS Tools for Windows PowerShell)

## **Deleting a Placement Group**

You can delete a placement group if you need to replace it or no longer need a placement group. Before you can delete your placement group, you must terminate all instances that you launched into the placement group.

**To delete a placement group using the console**

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select and terminate all instances in the placement group. (You can verify that the instance is in a placement group before you terminate it by checking the value of **Placement Group** in the details pane.)
4. In the navigation pane, choose **Placement Groups**.
5. Select the placement group, and then choose **Delete Placement Group**.
6. When prompted for confirmation, choose **Yes, Delete**.

**To delete a placement group using the command line**

You can use one of the following sets of commands.

* terminate-instances and delete-placement-group (AWS CLI)
* Stop-EC2Instance and Remove-EC2PlacementGroup(AWS Tools for Windows PowerShell)

**Amazon EMR**

Amazon EMR provides a managed Hadoop framework that makes it easy, fast, and cost-effective to process vast amounts of data across dynamically scalable Amazon EC2 instances. You can also run other popular distributed frameworks such as Apache Spark, HBase, Presto, and Flink in Amazon EMR, and interact with data in other AWS data stores such as Amazon S3 and Amazon DynamoDB.

Amazon EMR securely and reliably handles a broad set of big data use cases, including log analysis, web indexing, data transformations (ETL), machine learning, financial analysis, scientific simulation, and bioinformatics.

**AWS Snowball**

AWS Snowball is a service that accelerates transferring large amounts of data into and out of AWS using physical storage appliances, bypassing the Internet. Each AWS Snowball appliance type can transport data at faster-than internet speeds. This transport is done by shipping the data in the appliances through a regional carrier. The appliances are rugged shipping containers, complete with E Ink shipping labels.

With a Snowball, you can transfer hundreds of terabytes or petabytes of data between your on-premises data centers and Amazon Simple Storage Service (Amazon S3). AWS Snowball uses Snowball appliances and provides powerful interfaces that you can use to create jobs, transfer data, and track the status of your jobs through to completion. By shipping your data in Snowballs, you can transfer large amounts of data at a significantly faster rate than if you were transferring that data over the Internet, saving you time and money.

There are many options for transferring your data into AWS. Snowball is intended for transferring large amounts of data. If you want to transfer less than 10 terabytes of data between your on-premises data centers and Amazon S3, Snowball might not be your most economical choice.

Snowball uses Snowball appliances shipped through your region's carrier. Each Snowball is protected by AWS Key Management Service (AWS KMS) and made physically rugged to secure and protect your data while the Snowball is in transit. In the US regions, Snowballs come in two sizes: 50 TB and 80 TB. All other regions have 80 TB Snowballs only.

## **Snowball Features**

Snowball with the Snowball appliance has the following features:

* You can import and export data between your on-premises data storage locations and Amazon S3.
* Snowball has an 80 TB model available in all regions, and a 50 TB model only available in the US regions.
* Encryption is enforced, protecting your data at rest and in physical transit.
* You don't have to buy or maintain your own hardware devices.
* You can manage your jobs through the AWS Snowball Management Console, or programmatically with the job management API.
* You can perform local data transfers between your on-premises data center and a Snowball can be done through the Snowball client, a standalone downloadable client, or programmatically using Amazon S3 REST API calls with the downloadable Amazon S3 Adapter for Snowball. For more information, see Transferring Data with a Snowball.
* The Snowball is its own shipping container, and its E Ink display changes to show your shipping label when the Snowball is ready to ship. For more information, see Shipping Considerations for AWS Snowball.
* For a list of regions where the Snowball appliance is available,

**AWS Cloud Trail**

AWS CloudTrail is an AWS service that helps you enable governance, compliance, and operational and risk auditing of your AWS account. Actions taken by a user, role, or an AWS service are recorded as events in CloudTrail. Events include actions taken in the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs.

Visibility into your AWS account activity is a key aspect of security and operational best practices. You can use CloudTrail to view, search, download, archive, analyze, and respond to account activity across your AWS infrastructure. You can identify who or what took which action, what resources were acted upon, when the event occurred, and other details to help you analyze and respond to activity in your AWS account.

You can integrate CloudTrail into applications using the API, automate trail creation for your organization, check the status of trails you create, and control how users view CloudTrail events.

CloudTrail is enabled on your AWS account when you create it. When activity occurs in your AWS account, that activity is recorded in a CloudTrail event. You can easily view events in the CloudTrail console by going to **Event history**.

Event history allows you to view, search, and download the past seven days of supported activity in your AWS account. In addition, you can create a CloudTrail trail to further archive, analyze, and respond to changes in your AWS resources. A trail is a configuration that enables delivery of events to an Amazon S3 bucket that you specify. You can also deliver and analyze events in a trail with Amazon CloudWatch Logs and Amazon CloudWatch Events. You can create a trail with the CloudTrail console, the AWS CLI, or the CloudTrail API.

You can create two types of trails:

**A trail that applies to all regions**

When you create a trail that applies to all regions, CloudTrail creates the same trail in each region. It then records events in each region and delivers the CloudTrail event log files to an S3 bucket that you specify. This is the default option when you create a trail in the CloudTrail console.

**A trail that applies to one region**

When you create a trail that applies to one region, CloudTrail records the log files in that region only. It then delivers the CloudTrail event log files log to an S3 bucket that you specify. If you create additional single trails, you can have those trails deliver CloudTrail event log files to the same Amazon S3 bucket or to separate buckets.

For both types of trails, you can specify an Amazon S3 bucket from any region.

By default, CloudTrail event log files are encrypted using Amazon S3 server-side encryption (SSE). You can also choose to encrypt your log files with an AWS Key Management Service (AWS KMS) key. You can store your log files in your bucket for as long as you want. You can also define Amazon S3 lifecycle rules to archive or delete log files automatically. If you want notifications about log file delivery and validation, you can set up Amazon SNS notifications.

CloudTrail typically delivers log files within 15 minutes of account activity. In addition, CloudTrail publishes log files multiple times an hour, about every five minutes. These log files contain API calls from services in the account that support CloudTrail.

**What Is Auto Scaling?**

Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called *Auto Scaling groups*. You can specify the minimum number of instances in each Auto Scaling group, and Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in each Auto Scaling group, and Auto Scaling ensures that your group never goes above this size. If you specify the desired capacity, either when you create the group or at any time thereafter, Auto Scaling ensures that your group has these many instances. If you specify scaling policies, then Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

For example, the following Auto Scaling group has a minimum size of 1 instance, a desired capacity of 2 instances, and a maximum size of 4 instances. The scaling policies that you define adjust the number of instances, within your minimum and maximum number of instances, based on the criteria that you specify.


   An illustration of a basic Auto Scaling group.
  

## Auto Scaling Components

The following table describes the key components of Auto Scaling.

|  |  |
| --- | --- |
| A graphic representing an Auto Scaling group. | **Groups**  Your EC2 instances are organized into groups so that they can be treated as a logical unit for the purposes of scaling and management. When you create a group, you can specify its minimum, maximum, and, desired number of EC2 instances. |
| A graphic representing a launch configuration. | **Launch configurations**  Your group uses a launch configuration as a template for its EC2 instances. When you create a launch configuration, you can specify information such as the AMI ID, instance type, key pair, security groups, and block device mapping for your instances. |
| A graphic representing a launch configuration. | **Scaling plans**  A scaling plan tells Auto Scaling when and how to scale. For example, you can base a scaling plan on the occurrence of specified conditions (dynamic scaling) or on a schedule. |

**Minimum, Maximum and Desired values:**

The ASG will always try to maintain the Desired Capacity. If you scale up or down, and the new number of instances is different than the Desired Capacity, the ASG will add or remove an instance to go back to the desired capacity.

If you use scaling policies, and the policy condition is met, the ASG will change the Desired Capacity to match the result of your scaling policy. E.g., you have a Desired Capacity of 2, and a policy that says to scale up if the CPU utilization goes over a threshold.

If that policy is fulfilled, then the Desired Capacity will increase to 3, and so on.

So manually scaling up and down will result in your ASG restoring the number of instances to the Desired Capacity.

If you want to manually scale up and down, you could set your Max and Min to a wide value, and move Desired Capacity within it.

So, you could do Max=10, Min=1, Desired=3. Then you could scale up or down just by changing the Desired Capacity. (This is how we use Auto Scaling, and I think why I gave you a bad answer before.)

If you want to terminate an instance and change the Desired Capacity at the same time, the CLI can do that.

**Route 53**

The **name** is a reference to TCP or UDP port **53**, where DNS server requests are addressed.

# Choosing a Routing Policy

When you create a resource record set, you choose a routing policy, which determines how Amazon Route 53 responds to queries:

* **Simple routing policy** – Use for a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website.
* **Failover routing policy** – Use when you want to configure active-passive failover.
* **Geolocation routing policy** – Use when you want to route traffic based on the location of your users.
* **Latency routing policy** – Use when you have resources in multiple locations and you want to route traffic to the resource that provides the best latency.
* **Multivalue answer routing policy** – Use when you want Amazon Route 53 to respond to DNS queries with up to eight healthy records selected at random.
* **Weighted routing policy** – Use to route traffic to multiple resources in proportions that you specify.

**Reserved IP Addresses in AWS**

* 10.0.0.0: Network address.
* 10.0.0.1: Reserved by AWS for the VPC router.
* 10.0.0.2: Reserved by AWS. The IP address of the DNS server is always the base of the VPC network range plus two; however, we also reserve the base of each subnet range plus two. For more information, see Amazon DNS Server.
* 10.0.0.3: Reserved by AWS for future use.
* 10.0.0.255: Network broadcast address. We do not support broadcast in a VPC, therefore we reserve this address.

**Elastic Search:**

Is a free open source engine used to search logs in their general, Amazon made it easier for us as service to create cloud, we connect to it cloud watch and can monitor,

After creating ES, we can add several ways to add data or connect to logs, we can use it by API,

We use Elastic Bean stalk to quickly deploy some applications on AWS, and the Elastic Bean stalk will create the own infrastructure itself. It will setup the load balancing and auto scaling, and configures monitoring.

First, we should create infrastructure, and then we have to create a pipeline to deploy our code to that infrastructure. And new code can be deployed to a testing or QA environment before going to production. For these developers must learn how to deploy resources and how-to setup those pipeline, instead of creating the features, The EBS aims to solve that problem using EBS we can deploy our code and service will automatically provision our capacity such as Auto Scaling, Load Balancing, and also monitors the environment. And we can customize this auto generated environment according to the requirement.

If we have application with shorter life cycles which don’t require more flexibility, if we are working with longer lifecycle applications which requires more configuration, customization it is best to use cloud formation template.

Languages:

* Java, .Net, PHP, Node.js, Python, Ruby,

Web Servers:

* Apache, Tomcat, Docker, Nginx, Java SE, IIS,

Supported Deployment Platforms

* Git, IDEs (Eclipse, Visual Studio), Manual upload (eg WAR files, ZIP)

**We use Elastic Beanstalk**

For spending minimal time to set up infrastructure, and testing, maintain some flexibility and control over the resources used to power applications.

**How EBS Work**

**Deployment Methods in EBS:**

**All at once**: It updates all instances at the same time for the reason it causes downtime, and this method is the fastest method among all. While deploying an application, instead of taking batches and updating, this method will update all instances all at same time.

**Rolling:** Rolling Deployment prevent the downtime, because of not updating all at once, and will update the application using batch method. Batch is a group of instances which are fixed in number, during a deployment each batch will take out of service, rest of them in service in serving traffic. While deployment, ELB will detach the batch and will serve the traffic with other batches, and then EBS will create a new batch for the updated version and the will reattach to the ELB. And ELB will check the traffic and instance health, and once the health check reaches the min threshold, then it will start serve the traffic.

**Rolling with additional batch:** It Is same as the rolling deployment method, instead of detaching the batching directly, we will add a new batch before detaching and then serve the traffic to the batch, and the detached batch will update the version and will reattach the batch.

**Immutable:** EBS will create a second Auto scaling group with a new application version and puts I our environment, then EBS launch a single instance with the updated application/configuration. That Instance start serving the traffic, once instance passes health checks EBS starts to launch more instances with the new application configuration until the no of instances in the new auto scaling group matches the no in the original.

**Blue**/**Green:** It follows the Immutable deployment method, we will create new resources like Immutable. We need to clone our current environment, or launch a new environment with different configurations, then we deploy the new application version to that new environment, once we test the new version in the new environment and we verify that is good to go we swap environment URLs which is basically modifies route 53 DNS configurations to point your green deployment to become original blue deployment.

# VPC Peering

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account. In both cases, the VPCs must be in the same region.

AWS uses the existing infrastructure of a VPC to create a VPC peering connection; it is neither a gateway nor a VPN connection, and does not rely on a separate piece of physical hardware. There is no single point of failure for communication or a bandwidth bottleneck.

**Connection Draining:**

To ensure that a Classic Load Balancer stops sending requests to instances that are de-registering or unhealthy, while keeping the existing connections open, use *connection draining*. This enables the load balancer to complete in-flight requests made to instances that are de-registering or unhealthy.

When you enable connection draining, you can specify a maximum time for the load balancer to keep connections alive before reporting the instance as de-registered. The maximum timeout value can be set between 1 and 3,600 seconds (the default is 300 seconds). When the maximum time limit is reached, the load balancer forcibly closes connections to the de-registering instance.

While in-flight requests are being served, the load balancer reports the state of a de-registering instance as InService: Instance deregistration currently in progress. When the de-registering instance is finished serving all in-flight requests, or when the maximum timeout limit is reached, the load balancer reports the instance state as OutOfService: Instance is not currently registered with the LoadBalancer.

If an instance becomes unhealthy, the load balancer reports the instance state as OutOfService. If there are in-flight requests made to the unhealthy instance, they are completed. The maximum timeout limit does not apply to connections to unhealthy instances.

If your instances are part of an Auto Scaling group and connection draining is enabled for your load balancer, Auto Scaling waits for the in-flight requests to complete, or for the maximum timeout to expire, before terminating instances due to a scaling event or health check replacement.

You can disable connection draining if you want your load balancer to immediately close connections to the instances that are de-registering or have become unhealthy. When connection draining is disabled, any in-flight requests made to instances that are de-registering or unhealthy are not completed.

**Terms in Load Balancing**

**LTM:** Local Traffic Manager; **GTM:** Global Traffic Manager

**Difference between GTM and LTM**

The **Global Traffic Manager**, aka **GTM**, and now referred to **DNS**, is one of the cutting-edge modules offered on F5 Networks BIG-IP® platform. “Global” is the right word for this module because it can make name resolution load balancing decisions for systems located anywhere in the World, not just the US. You can think of the GTM as an intelligent DNS that is security minded. In other words, its’ logic can make informed decisions on correlating a hostname to an IP address while keeping security in check.

Most things you do on the Internet or Private networks start with name resolution – so it makes sense if you’re going to load balance an application it would start at this layer – resolving names to IPs based on availability, Performance, and even Persistence. It’s important to note, traffic does not “route” through the GTM, the GTM simply tells you the best IP to route to based on metrics for the URL in question. That IP can be almost anything really, but usually it’s an actual server, or a virtual IP that fronts multiple servers. Like a traditional DNS architecture, you usually have multiple GTMs in your architecture, this is for redundancy/availability.

The main configuration element in a GTM is called a Wide IP or WIP for short. There are many configuration elements that work in concert with a WIP, but at the base of it all is the Wide IP. A WIP equates to the common URL you’re load balancing, for example www.yourcompany.com. A pool or pools are usually attached to a WIP which contain the IPs it’s intelligently resolving. Like your run of the mill DNS server, the GTM does not tell the requester any information about ports. Though, the monitors associated with the pool members can indeed monitor availability or performance on ports.

## **LTM – Local Traffic Manager Overview**

The **Local Traffic Manager**, aka **LTM**, is the most popular module offered on F5 Networks BiG-IP® platform. The real power of the LTM is it’s a Full Proxy, allowing you to augment client and server-side connections. All while making informed load balancing decisions on availability, performance, and persistence. “Local” in the name is important, opposed to the GTM, traffic flows through the LTM to the servers it balances traffic to. Usually the servers it’s load balancing sit “locally” in the same data center as the LTM, though that is not a requirement. With SNAT (secure network address translation) configured on the VIP, if you can route to it you can load balance it – so it’s possible to have servers in different data centers be a part of the same pool in an LTM VIP.

The main configuration element on an LTM is the Virtual IP or VIP for short. There is a plethora of configuration elements that work with VIPs, but at the heart of the technology it’s a VIP they are all a part of. Like a WIP, VIPs equate to the URL you’re load balancing, but at its lowest level. Like a WIP it usually contains a pool with the servers it’s load balancing & monitor(s) to measure availability / performance.

## **Some of the Key differences of the GTM vs LTM**

* The biggest difference between the GTM and LTM, as mentioned earlier, is traffic doesn’t flow through the GTM to your servers.
* The GTM is an intelligent name resolver, intelligently resolving names to IP addresses.
* Once the GTM provides you with an IP to route to you’re done with the GTM until you ask it to resolve another name for you.
* Like a usual DNS server, the GTM does not provide any port information in its resolution.
* The LTM doesn’t do any name resolution and assumes a DNS decision has already been made.
* When traffic is directed to the LTM traffic flows directly through its’ full proxy architecture to the servers it’s load balancing.
* Since the LTM is a full proxy it’s easy for it to listen on one port but direct traffic to multiple hosts listening on any port specified.

## **How do the GTM & LTM work together?**

The GTM and LTM can work together or they can be totally independent. If your organization owns both modules it’s usually using them together, and that’s where the real power comes in. They do this via a proprietary protocol called iQuery. iQuery, functioning on TCP port 4353, reports VIP availability / performance back to the GTMs. The GTMs can then dynamically resolve VIPs that live on an LTM(s).

When a GTM has LTMs as servers in its’ configuration there is no need to monitor the actual VIP(s) with application monitors, as the LTM is doing that & iQuery reports the information back to the GTM.

As you can see the GTM & LTM modules are powerful tools in the world of Application Delivery / Load Balancing. Together the GTM & LTM make one mean lean Application Delivery machine.

**Difference between S3 and EBS**

Think of EBS as a removable hard disk accessible (AFAIK) only to EC2 instances. You typically would attach (mount) it to an EC2 instance to provide persistent storage to the OS that's running in that instance. Access to the EBS is cheaper (per I/O request) and faster than it would be to an S3 bucket. Also, you can snapshot an EBS volume and share copies of it (likewise, there are publicly available EBS volumes with various data stored in them). However, EBS costs more/month per megabyte of storage.

EBS is a fixed size (set when you create it). You can attach multiple EBS devices to a single EC2 instance and even set them up as a RAID for high speed storage.  
  
S3 is more like a Content Delivery Network, accessible via HTTP or other means. It allows for the storage of vast amounts of data. Typical use case for S3 is to serve up static site content (large images, Flash files, videos, etc.) to offload the work and storage from your web server (object links in a web page would point directly to the object stored in your S3 bucket, rather than to your web server).

S3's per/transaction costs are higher and I/O rate is slower than EBS, but cost per megabyte stored is much lower. S3 size is dynamic, your files can be up to 5TB in size (each) and you only get charged for space you're using.