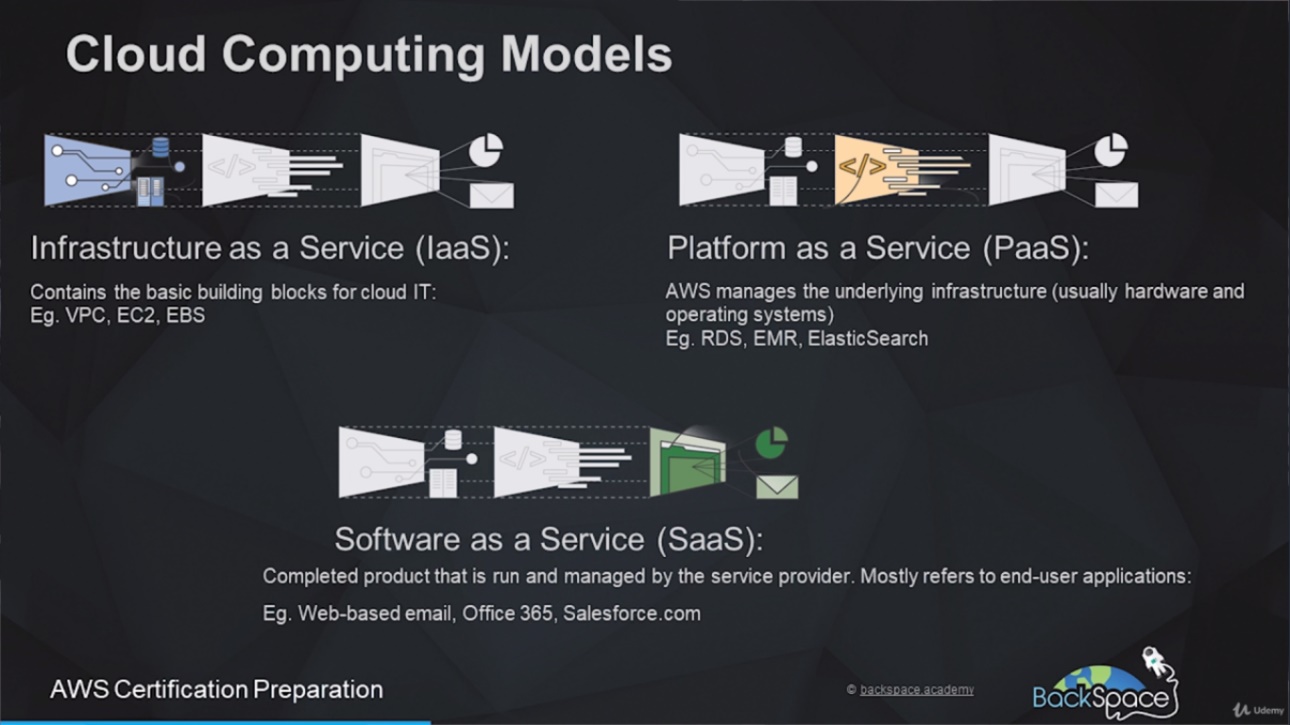
**Amazon Web Services**

**Introduction to AWS Cloud Computing Models**

**Infrastructure as a Service (IaaS)**: It is basic building block of cloud IT, if we want to launch a Linux server and want to manage that server ourselves that is how we do that as infrastructure as a service and we would do that using elastic compute cloud or EC2 Service

**Platform as a Service (PaaS)**: this is where AWS take a little more control over the underline infrastructure, so aws manages that underline infrastructure and the hardware & software. Example RDS, in that service, aws provision all the operating system, the server and everything to run that, but we still need to do the high-level administration of that database.

**Software as a Service (SaaS)**: it is a complete product that normally runs on the browser and it is mostly refers to end-user application. Example: Office 365, SalesForce.com

**Server-less Computing**: Allows you to build and run applications without thinking about servers, we don’t need to provision the server yourself, aws will do that job for us. Its also referred to as **Function-as-a-Service (FaaS)** or Abstract services. Example, S3 bucket, AWS Lambda, Amazong DynamoDB, Amazon SNS.

**Introduction to Storages Services**:

**Simple Storage Service (S3)**: its designed to store and access any type of data over the internet, it’s a serviceless service and as such we don’t need to worry about whats behind it, there must be a file server, an OS, a hard drive, but we don’t need to worry about that at all, we just need to create this bucket and just upload the data on the bucket, the bucket grows as we add object to it, and the size of the bucket is practically unlimited.

**Amazon Glacier**: Amazon Glacier is the cheapest storage option on AWS and it's used for long-term archiving of data it's a server-less service just like S3 bucket, but it is not as readily accessible as S3 so it should only be used for content that is to be archived. We can also set up a lifecycle rule that will automatically migrate old data from Amazon S3 automatically over to Glacier for long-term archiving.

**Amazon Elastic Block Store (EBS)**: It a highly available low latency block storage and it's specifically for attaching to servers that are launched with the Amazon EC2 service and it's similar to attaching a hard drive to your computer at home works in the same manner it's block device storage.

**Elastic File System (EFS)**: It is a network attached storage and it's specifically for Amazon EC2 servers because it is network attached storage this allows multiple servers to access a one data source in a similar way to a NAS on your network at home can be accessed by multiple computers on that network.

**AWS Storage Gateway**: The AWS Storage Gateway enables hybrid storage between on-premise environments and the AWS cloud it provides a low latency performance by caching frequently used data on-premises while storing the less frequently data in Amazon Cloud storage services.

**Snowball Device**: a snowball device is a portable petabyte scale data storage device that can be used to migrate data and large amounts of data from on-premise environments over to the AWS cloud you simply download your data to the snowboard device then you send it off to AWS who will then upload that data to an AWS storage service for you.

Example of Storage Service: 3. 01-010-02 Intro to Storage Services: Time- 7:00

Hybrid Storage Example: 3. 01-010-02 Intro to Storage Services: Time- 11:10

**Introduction to AWS DataBase Services**

**Relational DataBase Service (RDS)**: It is a fully managed database service that makes it easy to launch database servers in the AWS cloud and scale them when required the RDS service can launch service for mySQL including variations of the mySQL database engine with MariaDB and Amazon's own enterprise version of mySQL Amazon Aurora, standard postgre SQL is also available and also available as Amazon's Enterprise Aurora postgre SQL, Microsoft SQL server and oracle are also available.

**Amazon DynamoDB**: It is AWS as no SQL database as a service it's a service-less service like Amazon s3 and as such you don't need to worry about the underlying infrastructure behind it AWS takes care of everything for you and it provides high speed extremely low latency performance.

**Amazon RedShift**: Amazon redshift is a fast fully managed petabyte scale data warehouse that is based upon the postgre SQL database engine if you're looking for a big data storage solution redshift is perfect for this.

**Amazon ElastiCache**: Amazon ElastiCache is an in-memory data store or cache in the cloud it allows you to retrieve information from fast fully managed in-memory caches instead of relying for slower disk based databases.

**Amazon Migration Services (DMS)**: The AWS database migration service or orchestrates a migration of databases over to AWS easily and securely it can also migrate data from one database engine type to another totally different database engine type for example you can use it to migrate from Oracle over to Amazon Aurora.

**Amazon Neptune**: Amazon Neptune is a fast reliable fully managed graph database service it has a purpose-built high-performance graph database engine optimized for storing billions of relationships and clearing the graph with millisecond latency.

DataBase Example: 3.010-03 Intro to Database Services: Time- 3:00

**Introduction to AWS Compute Services**

**Elastic Compute Cloud 2 (EC2)**: It provides virtual servers in the AWS cloud you can launch one or thousands of instances simultaneously and only pay for what you use there's a broad range of instance types with varying compute and memory capabilities and those will be optimized for different use cases.

**EC2 Autoscaling**: It allows you to dynamically scale your Amazon ec2 capacity up or down automatically according to conditions that you define it can scale up or down by launching or terminating instances based on demand it can also perform health checks on those instances and replace them when they become unhealthy.

**Amazon Lightsail**: Amazon Lightsail, it's the easiest way to launch virtual servers running applications in the AWS cloud AWS will provision everything you need including DNS management and storage to get you up and running as quickly as possible.

**Elastic container service (ECS)**: It is a highly scalable high-performance container management service for docker containers the containers they will run on a managed cluster of EC2 instances.

**AWS Lambda**: It is service-less service and lets you run code in the AWS cloud without having to worry about provisioning or managing that service you just upload your code and AWS takes care of everything for you.

WebServer Example: 4. 010-04 Intro to Compute Services: Time- 2:20

**Introduction to AWS Networking & Content Delivery (CDN)**

**Amazon CloudFront**: Amazon CloudFront is a global content delivery network or CDN for short that securely delivers your frequently requested content to over 100 edge locations across the globe and by doing this it achieves low latency and high transfer speeds for your end-users it also provides protection against DDoS attacks.

**Virtual Private Cloud (VPC)**: VPC lets you provision a logically isolated section of the AWS cloud and you can launch AWS resources in that virtual network that you yourself define and this is your own personal private space within the AWS cloud and no one can enter it unless you allow them to enter it.

**Direct Connect**: AWS Direct Connect is a high-speed dedicated network connection to AWS. Enterprises can use it to establish a private connection to the AWS cloud in situations where a standard internet connection won't be adequate.

**Elastic Load Balancing (ELB):** Automatically distributes incoming traffic for your application across multiple EC2 instances and in multiple availability zones so if one availability zone goes down the traffic will still go to the other availability zone and your application will continue to deliver responses to requests. It also allows you to achieve high availability and fault tolerance by distributing traffic evenly amongst those instances and it can also bypass unhealthy instances.

**Route 53**: Amazon route 53 is a highly available and scalable domain name system or DNS for short and it can handle direct traffic for your domain name and direct that traffic to your back-end web server.

**API Gateway**: Amazon API gateway is a fully managed service that makes it easy for developers to create and deploy secure application programming interfaces (APIs) at any scale. It handles all of the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls. It's a service-less service and as such you don't need to worry about the underlying infrastructure AWS looks after everything for you.

Networking Example: 4. 010-04 Intro to Compute Services: Time- 9:47

**Introduction to AWS Management Tools**

**CloudFormation**: cloud formation allows you to use a text file to define your infrastructure and to use his text file to deploy resources on the AWS cloud this allows for the defining of your infrastructure as code and you can manage your infrastructure with the same version control tools that you use to manage your code.

**AWS Service Catalog**: Service Catalog allows enterprises to catalogue resources that can be deployed on the AWS cloud, this allows an enterprise to achieve common governance and compliance for its IT resources by clearly defining what can be deployed on the AWS cloud.

**AWS CloudWatch**: AWS cloud watch is a monitoring service for AWS cloud resources and applications that are deployed on the AWS cloud, it can be used for triggering scaling operations or it can also be used for providing insight into your deployed resources.

**AWS System Manager**: AWS Systems Manager provides a unified user interface that allows you to view operational data from multiple AWS services and to automate tasks across your AWS resources that helps to short the time to detect and resolve operational problems.

**AWS CloudTrail**: AWS cloudtrail monitors and logs AWS account activity including actions taken through the AWS management console, the AWS software development kits, the command-line tools and other AWS services. So this greatly simplifies security analysis of the activity of users of your account.

**AWS Config**: AWS config enables you to assess audit and evaluate the configurations of your AWS resources this simplifies compliance auditing, security analysis, change management and control and also operational troubleshooting.

**AWS OpsWorks**: AWS opsworks provides managed instances of chef and puppet. chef and puppet can be used to configure and automate the deployment of AWS resources.

**AWS Trusted Advisor**: AWS trusted advisor is an online expert system that can analyze your AWS account and the resources inside it and then advise you on how to achieve high security and best performance from those resources.

**Introduction to AWS Application Integration**

**Step Functions**: It makes it easy to coordinate the components of distributed applications and micro services using a visual workflow for example

you may want a second function to always follow the first and only run if and when the first succeeds and you may want to execute two functions in parallel for example you define your application visually as a series of steps you define the process flow of those steps and then you can deploy your application automatically.

**Simple WorkFlow Service (SWF)**: Amazon simple workflow service works in a similar way to step functions in coordinating multiple components of a business process for new applications it's recommended to use step functions not the SWF service Amazon.

**Simple Notification Service (SNS)**: SNS is a flexible fully managed pub/sub messaging service, it means that you can create a topic and users subscribe to that topic and when you publish a message to the topic the users that have subscribed to that topic will receive that message it can also be used for push notifications for mobile devices.

**Amazon Simple Queue Service (SQS)**: SQS is a fully managed Message Queuing service and that makes it easy to decouple your applications from demand what that means is that it allows messages to build up in a queue until the processing server that processes those messages can catch up with the demand.

Process Decoupling Example: 6. 010-06 Intro to Application Services: Time- 2:15

**Introduction to AWS Customer Engagement Services**

**Amazon Connect**: A self-service contact center in the AWS cloud and that is delivered on a pay-as-you-go pricing model it has a drag-and-drop graphical user interface and that allows you to create process flows that define customer interactions without having any coding at all.

**Amazon Pinpoint**: Amazon Pinpoint allows you to send email SMS and mobile push messages for targeted marketing campaigns as well as direct messages to your individual customers for example an order confirmation.

**Simple Email Services (SES)**: A cloud-based bulk email sending service.

**Introduction to AWS Analytics**

**Amazon Elastic MapReduce (EMR)**: It is AWS's Hadoop framework as a service, you can also run other frameworks in Amazon EMR that integrate with Hadoop such as Apache spark, HBase, Presto and Flink. Data can be analyzed by EMR in a number of AWS data stores including Amazon s3 and Amazon DynamoDB.

**Amazon Athena**: Amazon Athena allows you to analyze data stored in an Amazon s3 bucket using standard SQL statement.

**Amazon Elastic Search**: Amazon Elastic search is a fully managed service for elastic.co's ElasticSearch framework this allows high-speed querying and analysis of data that is stored on AWS.

**Kinesis**: it allows you to collect, process and analyses, real time streaming data.

**QuickSight**: It is a business Intelligent reporting tool, similar to tableau or if you’re a java programmer, similar to BIRT & is fully managed by AWS.

**Introduction to AWS Machine Learning**

**DeepLens**: AWS deeplens is a deep learning enabled video camera, it has a deep learning SDK, that allows you to create advanced vision system applications.

**SageMaker**: It is AWS’s flagshit machine learning product, it allows you to build and train your own machine learning models and then deploy them to AWS cloud and use them as a back-end for your applications.

**Amazon Rekognition**: Rekognition provides deep learning based analysis of video and images.

**Amazon Lex**: Lex allows you to build conversational chatbots, these can be used in many applications, such as first-line support for customers.

**Amazon Polly**: Polly provides natural sounding text-to-speech.

**Amazon Comprehend**: It can use deeper learning to analyze text for insights and relationships this can be used for customer analysis or for advanced searching for documents.

**Amazon Translate**: It can use machine learning to accurately translate text to a number of different languages.

**Amazon Transcribe**: It is an automatic speech recognition service, that can analyze audio files that are stored in Amazon S3 and then return the transcribe text.

**Introduction to AWS Security, Identity & Compliance**

**Artifact**: AWS artifact is an online portal that provides access to AWS security and compliance documentation and that documentation can be readily available when needed for auditing and compliance purposes.

**AWS certificate manager**: It issues SSL certificates for HTTPS communication with your website it integrates with AWS services such as route 53 and cloudfront and the certificates that are provisioned through AWS certificate manager are completely free.

**Amazon Cloud Directory**: Amazon Cloud directory is a cloud-based directory service that can have hierarchies of data in multiple dimensions unlike conventional LDAP based directory services that can only have a single hierarchy.

**AWS Directory Service**: It is a fully managed Microsoft Active Directory service in the AWS cloud.

**AWS CloudHSM**: It is a dedicated hardware security module in the AWS cloud this allows you to achieve corporate and regulatory compliance while at the same time greatly reducing your costs over using your own HSM in your own infrastructure.

**Amazon Cognito**: It provides sign in and sign up capability for your web and mobile applications you can also integrate that signup process with external orth providers such as Google and Facebook and also SAML 2 providers as well.

**Identity and Access management (IAM)**: IAM allows you to manage user access to your AWS services and resources in your account, users and groups of users have individual permissions that allow or deny access to your resources.

**AWS Organizations**: It provides policy-based management for multiple AWS accounts this is great for large organizations that have multiple accounts and they want to manage those and the users that use those accounts centrally.

**Amazon Inspector**: inspector is an automated security assessment service it can help to identify vulnerabilities or areas of improvement within your AWS account.

**Key Management Service (KMS)**: KMS makes it easy to create and control encryption keys for your encrypted data and it also uses hardware security modules to secure your keys it's integrated well with AWS services such as Amazon S3, redshift and EBS.

**AWS Shield**: AWS shield provides protection against distributed denial of service or DDoS for short protection against DDoS attacks the standard version of a double shield is implemented automatically on all AWS accounts.

**Web Application Firewall (WAF)**: WAF is a Web Application Firewall that sits in front of your website to provide additional protection against common attacks such as SQL injection and cross-site scripting it has different sets of rules that can be used for different applications.

**Introduction to AWS Developer Tools**

**Cloud9**: AWS Cloud9 is an integrated development environment running in the AWS cloud it allows you to deploy servers directly to AWS from an integrated development environment we will be using cloud9 extensively if you go on to the developer associate pathway.

**CodeStar**: AWS code star makes it easy to develop and deploy applications to AWS it can manage the entire CI CD pipeline for you it has a project management dashboard including and integrated issue tracking capability powered by Atlassian JIRA software.

**X-RAY**: Makes it easy to analyze & debug applications, this provides you a better insight to the performance of your application and the underlying services that it relies upon.

**CodeCommit**: AWS code commit is a git repository just like github and it's running in the AWS cloud.

**CodePipleline**: AWS code pipeline is a continuous integration & continuous delivery (CICD) service. It can build test and then deploy your code every time a code change occurs.

**CodeBuild**: AWS code build compiles your source code runs tests and then produces software packages that are ready to deploy on AWS.

**CodeDeploy**: AWS code deploy is a service that automates software deployments to a variety of compute services including Amazon EC2, AWS lambda and even instance that are running on premises.

**Introduction to AWS Media Services**

**Elemental MediaConvert**: AWS elemental media convert is a file based video transcoding service for converting video formats for video on-demand content.

**Elemental MediaPackage**: Media package prepares video content for delivery over the internet it can also protect against piracy through the use of digital rights management.

**Elemental MediaTailor**: Mediatailor inserts individually targeted advertising into video streams viewers receive streaming video with ads that are personalized for them.

**Elemental MediaLive**: Medialive is a broadcast grade live video processing service for creating video streams for delivery to televisions and Internet connected devices

**Elemental MediaStore**: Mediastore is a storage service in the AWS cloud that is optimized for media.

**Kinesis Video Streams**: Streams video from connected devices through to the AWS cloud for analytics machine learning and other processing applications.

**Introduction to AWS Mobile Services**

**AWS Mobile Hub**: Mobile hub allows you to easily configure your AWS services for mobile applications in one place it generates a cloud configuration file which stores information about those configured services.

**AWS Device Farm**: Device farm is an app testing service for Android iOS and web applications, it allows you to test your app against a large collection of physical devices in the AWS cloud.

**AWS AppSync**: AppSync is a graph QL Back-end for mobile & web applications. if you're a developer and you don't know what graph QL is then make sure you go out and find out because it is absolutely revolutionizing the way we think about data

**Introduction to AWS Migration Services**

**AWS Application Discovery Service**: AWS application discovery service gathers information about an enterprise's on-premises data centers to help plan migration over to AWS, data that is collected is retained in an encrypted format in an AWS application discovery service data store.

**AWS Database Migration Service**: orchestrates the migration of databases over to the AWS cloud you can also migrate data bases with one database engine type to another totally different database engine type for example you can migrate from Oracle over to AWS Aurora.

**AWS Server Migration**: Server Migration can automate the migration of thousands of on-premise workloads over to the AWS cloud this reduces costs and minimizes our downtime for migrations.

**AWS SnowBall**: AWS snowball is a portable petabyte scale data storage device that can be used to migrate data from on-premise environments over to the AWS cloud you can download your data to the snowball device and then send it to AWS who will then upload that to a storage service for you.

**Introduction to AWS Bussiness Productivity & App Streaming**

**Amazon WorkDocs**: Amazon work Docs is a secure fully managed file collaboration and management service in the AWS cloud the web client allows you to view and provide feedback for over 35 different file types including Microsoft Office file types and PDF.

**Amazon WorkMain**: Amazon WorkMail is a secure managed business email and calendar service.

**Amazon Chime**: Amazon chime is an online meeting service in the AWS cloud, it is great for businesses for online meetings video conferencing calls chat and to share content both inside and outside of your organization.

**Amazon WorkSpaces**: Amazon Workspaces is a fully managed secure desktop as a service it can easily provision streaming cloud-based Microsoft Windows desktops.

**Amazon AppStream 2.0**: Appstream is a fully managed secure application streaming service that allows you to stream desktop applications from AWS to an html5 compatible web browser, this is great for users who want access to their applications from anywhere.

**Introduction to AWS Internet of Things (IoT)**

**AWS IoT**: AWS IOT is a managed cloud platform that lets embedded devices such as microcontrollers and Raspberry Pi, to securely interact with cloud applications and other devices.

**Amazon FreeRTOS**: It is an OS for microcontrollers such as the microchip pic32, allows small low-cost low-power devices to connect with IoT.

**AWS GreenGrass**: AWS greengrass is a software that lets you run local AWS lambda functions and messaging data caching sync and machine learning applications on AWS IOT connected devices. AWS greengrass extends AWS services to devices so they can act locally on the data they generate while still using cloud-based AWS IOT capabilities.

**Introduction to AWS Games Development**

**Amazon Gamelift**: Amazon Gamelift allows you to deploy scale and manage your dedicated game servers in the AWS cloud.

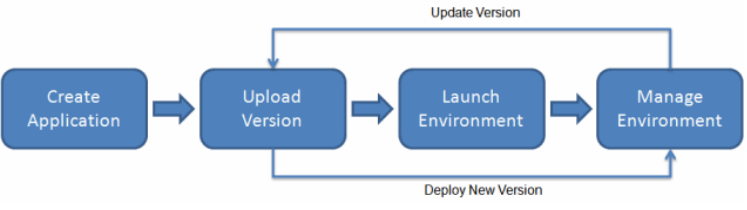
**Amazon Lumberyard**: Amazon Lumberyard you can see there we've got some images of some pretty cool stuff it's a game development environment and cross-platform triple-A game engine on the AWS cloud.

**Elastic Beanstalk**

With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS Cloud without worrying about the infrastructure that runs those applications. AWS Elastic Beanstalk reduces management complexity without restricting choice or control. You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring. Elastic Beanstalk uses highly reliable and scalable services that are available in the AWS Free Tier.

If you need to change your code after you have deployed it, it is quite easy to upload new versions of that code and that can be done through the console or the command-line interface and also complete environments can also be redeployed if need be

Elastic Beanstalk supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby, as well as different platform configurations for each language, such as Apache, Nginx, passenger and IIS. A configuration defines the infrastructure and software stack to be used for a given environment.

Elastic Beanstalk Example: 10. 010-10 Elastic Beanstalk- Foundations: Time- 3:15

**Deployment Options:**

* All at once
* Rolling (a batch at a time), Rolling with additional batch.
* Immutable (2 environments temporarily)
* Blue-Green (two environments):

They will have two environments running your application under the one elastic Beanstalk application and so what that is, you will have a blue environment and in green environment one of those could be a development environment and the other one your production environment.

So when you get to the stage where your development environment is ready to go to be deployed. To deploy that all you simply need to do is to switch over from one environment to the other environment and then your old environment will then become your new development environment.

It will simply allow you to switch the domain names for those two environments automatically for you and so that makes sure that your changeover doesn't read or doesn't involve any downtime for returning of requests

**AWS 6 advantages and benefits of cloud computing**

* Trade capital expense for variable expense
* Benefit from massive economies of scale
* Stop guessing capacity and cost (AWS have dynamic storage and AWS monthly Calculator, Total Cost of Ownership)
* Increase speed and agility
* Stop spending on running and maintaining data centers
* Go global in minutes
* Amazon Inspector

Amazon Inspector:

* Automated security assessment service.
* Reduces cost and increases effectiveness of security assessments and compliance,
* Pricing starts at .30$ per agent-assessment per moth with volume discounting to achieve as low as 0.05$ per agent-assessment per month

Architecture Example: 14. 020.

Steps:

1. Purchasing the domain Name with Route 53
2. Creating a S3 bucket and hosting our Website (by uploading the whole site code on it)
3. Creating SSL Certificate with AWS Certificate manager
4. Creating a CloudFront Distribution
5. Routing Traffic with Route 53
6. Cloud Invalidate (To update or to make changes)
7. Redirecting website with S3 Bucket redirection

**Shared Responsibility Models**

**AWS**: they are going to be responsible for security of their AWS cloud. AWS is going to be responsible for all of that low level architecture. They're going to be responsible for those regions of availability zones edge locations and those services that they're going to run on that.

**Customer**: they're going to be responsible for what you put in the AWS cloud for example,

you have a responsibility to make sure that what you put in that AWS cloud is secure for example if you have an ec2 instance and you have sensitive data that is being controlled by their ec2 instance you're going to have to make sure that you look after client-side encryption of that data

Server-side encryption of that data if it has drives attached to it or or devices attached to it you need to make sure that that data is secure you also have to make sure that the network traffic coming backwards and forwards from that ec2 instance is secure for example you might have a word-press application that has sensitive information for a customer

You're also going to be responsible for the Linux operating system or the Windows operating system that is going to be running on that ec2 server

You're going to be responsible for not only the Linux firewall configuration but also on the AWS side with the security groups.

You're going to be responsible for that WordPress application that you're running you're going to have to make sure that the administrator access to that application has good password protection.

You need to make sure that the users that are accessing your AWS account will have good Identity and Access Management on there and they have least privileges granted to them.

You've got that customer data you need to make a decision as to what you are going to store as far as sensitive customer data goes.

**Infrastructure Service**

This is the lowest level of service with AWS. AWS and that includes the Amazon ec2 related services such as elastic block storage auto-scaling Amazon virtual private cloud.

So you control the operating system, you've got a Linux server there, you look after that Linux operating system, you make sure that it's up-to-date, you make sure that it is secure.

You're also looking after the Identity and Access Management of your customers that are accessing that ec2 server.

**Container Service**

This service is a little bit more abstract than Infrastructure Service, includes Amazon RDS, Amazon Elastic MapReduce, and AWS Elastic Beanstalk.

AWS are looking after the actual platform that's running on there, the operating system so that Linux operating system, you're not going to be looking after that with Amazon RDS, you're not going to be looking after patching that my sequel database application that's running on.

So you're responsible for the network controls you're responsible for your firewall configuration, your security groups and platform level security, for example if you have got a MySQL database you need to make sure that your administration for that database is actually secure.

**Abstract Service**

High-level database storage and messaging services: may include Amazon S3, Amazon Glacier archiving, Dynamodb, AWS Lambda, Amazon SQS and Amazon SES

AWS is going to be managing the underlying service components or the operating system on which they reside, so if Amazon S3 they're going to be taking a lot more control around a server side encryption you just set that up it through the Console, but AWS manages that for you, network traffic protection to Amazon S3, AWS will be managing that, AWS will also be managing a lot of the Identity and Access Management with Amazon S3.

**Identity and Access Management**

Well, if we have an account for an enterprise, that may have thousands of employees,

* So how do we manage all those individual employees.
* How do we make sure that they've got the correct access levels for our account?
* How do we manage people that are outside of our enterprise?
* How do we manage users of our application that might be hundreds of thousands of the users that are accessing a web application?

Access Management allows us to control all of that.

It's a web service you run it through the console and it allows you to securely control individual and groups of individuals access to your AWS resources, you do that by creating and managing user identities which are called IAM users and then you grant those users permissions.

* Shared access to your AWS account
* Granular permissions
* Secure access to AWS resources for applications that run on Amazon EC2
* Identity federation to grant permission for users outside of AWS
* Payment Card Industry (PCI) and Data Security Standard (DSS) Compliance
* Access log auditing using CloudTrail
* Eventually Consistent
* Free to use

**IAM Policies**

IAM policies allow you to give fine-grained control around what a user can do, so by default user cannot access anything in your account.

So you need to grant permission for that user for policies that you associate with the user, and they define the effect, action, resources and optional conditional, for the privileges for the user.

{“Version”: “2012-10-17”,

“Statement”:

{ “Effect”: “Allow”,

“Action”: “Dynamodb.\*”,

“Resource”: “arn:aws:dynamodb:us-west-2:123456789012:table/Books” } }

**Users**: When someone joins your organization, you would first set them up, as a user on your account.

* Represent the person or service accessing your account
* Consists of a name and credentials
* Users are identified by:
  + A “friendly name”, eg, “Bill”
  + Amazon Resource Name (ARN)
    - “arn:aws:iam::account-ID-without-hyphens:user/Bill”
  + Unique identifier which is return only when you use the API, SDKs, tools for Windows PowerShell, or AWS CLI to create the user.
* Credentials can be associated to a user:
  + Console password- Simple username and password, user will have a URL link to login to the console.
  + Assess Keys- assess key ID and a secret assess key, (max 2 per user).
* *Never use root user* to access resources unless absolutely essentials. Create admin users with required permissions, always enable multi0factor authentication of the root user.

**User Password Policies**

* Set a minimum password length.
* Require specific character types.
* Allow all IAM users to change their own passwords.
* Password Expiration
* Prevent users form reusing previous passwords.
* Prevent users from reusing previous password.
* Force users to contact an account admin, when the password has expired.

**Sign-in URL**

Your sign-in page URL has the following format, by default:

https://your\_aws\_account\_\_ID.signin.aws.amazon.com/console

The URL for your sign-in page to contain your company name (or other friendly identifier) instead of your AWS account ID, you can create an alias for your AWS account ID:

https://your\_alias.signin.aws.amazon.com/console/

we can also put users in groups, for example: Group Admin, Group Developers, Group test

**Roles**

* Defined permissions that can be assumed by the user or resources.
* Allow EC2 instance to access other AWS resources.
* Grant access to your resources to users in another AWS account.
* Can be used to allows users to temporarily assume a role with least privilege access to critical resources.
* Identity federation using:
* AWS Cognito
* OAUTH (Facebook, Google, etc)
* Enterprise Single Sign ON with LDAP or Active Directory

AWS organizations

* Allow multiple AWS accounts used by an organization to be part of an Organizational Unit (OU).
* Service Control Policies (SCPs) allow the whitelisting or blacklisting of services within an organizational unit.
* A blacklisted service will not be available even if the IAM user or group policy allows it.
* Benefits:
  + Centrally Manage- Policies across multiple AWS accounts
  + Control access- to AWS services
  + Automate- AWS account creation and management programmatically with APSs
  + Consolidate billing- across multiple AWS accounts

**AWS CloudTrail**

* AWS Management Console, SDK and CLI all use the AWS API to communicate to AWS services.
* AWS can log calls to AWS services from the AWS API.
* Logs are stored in a bucket and can be analyzed (Amazon Athen, EMR, etc)
* SNS topic can alert security issues.