AWS Cloud Solutions Architect Associate Course:

**AWS – History So Far:**

“Invention requires two things: ability to try a lot of experiments, and 2. Not having to live with the collateral damage of failed experiments.”

2003 – paper on what Amazon’s own internal infrastructure should look like

selling it as a service and prepared a business case.

SQS launched 2004

AWS launched 2006

2007 had over 180,000 developered

2010 all of amazon.com moved over

2012 Re-Invent conference

2013 Certifications Launched

2014 Committed to achieved 100% renewable energy

2015 AWS breaks out its revenue. $6 billion per year

2016 Run rate of 13 billion USD

2016 Cisco, Dell, IBM had a 15 billion reduction

New service announcements grow substantially every year.

Subscribe to YouTube on Cloudguru, Jeff Bar’s Blog and twitter of ACloudGuru.

May 2015 AWS was named as a leader in the IaaS for the 5th consecutive year in the leader category. Same for 2016.

CenturyLink VmWare and Softlayer are now niche players.

**10,000 ft overview:**

Messaging, Desktop& App Streaming, Security & Identity, Management Tools, Storage, Databases, Networking & Content Delivery, Compute, and AWS Global Infrastructure will need to be known for **the exam.**

Global Infrastructure

* **Regions** – where availability zones reside. Geographical area. Consists of 2 or more availability zones
  + some regions don’t have all the services in them.
* **Availability Zones** – simply a data center. Isolated from other availability zones.
* **Edge Location** – CDN end point for CloudFront. A way to cache media objects in the cloud. More edge locations than regions. POP for CloudFront

Networking & Content Delivery

* **VPC (Virtual Private Cloud)** – virtual datacenter. Multiple VPC’s to a region and connect other VPC’s to another. **Important on exam.** How to build a VPC from memory?
* **Route53 –** DNS service. Can register domain names through this. 53 is the DNS port.
* **CloudFront** – Part of CDN. Edge locations that cache assets.
* **Direct Connect** – a way to connect an office or physical datacenter to AWS directly with a dedicated telephone line. **Can come up in the exam.**

Compute

* **EC2 (Elastic Compute Cloud)** – virtual machines in the cloud.
  + SSH/RDP
* **EC2 Container Service** – container management service geared towards Docker containers. Don’t have to install cluster management system. Doesn’t come up in the system.
* **Elastic Beanstalk** – comes up in developer exam. Upload code and then this service will deploy the needed infrastructure for the application.
* **Lambda** – Serverless, upload code and code will respond to events. Not in exams yet.
* **Lightsail** – out of the box cloud. Wordpress/etc.

Storage

* **S3 (Simple Storage Service)** – virtual disk in the cloud which can store object. Files, pictures, movies, etc. Does come up in the developer and CSAA exam. Object based storage.
* **Glacier** – place to archive your files off of S3.
* **EFS** (Elastic File Service) – Block based storage which can be shared. Could install databases/applications and share it with multiple VM’s.
* **Storage Gateway –** a way to connect S3 to your physical datacenter. VM on premise but pulls the data from S3.

Databases

* **RDS (Relational Database Service)** – MariaDB, SQL, MySQL, Oracle, Comes up in the CSAA
* **Aurora** – MySQL, PostGres
* **DynamoDB** – non-relational database. NoSQL. Developer exam
* **Redshift** – datawarehousing solution. Big Data.
* **Elasticcache** – a way to cache your data in the cloud.

Migration

* **Snowball** – initially import/export. Briefcase size appliance for loading the data on it. S3/EBS transfer. Traditionally consisted of just storage. Added compute capacity to it which can be taken on premise. SysOps/Associate exam.
* **DMS (Database Migration Services)** – migrate databases on premise or databases in the cloud for shifting them over. Migrate databases inside AWS to other regions or into Redshift. Don’t have to stay with the database you’re migrating from. Handles the entire conversion process. Zero downtime. Oracle, Aurora, SAP ESE, MySQL, SQL. Not in exam yet.
* **SMS (Server Migration Services)** – migrate VM (vmWare) on premise to replicate them to the cloud and can do 50 concurrent at the same time. Not on exam yet.

Analytics

* **Athena** – run SQL queries on S3. JSON files. Flat files turning them into searchable details.
* **EMR (Elastic Map Reduce)** – Produce large amount of data. Web indexing. Using Hadoop and other frameworks. Big Data.
* **Cloud Search –** Managed service. Create search capabilities with your website.
* **Elastic Search** – open source framework. Create search capabilities with your website.
  + Angola
* **Kinesis** – A way of analysis streaming data and storing data like financial transactions. Social media streams.
* **Data Pipeline** – Data that allows to be transferred from one location to another.
* **Quick Sight** – Business analytics tool, create visualizations and can use data from S3. Dashboards.

Security & Identity

* **IAM (Identity Access Management)** – comes up in every exam. Basic way to sign in/authenticate. Group users
* **Inspector** – agent installed on VM. Inspects them. Does reporting
* **Certificate Manager** – free SSL certs for your domains
* **Directory Service** – way to connect AD to AWS.
* **WAF (Web Application Firewall)** – application level protection to your website. SQL injection, cross-site scripting.
* **Artifacts** – where to get your documentation in the console.

Management Tools

* **Cloud Watch** – monitor performance on AWS environment. Monitor disk utilization, filters.
* **Cloud Formation** – a way to turn your infrastructure into code. Document that describes your environment. Templates. Know inside and out. Doesn’t come up in exam.
* **Cloud Trail** – a way to audit changes to your AWS environment.
* **Opsworks** – automating deployments via chef.
* **Config** – a way to monitor and automate your environment. You can set alerts.
* **Service Catalog** – allows enterprises to build out what can be authorized and not authorized.
* **Trusted Advisor** – a way to recommend tips like cost optimization, etc.

Application Services

* **Step Functions** – a way to visualize what is happening inside your application or microservices.
* **SWF (Simple Workflow Service)** – a way to coordinate human ordinated tasks and non-human tasks.
* **API Gateway** – a door to create publish, secure API’s at scale. A way for apps to access backend data.
* **AppStream** – streaming desktop applications to your users
* **Elastic Transcoder** – changes video format to suite multiple devices.

Developer Tools

* **CodeCommit** – Github. A place to store your code like Github
* **CodeBuild** – a way to compile your code
* **CodeDeploy** – a way to deploy your code to EC2 instances
* **CodePipeline** – a way to keep track of all versions of your codes.

Mobile Services

* **Mobile Hub** – let’s you add/configure features for your apps. Backend logic, notfications. Own console for the apps.
* **Cognito** – easy for users to sign in and sign up for your apps. Gmail credentials for example.
* **Device Farm** – provides a way to enhance usable android/ios apps. Can test on hundreds of real smart phones.
* **Mobile Analytics** – Cost effective collect mobile app usage data
* **Pinpoint** – enables you to understand and engage with your users. Gather data on what users are doing what with your apps. Can look at human behavior. Can market to your users.

Business Productivity

* **WorkDocs** – store your important work documents in the cloud.
* **WorkMail** – exchange for AWS. A way to send/receive email

Internet of Things

* **iOT** – millions of devices out there and a way to keep track of them.

Desktop & App Streaming

* **WorkSpaces** – VDI, a way of having your desktop in the cloud.
* **AppStream 2**.**0** – a way to stream desktop applications to your users.

Artificial Intelligence

* Superintelligence book by Nick Bostrom
* **Lex** – don’t need an echo to communicate with alexa
* **Polly** – text to speech in 47 different languages. Mp3 file. Synthesis Speech language.
* **Machine Learning** – Give AWS a data set and what the outcomes are. Uses machine learning to predict outcomes based on the data center. Predict data based on previous data.
* **Rekognition** – Upload a picture. Provides tags, facial recognition.

Messaging

* **SNS (Simple Notification Service)** – email, text message, HTTP endpoints
* **SQS (Simple Queue Service)** - a way of decoupling your applications. Post jobs to a queue. If message dies, it stays in the SQS queue. No tightly coupled dependencies.
* **SES (Simple Email Service)** – a way of sending/receiving email in this service.

**IAM (Identity Access Management) 101**

* Allows you to manage users and their level of access to the AWS Console.
* Gives you centralized control of your AWS account
* Shared access to your AWS account
* You can peer AWS accounts to other accounts
* Granular Permissions
* Identity Federation (AD, FB, LinkedIn). SSO
  + Create apps that authenticate against FB, etc.
* Multifactor authentication
* Provide temporary access for users/services when necessary.
* Allows you to setup your own rotation policy.
* Integrate with other AWS services.
* Supports PCI DSS compliance – taking online payments and storing credit card details yourself.
* **Users** – End Users (think people)
* **Groups** – collection of users under one set of permissions
* **Roles** – create roles and can then assign them to AWS resources.
* **Policies** – a document that defines one or more permissions (JSON) – Attribute/Value

Lab:

**Global regions can come up on the exam for IAM. It can be accessed in any region.**

Root account is your email address you used to create your account.

**Access key id and secret access key id can only be used for programmatic access on IAM.**

You can attach more than one policy to a user and group.

New users have no permissions when created.

Can only view access key and secret access key once.

Always setup MFA on your root account.

Read AWS Security Best Practices - AWS Security Best Practice

**S3 Storage 101**

Simple Storage Service provides developers and IT teams with secure, durable, highly-scalable object storage. S3 is easy to use, with a simple web services interface to store and retrieve any amount of data from anywhere on the web.

Place to store files in the cloud. Object based (Video’s, word docs, flat files). Data is spread across multiple devices and facilities. Designed to sustain the loss of 2 facilitiesBlock storage (OS, database).

Allows you to upload files. 0 to 5 TB per files. Unlimited storage. Files stored in buckets

Name must be unique globally as it’s a universal namespace. 5 GB maximum PUT in a single upload. Multipart upload if going over 100 MB. Multi-object delete can be done and you aren’t charged for it. 100 buckets per account but can be increased. Data not charged for going between EC2 or S3 in the same region. S3 will be charged by being accessed by another account but you can choose to charge the other person.

**Always receive an HTTP 200 status code**

**Read S3 FAQ before taking the exam.**

* **Key** – name of object (Can put data in alphabetical order. Can add dates to make sure their unique. )
* **Value** – simply the data and is made up of a sequence of bytes
* **Version ID** – Version
* **Metadata** – data about the data you are storing
* **Subresources** – exist underneath an object
  + **ACL’s**
  + **Torrent**

Built for 99.99% availability

Guarantee 99.9% availability

Amazon Guarantee’s 99.9^11 durability

Tiered storage options

Lifecycle Management

Versioning

Encryption

Secure your data using Access Control Lists and Bucket Policies. Only owners of a bucket can permanently delete a version.

Best practice: S3 data to include secure access permissions, cross-region replication, versioning, and tested backups.

S3 uses a combination of Content-MD5 checksums and cyclic reduandcey checks (CRC) to detect data corruption.

Event notifications are free but you can perform actions like transcoding media, processing data files and syncing objects with other data stores.

**S3** – 99.99% availability, 99.99^11% durability, stored redundantly across multiple devices in multiple facilities and is designed to sustain the loss of 2 facilities concurrently.

* MFA delete can be enabled on buckets.
* Can have up to 10 tags on an object.
  + Tags are charged $0.01 per 10,00 tags.
* **S3 Class Analytics** – can analyze storage access patterns and transition the right data to the right storage class.
* **S3 Inventory** – provides a CSV flat-file output of your objects and metadata.
* Storage metrics in cloudwatch show up 15 minutes of being enabled.
* Cross-Region Replication – metadata and ACL’s are replicated along with data
  + Existing data with the bucket is not replicated.
* IPv6 doesn’t work with website hosting or access via bittorrent. IPv6 is available except in China.
  + Doesn’t require additional on premise networking hardware to work with S3.
  + Can point to dual-stack endpoints. IPv6 can work with server access log.
  + Bucket policies and IAM policies to accommodate IPv6 before switching to Dual-stack.

**S3 – IA(infrequently accessed)** –for data that is accessed less frequently, but requires rapid access when needed. Lower fee than S3, but you are charged a retrieval fee.

* 99.9% availability & 99.999999999% durability
* can use lifecycle policy or specify x-amz-storage-class header to put in S3-IA
* minimum object size of 128KB.

**Reduced Redundancy Storage** – Designed to provide 99.99% durability and 99.99% availability of objects over a given year. (reproducible data like thumb nails.)

**Glacier** – very cheap, but used for archival only. It takes 3-5 hours to restore from Glacier.

Data Consistency:

* **Read after Write consistency for PUTS of new Objects**
* **Eventual Consistency for overwrite PUTS and DELETES (can take some time to propagate) Atomic – get new or old data. Won’t get partial data.**

<https://s3-eu-west-1.amazonaws.com/acloudguru>

* **region – bucket name at end.**

**Glacier** – extremely low-cost storage service for data archival. Stores data as little as $0.01 per gigabyte per month and is optimized for data that is infrequently accessed and for which retrieval times of 3 to 5 hours are suitable.

* Retrieved data gets put into RRS for a temporary period and temporary copy.
* Different retrieval methods (Expedited, standard, or bulk
* Requires 32 KB of data per object of index and metadata for storage.
* You can retrieve 10 GB of Glacier per month for free.
* You can delete day after a 3 month period for free otherwise you get charged within the 3 month period.

Charges:

* Storage
* Requests
* Storage Management Pricing
* Data Transfer Pricing. Data in S3 is free. Data around in S3 is charged
* **Transfer Acceleration** – enables fast, easy and secure transfers of file over long distances between you end users and an S3 bucket. Transfer Acceleration takes advantage of Amazon CloudFront’s globally distributed edge locations. As the data arrives at an edge location, data is routed to Amazon S3 over an optimized network path. Users upload to edge location

Lab:

Bucket names can only be created with lowercase characters.

Can run a static website hosting. Tags, cross-region replication, logging.

Lifecycle allows objects to be moved over to different S3 Tiers.

Can allow permissions on specific buckets, etc.

**Have to make an object public before being able to view it**.

Use permissions at the object or bucket level.

Lab:

**Cant delete versioning once it is enabled.**

Versions will be the sum of all the different documents in one in the bucket.

**If you delete a version, it can’t be restored. If you delete and object, it can be restored.**

**If an object is deleted, it shows as a delete Marker.**

**Versioning MFA Delete capability** – uses multi-factor authentication can be used to provide an additional layer of security.

Lab:

Prefix means sub folders in bucket.

Doesn’t support multiple replication region buckets

Delete markers/other versions don’t replicate to other regions. Unless you delete the object will it come up in the replicated region.

Versioning must be turned on for cross replication on target and source endpoints.

Files in an existing bucket are not replicated automatically. All subsequent updated files will be replicated automatically.

<http://docs.aws.amazon.com/AmazonS3/latest/dev/crr-what-is-isnot-replicated.html>

All previous versions are replicated.

Lab:

Versioning can be on with lifecycle management or not. **Lifecycle rules** – help manage your storage costs by controlling the lifecycle of your objects.

Can’t recover permanently deleted items.

Glacier has a 90-day minimum of storage duration.

Can be used in conjunction with versioning. Can be applied to current version or previous versions.

**CloudFront CDN Overview**

A system of distributed servers that deliver webpages and other web content to a user based on the geographic location of the user, The origin of the webpage and a content delivery server.

* **Edge Location** – location where content will be cached. This is separate to an AWS Region/AZ. **Cache is TTL. Not just for read only. Clearing cache objects means you’ll be charged. TTL is 24 hours.**
* **Origin** – origin of all the files that the CDN will distribute. This can be an S3 Bucket, EC2 Instance, Elastic Load Balancer or Route 53.
* **Distribution** – this is the name given the CDN which consists of a collection of edge locations.
* **Cloud Front** - can be used to deliver your entire website, including dynamic, static, streaming, and interactive content using a global network of edge locations. Requests for your content are automatically routed to the nearest edge location, so content is delivered with the best possible performance. Optimized to work with other AWS services. Works seamlessly with any non-AWS origin server which stores the original, definitive versions of your files. Can use multiple origins.
* **Web Distribution** – Typically used for websites.
* **RTMP** – Used for Media Streaming.

<https://aws.amazon.com/cloudfront/details/>

**Lab:**

Can restrict users to use cloudfront only versus the s3 url. If objects consistently change, it would be good to change the TTL if it happens often. Acloud guru uses signed urls that are geared towards users and can’t be shared with users based on who or who didn’t pay.

Supports HTTP/2 and IPv6.

**You can enable Geo-restriction with either whitelist/blacklist. You can only have one or the other.**

**Invalidating objects can remove them from CloudFront edge locations. Costs money by removing it Stops content from being cached before the TTL if it is in process. Used not only for downloading not uploading.**

Hi Team,

The have been a number of Forum posts and Support contacts about problems with this Lab. I have been able to recreate the problem and find a resolution.

In a word, Patience.

The Lab does work, however there appears to be a problem that the delay caused by the distribution's settings and DNS values replication causes the initial accesses to fail and cache the 'failure'.

While the Distribution is being created and activated, CloudFront will simply redirect back to S3. However if the objects in S3 are not available an 'AccessDenied' is cached. This behaviour also appears to be inconsistent which doesn't help.

I have confirmed that by waiting (I have been testing for at least an hour) or Disabling and then Enabling will also allow the system to stabilize.

Here are my recommendation as a work around until we get to the bottom of this.

A/ Do not attempt to access the objects in the S3 bucked until after the CloudFront distribution is fully "Deployed". And even then I suggest waiting another 15 - 20 minutes.

B/ Delay the UpLoading of your test images until after the Cloud Distribution is Deployed.

C/ If you are stuck now. UpLoad a different image and use the new Image for testing

Here is some additional info.

https://forums.aws.amazon.com/thread.jspa?threadID=216814

Rusty

**S3 – Security & Encryption**

All buckets are private.

* Access controls can be setup by bucket policies or ACL’s.
* S3 buckets can be configured to create access logs which log all requests made to the S3 bucket. Can be done to another bucket.

**In Transit** – when sending information to and from buckets

* **SSL/TLS**

**At Rest** –

* **Server Side Encryption**
  + **S3 Managed Keys** **– SSE-S3** (encrypt it with a master key. AWS handles keys) AES-256
  + **AWS Key Management Service, SSE-KMS** ( Envelope key protects the encryption key from unwanted access, audit trail when keys are used, or you can use own key)
  + **Service Side Encryption with Customer Provided Keys – SSE-C –** Management of key is done by the key.
* **Client Side Encryption** – encrypt on the client side then upload to S3.

**Storage Gateway**

**Storage Gateway** is a service that connects an on-premises software appliance with cloud-based storage to provide seamless and secure integration between an organization’s on-premises IT environment and AWS’s storage infrastructure. The service enables you to securely store data to the AWS cloud for scalable and cost-effective storage.

* Available for download as a VM image that you install on a host in your datacenter
* Supports VMware ESXi or Hyper-V
* Could be on a VM on EC2 instance in AWS

Types:

* **File Gateway (NFS)** – store flat files in S3.
  + Files are stored in S3. Accessed through NFS (Network File System) mount point. Once objects are transferred to S3, they can be managed natively in S3
* **Volume Gateway (iSCSI)** – block based storage.
  + - Presents applications with disk volumes by iSCSI. Data can be asynchronously backed up as PIT snapshots and stored in EBS snapshots.
    - EBS virtual block disk.
    - **Snapshots** – **incremental backups** that capture only changed blocks. Compressed to minimize your storage charges.
  + **(Gateway) Stored Volumes** – store entire copy of data set on premise.
    - * Store data locally and asynchronously backing up data to AWS. Provide durable off-site backup. Backed to S3 in the form of EBS snapshots. 1GB to 16TB in size. 100% copy on site and incremental to S3.
  + **(Gateway) Cached Volumes** – store most recently accessed data.
    - * Frequently accessed data (Cached volumes) is kept locally with S3 being your primary data storage. Can create up to 32 TB size and attach the volumes as iSCSI devices to on-premise application servers. 1 GB – 32 TB in cached volumes. Volumes are written to S3. Most likely EBS volumes in S3.
* **Tape Gateway (VTL (Virtual Tape Library))** – backup and archive solution that creates virtual tapes to send them to S3
  + Use existing tape-based backup application to store data on virtual tape cartridges that you create on your tape gateway. Preconfigured with media changer and tape drives which are available as iSCSI devices. Supported by Veam, NetBackup, etc. Sent to S3 and then shelfed to Glacier

**Snowball**

**Import/Export Disk service** – accelerate moving large amount of data in and out of AWS cloud with portable storage devices for transport. Transfers data directly onto and off of storage devices using AWS internal network and bypass internet. Still available.

**Snowball** – PB data transport solution that uses secure appliance to transfer large amount of data into and out of AWS. Cost is 1/5 that of High-speed internet. 80 TB in all regions. Supports TPM and 256-bit encryption with tamper-resistant enclosures. AWS performs software erasure after all the data is transferred. Import to S3 or export from S3.

**Snowball Edge** – contains 100 TB with onboard storage and compute capabilities. Little datacenter to bring on premise. You can run lambda functions locally as well. Supports local workloads in remote locations or transfer storage. Can cluster together and form a local storage tier, etc.

**Snowmobile** – PB or EB of data. Transfer 100PB per snowmobile. 45-foot long shipping container pulled by a semi-trailer truck. Easy to move massive amounts of volumes. Could take up to 6 months.

Lab:

There is a snowball client that can be downloaded.

./snowball start –I 192.168.1.116 –m manifest file –u access code

Snowball stop

**S3 Transfer Acceleration**

**S3 Transfer Acceleration** - utilizes the CloudFront Edge network to accelerate you’re uploads to S3. You can use a distinct URL to upload directly to an edge location

* Does cost a little more money
* Further the way you are the faster the speed increase you will see.
* <bucketname>.s3-accelerate.amazonaws.com or <bucketname>.s3-accelerate.dualstack.amazonaws.com
* If transfer acceleration isn’t faster, you won’t be charged for it.
* Can help in scenarios with snowball, direct connect, 3rd party gateway, and storage gateway.
* HIPPA eligible.

Lab (S3 Static Web Hosting):

Know Static website hosting for developer exam. Bucket name. s3-website.region. Serverless. No php .NET framework, etc. Cheap to do.

**Summary:**

S3 (Durable, immediately available, frequently accessed)

S3-IA (Durable, immediately available, infrequently accessed)

S3-Reduced Redundancy Storage (data easily reproducible like thumb nails)

Pay for all versions in S3.

LifeCycle Management – transition to Standard –IA have to be 128 Kb in size and 30 days.

You can upload files to S3 much faster by enabling multipart upload. **Read S3 FAQ.**

**EC2 – The Backbone of AWS 101**

**EC2** – web service that provides resizable compute capacity in the cloud. Reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. EC2 (Elastic Compute Cloud) **Read FAQ on EC2**

* Changes the economics of computing by allowing you to pay only for capacity that you actually use. EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.

EC2 options:

* **On Demand** – allow you to pay a fixed rate by the hour with no commitment.
  + users that want low cost and flexibility without any up-front payments or long-term commits.
  + applications with short term, spiky, or unpredictable workloads that cannot be interrupted
  + Applications being developed or tested on Amazon EC2 for the first time.
* **Reserved** – provide you with a capacity reservation, and offer a significant discount on the hourly charge for an instance. 1 year or 3 year terms.
  + Applications with steady state or predictable usage
  + Applications that require reserved capacity
  + Users are able to make upfront payments to reduce their total computing costs even further.
* **Spot** – enable you to bid whatever price you want for instance capacity, providing for even greater savings if your applications have flexible start and end times.
  + what the price is of the stock today. Changes on region and AZ. Stock Market.
  + spot price and bid price are the same. Instance will be purchased.
  + spot price is above bid price then EC2 instance will be terminated.
  + Applications that are only feasible at very low compute prices.
  + Users with urgent computing needs for large amounts of additional capacity.
  + **If a spot instance is terminated by AWS, you won’t be charged for a partial hour of usage. However, if you terminate the instance yourself, you wil be charged for any hour in which the instance ran.**
* **Dedicated Hosts** – physical EC2 server dedicated for your use. DH can help you reduce costs by allowing you to use your existing server-bound software licenses.
  + Useful for regulatory requirements that may not support multi-tenant virtualization.
  + Great for licensing which does not support multi-tenancy or cloud deployments.
  + Can be purchased on-demand (hourly)
  + Can be purchased as reservation for up to 70% off of the dedicated host.

EC2 Instance Types

* D2 – Dense Storage – Fileservers/data warehousing/hadoop
* R4 – Memory Optimized – Memory Intensive Apps/Dbs
* M4 – general purpose – application servers
* C4 – Compute optimized – CPU intensive Apps/DBs
* G2 – Graphics Intensive – Video Encoding/3D Application Streaming
* I2(IOPS) – High speed Storage – NoSQL DBs, Data Warehousing
* F1(FPGA) – Field Programmable Gate Array – Hardware acceleration for your code
  + Can change underlying hardware to match software needs
* T2(Cheap) – lowest cost, general purpose – web servers/small dbs
* P2(Pics) – Graphics/General Purpose GPU – Machine Learning, Bit Coin Mining
* X1(Extreme Memory) – Memory Optimized – SAP Hana/Apache Spark

DR Mc GIFT PX

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html>

**EBS** – allows you to create storage volumes and attach them to EC2 instances. Once attached, you can create a file system on top of these volumes, run a database, or use them in any other way you would use a block device. EBS volumes are place in specific AZ where they are automatically replicated to protect you from the failure of a single component. (Storage array going down) Stays in an AZ not multiple regions.

* **General purpose SSD (GP2)**
  + Balances price and performance
  + Ratio of 3 IOPS per GB with up to 10,000 IOPS and the ability to burst up to 3000 IOPS for extended periods of time for volumes under 1GB.
* **Provisioned IOPS SSD (IO1)**
  + I/O intensive applications like NoSQL or large relational database
  + If you need more than 10,000 IOPS
  + Can provision up to 20,000 IOPS
* **Magnetic Storage (Throughput Optimized HDD (ST1))**
  + Big Data, Data warehouses, log processing
  + Can’t be a boot volume
  + Frequently accessed workloads
* **Cold HDD (SC1)**
  + Lowest cost storage for IA workloads
  + File server
  + Cannot be a boot volume.
  + Less frequently accessed data
* **Magnetic (Standard)**
  + Lowest cost per GB of all EBS volume types that is bootable
  + Ideal for workloads where data is access IA.

You can’t mount 1 EBS volume to multiple EC2 instances, instead use EFS

Lab:

AMI are snapshots of machines. PV(Peravertal) HVM. Persistent request – is to run the machine every single weekend. Each account has its own default VPC to start with. 1 subnet is equal to one AZ. Monitoring is done every 5 minutes on CPU, network, etc. Detailed monitoring is done every 1 minute. Delete on termination – each volume is deleted upon deletion. Tags help control costs.

Security Groups are virtual firewalls. Source – restrict down to your own personal IP address. 0.0.0.0 means coming from anywhere.

Public key – pad lock

Private key – unlocks the pad lock.

They use public IP to SSH in. Do need to change the permissions of the .pem file.

Lab 2:

**System Status Check** – verify that the instance is reachable. Could be an issue with the infrastructure if this isn’t working.

**Instance Status Check** – checking to see if you can get traffic to flow to the OS.

Know default monitors that come with CloudWatch.

C**an’t encrypt your default root EBS volume. Can encrypt root EBS volume by cloning the instance and encrypting it that way shortly after. You can use a third party tool to encrypt the root volume.**

**Termination protection is turned off by default.**

**EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated.**

**Additional volumes can be encrypted.**

[**https://aws.amazon.com/blogs/aws/new-encrypted-ebs-boot-volumes/**](https://aws.amazon.com/blogs/aws/new-encrypted-ebs-boot-volumes/)

Lab 3:

Putty doesn’t support pem files and uses ppk files. Putty key generator. Ec2-user is the name for SSH before @.

Lab 4:

**Security Group** – is a virtual firewall. You can have multiple security groups. Rules can be added to each security group. Rules applied to security groups happen automatically. Security groups are **stateful**. Network ACL are stateless. Once an inbound rule is added, outbound rules are automatically applied. Can’t specifically deny traffic on ports, etc. instances within a security group and communicate with one another. Can have more than one security group assigned to an instance. All inbound traffic is blocked by default. All outbound traffic is allowed. Can’t block specific IP addresses using security groups instead use Network ACL’s.

Lab 5:

Lsblk shows us all the volumes attached to the instance. If you create a volume in a different AZ other than where the instance it, you wouldn’t be able to attach it to the instance. File –s /dev/xvdf shows if there is data on the volume or not. Only data that is changed on a snapshot is stored in S3. Volumes created from snapshots can change the disk type as needed. Mkfs –t ext4 /dev/xvdf

Volumes exist on EBS. Snapshot is stored on S3.

Lab 6:

**RAID, Volumes & Snapshots**

**RAID** – Redundant array of independent disks.

* **RAID 0** – Striped, No redundancy, good performance.
* **RAID 1** – Mirrored, Redundancy
* **RAID 5** – good for reads, bad for writes, AWS does not recommend ever putting RAID 5’s on EBS.
  + Parity is a checksum.
* RAID 10 – Striped & Mirrored, good redundancy, good performance

RAID’s help provide more disk I/O that you’re not likely getting with just a single volume.

Username – administrator for windows. Mstsc – Microsoft terminal services connection.

**Snapshot** – excludes data held in the cache. Take an application consistent snapshot.

* **Free file system**
* **Unmount RAID array**
* **Shut down the EC2 instance.**

Lab 7 – **Encrypted root device**:

You can rent or purchase AMI’s from the marketplace. Encrypted AMI’s will be private.

* Have to stop an instance before taking a snapshot.
* Snapshots of encrypted volumes are encrypted automatically.
* Volumes restored from encrypted snapshots are encrypted automatically.
* You can share snapshots but only if they are unencrypted. Can be shared with other AWS accounts or made public.

<https://aws.amazon.com/articles/9001172542712674>

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/building-shared-amis.html>

Lab 8 AMI Types:

Select by OS, Region, Architecture, Launch Permissions, storage for root devices

* **Instance Store (Ephemeral Storage)**
  + Can’t stop the instance. Can only reboot or terminate.
  + Get less durability if you use this than EBS.
  + Doesn’t start on a new host. VM is lost/done.
  + The root device for an instance launched from the AMI is an instance store volume created from a template stored in S3.
  + Takes more time to deploy rather than EBS.
  + Can reboot and you won’t lose the data.
* **EBS Backed Volumes**
  + Can stop the instance.
  + Root device for an instance launched from the AMI is an EBS volume created from an EBS snapshot
  + Start it on a new host if health check of hypervisor is degraded or failed. Won’t lose data if it is stopped.
  + Cant reboot and you won’t lose the data.

By default, both root volumes will be deleted on termination, however with EBS volumes, you can tell AWS to keep the root device volume.

Found under Community AMI’s

Lab 9 ELB’s:

Application load balancer are layer 7 appliances.

Classic ELB is a layer 4 level.

**Response timeout** – time to wait when receiving a response from the health check

**Interval** – time between health checks

**Unhealthy threshold** – number of consecutive health check failures before EC2 instance is deemed unhealthy.

**Healthy threshold** – number of consecutive health checks it has to pass before it becomes health again.

Load balancer won’t send traffic if the EC2 instance is out of service. Amazon manages the public IP address of the ELB.

Amazon wants you to use the DNS name instead of the public name.

1 subnet = 1 AZ.

Application load balancer uses target groups. Also has success codes. **Read FAQ of ELB. Know high level of ALB.**

* Instances monitored by ELB are reported as InService or Out of Service
* Health checks check the instance health by talking to it.

<http://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html>

<http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-backend-instances.html>

Lab 10 Cloud Watch EC2:

Basic monitoring monitors every 5 minutes. Advanced monitoring monitors every 2 minutes.

**CPU, disk, network, and status check (host or instance) are all monitors available by default**

RAM requires a custom metric.

Consecutive monitoring = basic monitoring. Detailed monitoring is every 1 minute.

**Events** – help you to respond to state changes with your EC2 instance.

* Uses lambda function, etc.
* Connects blocks of AWS services together.

**Logs** – monitor EC2 instances at application level and into kernel logs.

* Requires agent to be installed on the EC2 instance.

**Dashboards** – create awesome dashboards to see what is happening with your environment

**Alarms** – allow you to set alarms that notify you when particular thresholds are hit.

Cloud Trail = auditing in AWS account.

Cloud Watch = performance logging/monitoring.

Lab 11 AWS CLI:

AWS CLI installation documents. **Invoke command line by aws and then the service and command**

You first need to configure AWS by **aws config** command. Need access id and secret id with the region you work off of.

**Aws s3 help** to know what each command does.

You see where the commands are being listed.

**Cd ~, cd .aws.**

Stores AWS commands locally on the machine.

**Aws ec2 terminate-instances –instance-ids id** – terminates the instance.

<https://aws.amazon.com/blogs/security/new-attach-an-aws-iam-role-to-an-existing-amazon-ec2-instance-by-using-the-aws-cli/>

<https://aws.amazon.com/blogs/security/easily-replace-or-attach-an-iam-role-to-an-existing-ec2-instance-by-using-the-ec2-console/>

Lab 12 IAM with EC2:

Roles are created globally. You can attach a role to an EC2 instance. Don’t have to store your credentials locally with using roles instead of just users.

Lab 13 Bash Scripting:

Shabang and then the path of the interpreter.

#!/bin/bash

yum update –y

yum install httpd –y

service httpd start

chkconfig httpd on

aws s3 cp s3://bucketname/index.html /var/www/html \*--region

Lab 14 Instance Metadata:

**Curl** [**http://169.254.169.254/latest/meta-data/**](http://169.254.169.254/latest/meta-data/)**.** After metadata not user data.

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html>

Lab 15 Autoscaling:

You can put bootstrap scripts in the launch configuration start. Autoscale within a specific AZ or all.

Health Check Grace Period - waits for a period of time when an instance comes on and then checks for health status checks. More of a horizontal scaling than vertical.

EC2 Placement Groups:

**Placement group** – logical grouping of instances within a AZ. Enables applications to participate in a low latency, 10 Gbps network. Benefit from low network latency, high network throughput, or both.

* Can’t span placement groups across multiple AZ’s.
* Name must be unique in AWS account.
* Certain types of instanced can only be launched.
* AWS recommends homogenous(same size & family) instances with placement groups.
* Can’t merge placement groups
* Can’t move an existing instance into a placement group. Can launch Ami’s from existence instances into the placement group.

<http://docs.aws.amazon.com/cli/latest/reference/s3/>

Lab 16 EFS Lab:

**Elastic File System (EFS)** – file storage service for EC2 instances. EFS is easy to use and provides a simple interface that allows you to create and configure file systems quickly and easily. EFS capacity is elastic, growing and shrinking automatically as you add/remove files so your applications have the storage they need.

* NFSv4 protocol. Only pay for what you use.
* Can scale up to PB
* Support up to concurrent NFS connections
* Data is stored across multiple AZ’s within a region.
* Block based storage. Can be shared with other EC2 instances
* Read after write consistency.
* Can restrict files at directory level and root level.

Note: Make sure EC2 instances are part of the same SG of the EFS

Run the command to mount it to the EC2 instance but change it to mount to a different directory. Has to be done in the Server. Don’t need multiple copies of your code. **Remember the use case for this. Using it as a file server like a central repo.**

Lambda:

PaaS – Elasticbean stalk

**Lambda** – all you have to worry about is your code. Event triggers. Customer doesn’t have to worry about managing anything except for the code.

* Data Centres, hardware, assembly code, high level languages, OS, application layer encapsulated in all.
* **Lambda** – compute service where you can upload your code and create a lambda function. Takes care of provisioning and managing the servers that you use to run the code. Don’t have to worry about OS.
  + event-drive compute service where AWS lambda runs code in response to your events. Changes to AWS S3 or DynamoDB
  + Run your code in response to HTTP requests with API Gateway or API calls with AWS SDK’s.
  + Node.js, Python, Java, C#
  + Continuously scales
  + Price by # of requests. 1 million requests are free. $.20 per 1 million requests thereafter
    - Duration – calculated from the time your code begins executing until it returns or otherwise terminated. Price depends on amount of memory you allocate to your function.

**EC2 Summary:**

**Know difference between the following:**

* **On demand**
* **Spot**
  + **If you term the instance, you pay for the hour.**
  + **If AWS terminates the spot instance, you get the hour it was terminated in for free**
* **Reserved**
* **Dedicated hosts**

**Know Instance Types – Dr Mc GIFT PX**

**EBS Consists of:**

* **SSD, General Purpose – GP2 (Up to 10,000 IOPS)**
* **SSD, Provisioned IOPS – IO1 (More than 10,000 IOPS)**
* **HDD, Throughput Optimized – ST1 – frequently accessed workloads**
* **HDD, Cold – SC1 – less frequently accessed data.**
* **HDD, Magnetic – Standard – cheap, IA storage.**
* **Can’t mount 1 EBS volume to multiple EC2 instances instead just one**
* **Termination Protection is turned off by default, you must turn it on.**
* **EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated**
* **Root volume can’t be encrypted by default, you need third party tools to encrypt the root volume.**

**Volumes exist on EBS**

* **VHD**

**Snapshot exist on S3**

* **You can take a snapshot of a volume, this will store that volume on S3.**
* **Point in time copies of volumes**
* **Incremental, this means that only the blocks that have changed since your last snapshot are moved to S3.**
* **May take time to create if it is your first snapshot.**
* **Snapshots of encrypted volumes are encrypted automatically**
* **Volumes restored from encrypted snapshots are encrypted automatically**
* **You can share snapshots only if they are unencrypted.**
  + **Snapshots can be shared with other AWS accounts or make them public**
* **Stop the instance to take a snapshot**

**EBS vs Instance Store**

* **Instance Store volume are sometimes called Ephemeral Storage**
  + **Cant be stopped, if the host fails, data is lost**
  + **You can reboot but wont loose your data**
* **EBS backed instances can be stopped. Won’t lose data on the instance if it is stopped.**
  + **You can reboot but wont loose your data**
  + **You can tell AWS to keep the root device rather than deleting on termination.**

**Snapshot with RAID array**

**Problem – snapshot excludes data held by cache by the application and the OS. Multiple volumes have issues with this due to interdependencies**

**Solution – take application consistent snapshot**

* **Stop application writing to disk**
* **Flush all caches to the disk**
* **Freeze the file system**
* **Unmount the RAID array**
* **Shut down the associated EC2 instance.**

**AMI’s are regional – you can copy AMI’s to other regions by CLI, API or console.**

**Standard Monitoring = 5 minutes**

**Detailed Monitoring = 1 Minute**

**CloudWatch – performance monitoring**

* **Create dashboards, alarms, events, logs**

**CloudTrail – auditing**

**Roles are more secure than storing access key and secret access key on individual EC2 instance**

* **Roles are easier to manage.**
* **Roles can be assigned when an EC2 instance is being provisioned.**
* **Roles are universal, you can use them in any region**

**Instance meta-data:**

* **Used to get information about an instance**
* **Curl** [**http://169.254.169.254/latest/meta-data/**](http://169.254.169.254/latest/meta-data/)

**EFS**

* **Pay only for what you use**
* **NFSv4 support**
* **Data stored across Multiple AZ**
* **Read after Write consistency**
* **Support thousands of concurrent NFS**
* **Scale up by PB.**

**Lambda**

* **Compute service to upload code and create a lambda function**
* **Event driven compute service.**

[**https://aws.amazon.com/blogs/aws/new-encrypted-ebs-boot-volumes/**](https://aws.amazon.com/blogs/aws/new-encrypted-ebs-boot-volumes/)

**Route 53**

**DNS 101**

DNS is used to convert human friendly domain names into an IP address like 82.124.53.1

IP addresses are used by computers to identify each other on the network. IP is a 32bit field with over 4 billion different addresses.

IPv6 was used to solve this depletion issue and have an address space of 128 bits or 340 undecillion addresses. VPC can used IPv6.

**Top level domains** like .com, .edu, .gov. Second level domain is .co.uk, .gov.uk. IANA (Internet Assigned Numbers) control top level domains.

**Domain Registrars** – an authority that can assign domain names directly under one or more top-level domains. Route 53 doesn’t fall under free tier. Registered in a central database as whois database.

**SOA record** – name of the server that supplied the data for the zone

* Administrator of the zone
* Current version of the data file
* Number of seconds a secondary name server should wait before retying a failed zone transfer
* Number of seconds that a secondary name server can use data before it must either be refreshed or expire.
* Default number of seconds for a time-to-live file on resource needs.

**NS (Name Server)** – used by top level domain servers to direct traffic to the Content DNS server which contains the authoritative DNS records.

* Supply these to the hosting provider.

A record – Stands for address – used by computer to translate the name of the domain to IP address.

**TTL** – time to live. Length of time the DNS record is cached on either the resolving server or the users own local PC equal to the value of TTL. Lower the TTL, faster changes to DNS records to propagate. Plan two days before hand to migrate.

**Cname (Canonical Name)** – resolve one domain name to another. Mobile website to main domain name. Can’t be used for naked domain names. **Naked name** – name without www.

**Alias records** – used to map resource record sets in your hosted zone to ELB, S3 buckets. Can map one DNS name to another target DNS name.

**ELB don’t have pre-defined IPv4 but only resolve to DNS name.**

**Understand the difference between Alias record or CNAME. Always choose Alias record over CNAME.**

Lab 1 Register a Domain Name:

You can register a domain name through AWS. Registration could take up to 3 days to be completed. Comes with associated NS/SOA records. Comes with different top level domain names.

Lab 2 Setup our EC2 instances Lab:

2 EC2 with 1 ELB in one region. 1 EC2 instance with 1 ELB in another region.

#!/bin/bash

yum update -y

yum install httpd24 -y

service httpd start

chkconfig httpd on

echo "<html><body><h1>Hello Cloud Gurus, this is X</h1></body></html>" > /var/www/html/index.html

Lab 3 Simple Routing Policy:

**Simple** – default routing policy when you create a new record set. No failover, etc. Round Robin.

**Naked domain** – alias record.

Lab 4 Weighted Routing Policy:

**Weighted** – let you split your traffic based on different weights assigned. Weights it over the length of the day.

Lab 5 Latency Routing Policy:

**Latency** – allows you to route your traffic based on the lowest network latency for your end user. Selects the region based on the lowest latency. VyprVPN.

Lab 6 Failover Routing Policy:

**Failover** – used for when you want to create an active/passive setup. Route 53 will monitor the primary site with a health check. Healthchecks can be done by domain names or paths.

Lab 7 Geolocation Routing Policy:

**Geolocation** – let’s you choose where your traffic will be sent based on the geographic location of your users. Default – means everywhere else.

**Summary:**

**ELB don’t have public IP’s. AWS takes care of that. Found by domain name.**

**Know Alias and CNAME record. Always choose alias record over CNAME.**

**Know Routing policies. Read FAQ on Route 53.**

**Databases 101**

**Relational database** – based on columns, databases, rows, fields, records. Each field contains the same type of information. Different types like SQL, Oracle, MySQL, PostgreSQL, Aurora, MariaDB. **Read the FAQ of databases.**

**Non Relational Databases –** CouchDB, MongoDB. Document oriented databases.

* Database
  + Collection = Table
  + Document = Row
  + Key Value Pairs = Fields
* JSON based with {}
* DynamoDB – No SQL

**Data warehousing** – used with business intelligence. Used to pull in very large and compelx data sets. Usually used by management to do queries. SQL, SAP, Jaspersoft,etc

* OLTP (Online Transaction Processing) - RDS
* OLAP (Online Analytics Processing) - Redshift
* Uses different database and infrastructure architecture as part of data warehousing

**ElastiCache** – web service that makes it easy to deploy, operate, and scale an in-memory cache in the cloud. Service improves the performance of web applications by allowing you to retrieve information from fast, managed, in-memory cahces, instead of relying entirely on slower disk-based databases.

* Emcached
* Redis

**DMS (Database Migration Service)** – allows you to migrate your production database to AWS. Once the migration has started, AWS manages all the complexities of the migration process like data type transformation, compression, and parallel transfer while ensuring that data changes to the source database that occur during the migration process are automatically replicated to the target.

* Schema conversion tool – converts the source database schema and majority of the custom code, including views, stored procedures, and functions, to a format compatible with the target database.

Lab 1 Create First RDS instance:

**Know port 3306 for MySQL. Always make sure the inbound and outbound firewalls allow the RDS instance to come in via that port.**

#!/bin/bash

yum install httpd php php-mysql -y

yum update -y

chkconfig httpd on

service httpd start

echo "<?php phpinfo();?>" > /var/www/html/index.php

cd /var/www/html

wget <https://s3.eu-west-2.amazonaws.com/acloudguru-example/connect.php>

RDS – Backups, Multi-AZ & Read Replicas:

**Automated Backups** – allow you to recover your database to any point in time within a “retention period”.

* Retention period is 1 to 35 days.
* Takes a full daily snapshot and stores transaction logs throughout the day.
* AWS chooses the most recent backup. Allows Point in time recovery.
* Are enabled by default. Stored in S3
* RDS instance is the size of the 10GB will be 10GB of storage.
* Storage I/O may be suspended while data is being backed up

**DB Snapshots** – manually done. Stored even after the instance is deleted.

* Restored version of the DB will be a new RDS instance.

**Encryption at rest** – MySQL, Oracle, SQL Server & PostgreSQL & MariaDB. Done using KMS

* Can’t encrypt an existing DB instance. Have to build a new to encrypt.

**Multi-AZ** – synchronous replication to another AZ. Allows exact copy of your production db in another AZ. AWS handles the replication. RDS will automatically failover to the standby. Multi-AZ for DR only.

* For performance increase, do read replicas.
* SQL Server, Oracle, MySQL, MariaDB, PostgreSQL.

**Read Replica** – asynchronous replication which can be read from. You can have read replicas of read replicas with up to 5 of them (latency possible). Used for read-heavy database workloads.

* MySQL, PostgreSQL, MariaDB
* Used for scaling
* Automatic backups must be turned on to use read replica.
* Each read replica will have its own DNS end point.
* Can’t have read replicas that have Multi-AZ.
* Can crate read replica’s of Multi-AZ source.
* Read replica can be promoted to its own db. Breaks replication
* Can have read replica in a second region with MySQL or MariaDB

**DynamoDB** – offers push button scaling. Can scale database on the fly. RDS has to use bigger instances for scaling.

* NoSQL database for all applications needing single digit latency
* Fully managed and supports document and key-value data models
* Stored on SSD storage
* Spread across 3 geographically DC’s.
* **Eventual Consistent Reads**
  + Consistency across all copies of data is within a second. Repeating a read after a short time should return the updated data (best read performance)
* **Strongly Consistent Reads**
  + Returns a result that reflects all writes that received a successful response prior to the read. < 1 second needed
* Pricing:
  + Provision throughput capacity by write and read
  + Storage costs
  + Read units in 50
  + Write unit in 10s

**Know Multi-AZ & Read replica.**

**Redshift** – data warehousing service fully managed and PB scale. OLAP type transaction.

* Single node (160GB)
* Multi-Node
  + Leader Node (manages client connections and receives queries)
  + Compute Node (store data and perform queries and computations) up to 128 compute nodes.
* **Columnar Data Storage** – column based systems for data warehousing and analytics aggregate. Stores data sequentially on storage media so it requires less I/O.
* **Advanced Compression** – can be compressed much better than row data due to it being sequeinltly stored. Doesn’t require indexes or views and uses less space. Selects the appropriate schema when loaded.
* **Massively Parallel Processing (MPP)** – automatically distributes data and query across all nodes. Makes it easy to add nodes to maintain fast query power.
* Price
  + Compute Node Hours – number of hours across
  + Backup
  + Data Transfer (only in a VPC charged for not outside)
* Security
  + Encrypted in transit
  + Encrypted at rest in AES-256
  + Manage keys through HSM or KMS
* Availability
  + Only available in 1 AZ.
  + Can restore new snapshots to new AZ’s.

**Elasticache** – web service makes it easy to deploy and operate scale in-memory cache in the cloud. Improves performance by retrieving information from cache than disk-based database. OLAP type transactions.

* Improves latency and throughput with **read-heavy application workloads**
* Cached information results in low I/O
* Memcached
  + Object caching system
  + Doesn’t support multi-AZ.
* Redis
  + Open-source key value store with sorted sets and lists.
* Supports Master/Slav with Multi-AZ capability.

<https://aws.amazon.com/elasticache/>

**Aurora** – RDS with the combined speed of availability with high-end commercial database. Better performance than Mysql with less than the price point. Open-Source.

* Scales by 10Gb increments up to 64 TB and autoscales.
* Compute resources can scale up to 32vCPU’s and 244GB
* 2 copies of your data in each AZ with a minimum of 3 AZ with 6 copies of data.
* Self-healing. Disks are scanned for errors.
* Aurora replicas – 15 – failover is the main difference between the two.
* MySQL read replicas – 5

Summary

**VPC Overview**

**VPC –** logical DC. VPC don’t span regions but span AZ. Lets you provision a logically isolated section of AWS where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including own IP address range, creation of subnets, and configuration of route tables and network gateways.

* Can create public facing subnet and private facing subnet.
  + Private:
    - 10.0.0.0 -10.255.255.255
    - 172.16.0.0 – 172.31.255.255
    - 192.168.0.0 – 192.168.255.255
* Can create VPN (hardware) between your corporate DC with your VPC.
* Only allowed /16 size networks as max.
* S2S goes through Virtual Private gateway
* Network ACL and Security groups can span subnets.
* **Each subnet is mapped to an AZ.** Can’t span subnets across multiple AZ.
* **Only 1 internet gateway per 1 VPC.**
* **Security groups are stateful. ACL’s are stateless.**

**Default VPC** – all subnets have a route out to the internet. All public.

* EC2 has private/public ip address
* If you delete it, you have to contact AWS.

Custom VPC

**VPC Peering** – allow you to have one VPC with another via direct network with private IP addresses.

* Can peer to other AWS accounts.
* Star configuration. **No transitive peering** (no middle men)

Lab 1 Build your own custom VPC:

Doesn’t automatically create subnet. Automatically creates security groups, route, ACL.

AWS reserve 3 IP addresses per subnet.

* 10.0.0.1 - router
* 10.0.0.2 - dns
* 10.0.0.0 – network
* 10.0.0.3 – future addresses

New subnet’s are associated to your main route table.

#!/bin/bash

yum install httpd -y

yum update -y

service httpd start

chkconfig httpd on

echo "<html><h1>Hello Cloud Gurus!</h1></html>" > /var/www/html/index.html

Lab 2 Build your own custom VPC:

Internal server with different security group.

Lab 3 Nat Gateway & Nat instance:

Nat instances must be behind a security group. EC2 instances must be a source/destination.

NAT gateway is always deployed on public security groups.

Disable source destination check on the NAT instance. NAT Gateways are managed by AWS.

NAT Gateways used in production.

NAT Instances

* disable source/destination check on instance.
* must be in a public subnet.
* Must be a route out of the private subnet to the NAT instance.
* Amount of traffic supported depends on the size of the instance
* HA using autoscaling groups, multiple subnets in different AZ.
* Behind a security group

NAT Gateways

* Scale automatically up to 10 Gbps
* No need to patch
* Automatically assigned a public IP address
* Update your route tables.
* Not associated with security groups
* No need to disable source/destination checks.

Network ACL’s vs Security Groups:

**Security Group**

* First layer of defense on the instance
* Supports allow rules only
* **Stateful** – return traffic is automatically allowed
* All rules are evaluated before deciding on traffic.
* Instances must be placed within the security group

**Network ACL**

* Second layer of defense on subnet layer
* Supports allow and deny rules
* **Stateless** – return traffic must be explicitly allowed (have to make the change on ingress/egress).
* Rules are processed in order starts with lowest numbered rule.
* Applies to all instances within the subnet.
* 1 subnet can only apply to 1 ACL.
* **Default** – allows inbound/outbound.
* **Custom ACL** doesn’t allow inbound/outbound automatically
* Separate inbound and outbound rules.
* Block IP’s using network ACL not SG’s.

Each subnet must be associated with ACL. If it is not, it will be associated with your default network ACL with that subnet. Previous associations are removed if you repurpose the ACL.

Do rule numbers in increments of 100’s.

1024-65535 – ephemeral ports

**Achieved 77% on Diagnostic test on WhizLabs**

Lab 3 Custom VPC’s & ELB:

Always want 2 public subnets for HA. Possibly 2 private subnets.

NAT vs Bastion

**Bastion (Jump Box)** – initiate a private connection over to the private instance in the private Subnet.

NAT gateway is not behind a SG. Bastion is behind a SG. NAT would require a script to handle the failover.

**NAT instance** – used to provide internet traffic to EC2 instances to private subnets

**Bastion** – used to security administer EC2 instances in private subnets.

**VPC Flow Logs** – enable to capture IP traffic flow for the network interfaces. Reports it to cloudwatch.

VPC Clean Up

**Summary**

* **VPC Transitive Peering means VPC can only talk 1 -1 not through a middle man.**
* **NAT instances must have an elastic IP to work**
  + **Must be a route out of the private subnet to the NAT instance to work**
  + **Behind a SG**
* **HA can be achieved by**
  + **Autoscaling**
  + **Multiple subnets in different AZ**
  + **Script to automate failover**
* **NAT Gateways**
  + **No need to patch**
  + **Scale up to 10Gbps**
  + **Not associated to SG**
  + **Update your route tables**
  + **Automatically assign a public IP**
* **Network ACL**
  + **You can associate a network ACL with multiple subnets however subnet can be associated with only 1 ACL**
  + **Rules start by lowest numbered rule**
  + **Block IP via ACL**

**Resiliency can be achieved by 2 public subnets and 2 private subnets. EBLS should be in 2 public subnets in 2 AZ. Bastion hosts, put them behind an autoscale group with a minimum of 2. Route 53 to automatically fail over. NAT instances need to be in each public subnet with a script to failover.**

**Application Services**

**Simple Queue Service** – first AWS service available. Web service that gives you access to a message queue that can be used to store messages while waiting for a computer to process them.

* Distributed queue system that enables web service applications to quickly and reliably queue messages that one component in the application generates to be consumed by another component. Queue is a temporary repository for messages that are awaiting processing.
* A way of assigning jobs
* Pull based system
* SNS is a push based system.
* Removes dependencies and decouple the components of an application. Any component of distributed application can store messages in a fail-safe. Messages can save up to 256 kb in any text format.
* **Queue** acts as a buffer between the component producing and saving data, and the component receiving the data for processing. Queue resolves issues that arise if the producer is producing work faster than the consumer can process it.
* Default is 4 days kept in the queue. Can be kept in the queue from 1 minute to 14 days.
* **Visibility Time Out** – amount of time that the message is invisible in the SQS queue after a reader picks up that message. Provided the job is processed before the visibility time out expires, the message will then be deleted from the queue. If not processed, it will be visible again. Message could be delivered math. Maximum timeout is 12 hours.
* **Long polling** – a way to retrieve messages from SQS queues. Doesn’t return a response until a message arrives in the message queue. Short polling returns immediately.
* Message oriented API

**Standard Queues (default)**

* Nearly-unlimited transactions per second. Delivered at least once. One copy of a message might be delivered out of order. Doesn’t guarantee every message.

**FIFO Queues**

* Delivered in order in which they are received. Duplicates aren’t introduced. Supports message groups
* Limit to 300 transactions per second.

<http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-how-it-works.html>

**Simple Workflow Service –** web service makes it easy to coordinate work across distributed application components.

* Tasks represent invocations of various processing steps in an application which can be performed by executable code, web service calls, human actions, and scripts
* Enables applications to be designed as a coordination of tasks.
* Presents a task-oriented API
* Task is assigned only once and isn’t duplicated
* SWF sits up to 1 year for workflow executions.
* **Workflow starters** – an application that can initiate a workflow
* **Deciders** – control the flow of activity tasks in a workflow execution. If finished, decider decides what to do next.
* **Activity workers** – carry out the activity tasks

**Simple Notification Service –** web service that can send notifications from the cloud. Can public messages from an application and deliver them to subscribers or other applications.

* Apple, google, windows, Fire OS
* Can send SMS text message or email, SQS or HTTP/HTTPS endpoint.
* Can trigger lambda functions, application, email-JSON
* Push messaging system.
* Can group recipients using topics
* **Topics** – an access point for allowing recipients to dynamically subscribe for identical copies. Can support multiple endpoints.
* stored redundantly across multiple AZ.
* Used the most around autoscale.
* $.50 per 1 million SNS requests
  + $.06 per 100,000 over HTTP
  + $.75 per 100 over SMS
  + $2.00 per 100,000 over Email

**Elastic Transcoder** – media transcoder in the cloud. Convert medial files that will play on smartphones, tablets, PC.

* Pay based on minutes used.

**API Gateway** – fully managed service that makes it easier to publish, maintain, monitor, and secure API’s at any scale for developers.

* Can create a front door for applications to access x running on EC2 for example.
* Can be enabled by API caching – reduce number of calls made to the endpoint and improve latency.
* Can throttle requests to prevent attacks. Can connect to CloudWatch
* **Same origin policy** – web browser permits scripts contained in a first web page to access data in a second web page but only if there in the same origin.
* **CORS (Cross-Origin Resourcing Sharing)** mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served. Make sure it is enabled if your using multiple domains.

**Kinesis 101 –** send your streaming data too**.** Makes it easy to load and analyze streaming data and provide the ability for custom applications.

**Streaming data** – is data that is generated continuously by thousands of data sources which typically send in data records.

* Example: game data, stock prices, purchases online, social network data. Geospatial data, iOT sensor data.

**Kinesis Streams** – stores data 24 hours and can increase the retention 7 days. Stored in shards. EC2 instance takes the shards and does something nice with them.

* Shards gives 5 transactions per second with up to 2 MB per second with 1000 records per second and 1 MB per second writes
* Can have multiple shards in a stream.

**Kinesis Firehose** – automated. Don’t have to worry about shards, etc. No data retention periods. No management required. Doesn’t have shards.

**Kinesis Analytics** – can run sql queries on streams/firehose, etc.

Lab 1 Kinesis:

Shard level metrics are produced every 1 minute instead of 5 minutes.

Summary:

* SQS is the most important.
* Short polling returns immediately
* Long polling retrieves messages or until long polling times out.
* SWF keeps track of all the events in an application. SQS needs application level tracking implemented.
* Charged on Elastic transcorder based on the resolution as well.
* Kinesis – consumers (EC2) – producers (users,laptops).
  + Firehose=lambda

**Creating a fault tolerant Word Press Site**

Lab 1 Getting Setup:

Draw.io . 2 SG, 2 RDS, 2 S3, Cloudfront, 1 ELB, Route 53

Lab 2 Setting Up EC2:

#!/bin/bash

yum update -y

yum install httpd php php-mysql stress -y

cd /etc/httpd/conf

cp httpd.conf httpdconfbackup.conf

nano httpd.conf – Allowoverride control ALL

rm -rf httpd.conf

wget https://s3-eu-west-1.amazonaws.com/acloudguru-wp/httpd.conf

cd /var/www/html

echo "healthy" > healthy.html

wget https://wordpress.org/latest.tar.gz

tar -xzf latest.tar.gz

cp -r wordpress/\* /var/www/html/

rm -rf wordpress

rm -rf latest.tar.gz

chmod -R 755 wp-content

chown -R apache.apache wp-content

service httpd start

chkconfig httpd on

aws s3 ls

aws s3 cp –recursive /var/www/html s3://wordpresscode16acloudguru

aws s3 ls wordpresscode16acloudguru

aws s3 cp –recursive s3://wordpresscode16acloudguru /var/www/html

chmod –R 755 wp-content

chown –R apache.apache wp-content

Lab 3 – Automation & Setting Up Our AMI

Cd wp-content/uploads

Aws s3 ls

Aws s3 cp –recursive /var/www/html/wp-content/uploads/ s3://wordpressmedia16acloudguru

Aws s3 sync /var/www/html/wp-content/uploads/ s3://wordpressmedia16acloudguru –delete(perfect sync) –dryrun

Wget <https://s3-eu-west-1.amazonaws.com/acloudguru/config/htaccess>

Cp htaccess .htaccess

Rm –rf htaccess

Cd /etc

Nano crontab (scheduled task)

\*/2\*\*\*\*root aws s3 sync –delete /var/www/html/ s3://wordpressmedia16acloudguru/

\*/2\*\*\*\*root aws s3 sync –delete /var/www/html/wp-content/uploads/ s3://wordpressmedia16acloudguru/

\*/3\*\*\*\*root aws s3 sync –delete s3://wordpresscode16acloudguru/ /var/www/html/

Best practice to not select no reboot when creating an image.

When you stop an EC2 instance, it removes the public IP. When you start it up, the IP will change unless you do an elastic IP.

Lab 4 Autoscaling & Load Testing:

Best practices are to also select all the subnets. Stress –y is the stress load testing scenario.

Lab 5 CloudFormation:

Super Handy!

**Preparing for the EXAM!**

Overview of AWS architecture review:

<https://d0.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf>

Cloud Computing definition

Advantages of cloud

* Trade capital expense for variable expense
* Benefit from massive economies of scale
* Stop guessing about capacity
* Increase speed and agility
* Go global in minutes
* Stop spending money running and maintaining data centers.

5 Regions with most region having 3 AZ’s.

Security:

* 24x7 security guards
* Least privilege basis
* State of the art electronic surveillance

Compliance:

* **SOC1/SSAE 16/ISAE3402 (SAS 70 Type II)**
* **SOC2**
* **SOC3**
* FISMA, FedRAMP, DIACAP
* **PCI DSS Level 1 (QSA required) IaaS only.**
* **ISO 27001**
* **ISO 9001**
* ITAR
* FIPS 140-2
* HIPAA
* CSA
* MPAA

Overview of Security Processes (Part 1):

<https://d0.awsstatic.com/whitepapers/aws-security-whitepaper.pdf>

AWS is responsible for securing the underlying infrastructure. We are responsible for anything we put on the cloud or connect anything to the cloud. (EC2, VPC, S3)

AWS is responsible for services like DynamoDB, RDS, Redshift, EMR, Workspaces which are managed services. We are responsible for account management and user access with managed services. MFA should be implemented if API or SSL/TLS is setup.

Magnetic storage is degaussed and physically destroyed.

**Transmission Protection** – can connect to an AWS access point via HTTP with SSL protects against eavesdropping

**Amazon Corporate Segregation** – AWS production network is segregated from the Amazon Corporate network by means of a complex set of network security /segregation devices.

DDoS

MITM

**IP spoofing** – AWS controlled, host-based firewall infrastructure will not permit an instance to send traffic other than its own.

**Port scanning** – unauthorized vunerability scans by EC2 customers are violating the AUP. You can ask for permission but have to be **requested in advance** and limited to your stuff.

Packet sniffing

Credentials:

* Passwords, MFA, Access Keys, Key Pairs, X.509 Certificates (can only be viewed by the person who the link was sent too).

**Trusted Advisor** – inspects your AWS environment and makes recommendations when opportunities may exist to save money, improve system performance, or close security gaps.

Overview of AWS Security (Part 2):

AWS is a Xen hypervisor. Firewall resides within the hypervisor layer between physical network interface and instance virtual interface. All packets go through this layer. Neighbors have no more access to an instance. Same for physical ram.

Disk virtualization layer is resetting and disk zeroed so customers don’t have access to raw disks. Memory is scrubbed.

**Guest OS** – instances are controlled by the customer. AWS doesn’t have any access to instance or guest OS.

* Encryption occurs on the host. Encryption is only available on M3, C3, R3, G2 powerful EC2 instances. AES-256

**Firewall** – inbound firewall is default deny-all mode. EC2 customers must open the ports needed.

**ELB** – SSL termination is supported.

* Can identify originating IP address of a client connecting to your servers.

**Direct Connect** – bypasses ISP in your network path. 802.1q VLAN’s. extend iP address office inside your VPC. Can access other resources like S3 or VPC.

Overview of Risk & Compliance:

<https://d0.awsstatic.com/whitepapers/compliance/AWS_Risk_and_Compliance_Whitepaper.pdf>

Customer responsible for patching and guest OS for IaaS. AWS evaluates the business risk plan biannually. AWS scans all internet facing endpoint IP for vulnerabilities which is done on their stuff. Assessments are done by security firms.

Overview of Storage Options:

<https://d0.awsstatic.com/whitepapers/AWS%20Storage%20Services%20Whitepaper-v9.pdf>

AWS Import/Export often faster than internet transfer or upgrading connectivity.

* Per device fee
* Data load time charge
* Shipping charges

Storage Gateway

* Gateway-cached volumes use S3. Can create volumes up to 32 TB of storage.
* Gateway stored volume creates storage up to 1 TB. Stored onpremise and then asynchronously backed to S3.
* Gateway usage
* Snapshot storage usage
* Volume storage usage
* Data transfer

Overview of Architecting for the Cloud Best Practices:

<https://d0.awsstatic.com/whitepapers/AWS_Cloud_Best_Practices.pdf>

<https://d0.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf>

Benefits:

* JIT infrastructure
* Usage cost
* Efficient resource utilization
* Zero upfront infrastructure investment
* Reduced time to market.

Technical benefits:

* Automation – “scriptable infrastructure”
* Autoscaling
* Proactive scaling
* Efficient development lifecycle
* Improved Testability
* DR & BC
* Overflow traffic to the cloud

Rule of thumb: Be pessimist when designing things in the cloud. Assume hardware will fail. This helps designing an overall system better. Best to have loose coupling.

Elasticity:

* Proactive cyclic scaling – periodic scaling that occurs at fixed interval
* Proactive event-based scaling – scale when you expect a big surge
* Auto-scaling based on demand – monitoring service to go up/down.

Exam Practicalities:

55 questions, $150, 80 minutes in length. [www.webassessor.com](http://www.webassessor.com)

<http://media.amazonwebservices.com/AWS_Cloud_Best_Practices.pdf>

2 forms of ID. Reschedule by emailing [awscertification@amazon.com](mailto:awscertification@amazon.com). Reschedule more than 72 hours.

**Practice Test 1 – 58%.**

**Well Architected Framework**

<https://d0.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf>

Set of questions you can use to evaluate the architecture framework.

Pillars:

* **Security**
  + Design Principles
    - Apply security at all layers
    - Enable traceability
    - Automate responses to security events
    - Focus on securing your system
    - Automate security best practices (Center for internet security website)
    - Shared responsibility model
  + Definition
    - Data protection
    - Privilege management
    - Infrastructure protection
    - Detective controls
  + Best Practices
    - Data protection
      * Organize and classify your data in segments, Classify the data. Least privilege access system. Encrypt where possible.
      * Customers maintain full control over their data.
      * AWS makes it easy for encrypting data with key rotation.
      * Detailed logging is available
      * Designed storage systems for exceptional resiliency
      * Versioning can protect from accidental overwrites, deletes, similar harm
      * AWS never initiates the movement of data between regions. Data will remain in that region.
      * How are you encrypting and protecting your data at rest?
      * How are you encrypting and protecting your data in transit (SSL)?
    - Privilege management
      * ACL, RBAC, Password management
      * How are you protecting access to and use of the AWS root account credentials?
      * How are you defining roles and responsibilities of system users to control human access to the AWS management console and API’s?
      * How are you limiting automated access to AWS resources?
      * How are you managing keys and credentials?
    - Infrastructure protection
      * RFID controls, lockable cabinets, security guards, VPC
      * How are you enforcing network and host-level boundary protection?
      * How are you enforcing AWS service level protection?
      * How are you protecting the integrity of the operating systems on your AWS EC2 instances?
    - Detective controls
      * CloudTrail, CloudWatch, Config, S3, Glacier
      * How are you capturing and analyzing AWS logs?
  + Key Services
    - Data protection
      * ELB, EBS, S3 & RDS
    - Privilege management
      * IAM, MFA
    - Infrastructure protection
      * VPC
    - Detective controls
      * CloudTrail, CloudWatch, Config,
  + Exam Tips
    - 4 pillars – look at the questions asked.
* **Reliability**
  + Design Principles
    - Test recovery procedures (Simion army tool)
    - Automatically recover from failure
    - Scale horizontally to increase aggregate system availability
    - Stop guessing capacity
  + Definition
    - Ability to recover from service disrupt or failure
    - Foundations
    - Change management
    - Failure management
  + Best Practices
    - Foundations
      * Have prerequisite foundation in place. Miss provision can cause disruption of IT estate.
      * AWS handles the foundations for you. Service limits are set to over-provisioning resources. (AWS Service Limits)
      * How are you managing AWS service limits for your account?
      * How are you planning your network topology on AWS?
      * Do you have an escalation path to deal with technical issues?
    - Change management
      * Change control is done manually and carefully coordinate with auditing
      * Cloudwatch can help monitor environments for changes.
      * How does your system adapt to changes in demand?
      * How are you monitoring AWS resources?
      * How are you executing change management?
    - Failure management
      * Plan how to responds and prevent failures.
      * How are you backing up your data?
      * How does your system withstand component failures?
      * How are you planning for recovery?
  + Key Services
    - Foundations
      * IAM, VPC
    - Change Management
      * CloudTrail
    - Failure Management
      * CloudFormation
  + Exam Tips
    - 3 areas and look at the questions.
* Cost Optimization
  + Design Principles
    - Transparently attribute expenditure
    - Use managed services to reduce cost of ownership
    - Trade capital expense for operating expense
    - Benefit from economies of scale
    - Stop spending money on DC operations
  + Definition
    - Matched supply and demand
    - Cost-effective resources
    - Expenditure awareness
    - Optimizing over time
  + Best Practices
    - Matched supply and demand
      * Try to match supply with demand. Autoscale can scale with demand or use lambda. Cloudwatch can keep track on demand.
      * How do you make sure your capacity matches but does not substantially exceed what you need?
      * How are you optimizing your use of AWS services?
    - Cost-effective resources
      * Best to use the correct instance type
      * Have you selected the appropriate resource type to meet your cost targets?
      * Have you selected the appropriate pricing model to meet your cost targets?
      * Are there managed services that you can use to improve your ROI?
    - Expenditure awareness
      * Provision things in seconds. Don’t have to get quotes on hardware, etc.
      * What access controls and procedures do you have in place to govern AWS costs?
      * How are you monitoring usage and spending?
      * How do you decommission resources that you no longer need, or stop resources that are temporarily not needed?
      * How do you consider data-transfer charges when designing your architecture?
    - Optimizing over time
      * AWS moves FAST. Subscribe to a blog and use Trusted Advisor.
      * How do you manage and consider the adoption of new services?
  + Key Services
    - Matched supply and demand
      * autoscaling
    - Cost-effective resources
      * EC2 (reserved instances), Trusted Advisor
    - Expenditure awareness
      * CloudWatch alarms, SNS
    - Optimizing over time
      * AWS blog, Trusted Advisor
  + Exam Tips
    - 4 areas and look at the questions.
* Performance Efficiency
  + Design Principles
    - Democratize advanced technologies
    - Go global in minutes
    - Use server-less architectures
    - Experiment more often
  + Definition
    - Compute
    - Storage
    - Database
    - Spacetime trade-off
  + Best Practices
    - Compute
      * Can change a server by the click of a button. Applications require heavy CPU utilization. Jobs start to move to programming or server-less architecture
      * How do you select the appropriate instance type for your system?
      * How do you ensure that you continue to have the most appropriate instance type as new instance types and features are introduced?
      * How do you monitor your instances post launch to ensure they are performing as expected?
      * How do you ensure that the quantity of your instances matches demand?
    - Storage
      * Depends on a number of factors – constrains, frequency, throughput, patterns of access, etc.
      * S3 11 x9’s durability. Choose between different storage mediums.
      * How do you select the appropriate storage solution for your system?
      * How do you ensure that you continue to have the most appropriate storage solution as new storage solutions and features are launched?
      * How do you monitor your storage solution to ensure it is performing as expected?
      * How do you ensure that the capacity and throughput of your storage solutions matches demand?
    - Database
      * Availability, consistency, what do you need. Lots of options
      * How do you select the appropriate database solution for your system?
      * How do you ensure that you continue to have the most appropriate database solution and features as new database solution and features are launched?
      * How do you monitor your databases to ensure performance is as expected?
      * How do you ensure the capacity and throughput of your databases matches demand?
    - Spacetime trade-off
      * RDS read replicas, Direct connect, elasticcache or cloudfront
      * How do you select the appropriate proximity and caching solutions for your system?
      * How do you ensure that you continue to have the most appropriate proximity and caching solutions as new solutions are launched?
      * How do you monitor your proximity and caching solutions to ensure performance is as expected?
      * How do you ensure that the proximity and caching solutions you have matches demand?
  + Key Services
    - * Compute
        + Autoscaling
      * Storage
        + EBS, S3, Glacier
      * Database
        + RDS, DynamoDB, Redshift
      * Spacetime trade-off
        + Cloudfront, Direct Connect, RDS read replicas, Elasticcache.
  + Exam Tips
    - 4 areas and look at the questions.
* Operational Excellence
  + Design Principles
    - Perform operations with code
    - Align operations processes to business objectives
    - Make regular, small, incremental changes
    - Test for responses to unexpected events
    - Learn from operational events and failures
    - Keep operations procedures current
  + Definition
    - Preparation
    - Operation
    - Response
  + Best Practices
    - Preparation
      * Workloads ready for production. Make sure documentation is always up to date.
      * **Runbooks** – operations guidance that operations teams can refer to so they can perform normal daily tasks
      * **Playbooks** – guidance for responding to unexpected operational events.
      * What best practices for cloud operations are you using?
      * How are you doing configuration management for your workloads?
    - Operation
      * Standardized and manageable. Small frequent changes. Shouldn’t be large changes. Logs should be available. Setup continuous integration, etc.
      * Routine operations should be automated. Rollbacks avoided.
      * How are you evolving your workload while minimizing the impact of change?
      * How do you monitor your workload to ensure it is operating as expected?
    - Response
      * Should be automated for mitigation, remediation, rollback. QA should be in place. Escalation paths identified.
      * How do you respond to unplanned operational events?
      * How is escalation managed when responding to unplanned operational events?
  + Key Services
    - Preparation
      * Autoscaling, SQS, AWS Config, AWS Service Catalog
    - Operation
      * Code Commit, CodeDeploy, CodePipeline, SDK’s, CloudTrail
    - Response
      * CloudWatch
  + Exam Tips
    - 4 areas and look at the questions.

Design Principles

* Stop guessing your capacity needs
* Test systems at production scale
* Automate to make architectural experimentation easier
* Allow for evolutionary architectures
* Data-Drive architectures
* Improve through game days

Definition

Best Practices

Key AWS Services

Resources

**Additional Exam Tips**

Kinesis – fully managed service for real-time processing of streaming data at massive scale. You can configure hundreds of thousands of data producers to continuously put data into an Amazon Kinesis Stream.

* Website clickstreams, logs, media feeds
* A way to consume big data.
* Processing large amounts of data
  + Redshift for Business intelligence
  + EMR for Big Data processing

EBS Backed Storage

* Persistent well past the life of the VM
* Can be detached and reattached to other EC2 instances.
* Can be stopped and data persist
* Long term data store

EBS Instance Store

* Not persistent. Life to the end of the VM
* Can’t be detached or reattached
* Can’t be stopped. If you stop, the EBS root data will be lost
* Short term data store

OpsWorks

* Orchestration service that uses Chef. Receipts

Elastic Transcoder

* Media transcoder
* Convert media files into different formats. Pay based on minutes and resolution

SWF Actors need to know

* Workflow Starter
* Deciders
* Activity Workers

Get Public IP address:

* Curl <http://169.254.169.254/latest/meta-data/>
* get <http://169.254.169.254/latest/meta-data/>

**AWS Organizations and Billing**

**Organizations** – account management service that enables you to consolidate multiple AWS accounts into an organization that you create and centrally manage

* all features
* consolidated billing
* Paying account is independent and can’t access resources of the other accounts
  + Can only have 20 linked accounts
    - Can add more and request more.
  + Linked accounts are independent
  + Volume discount pricing
  + Billing alerts enabled on root account can include all linked accounts.
  + Billing alerts can be created an each individual account.
  + Unused Reserved EC2 instances are applied across the group.
* Use strong/complex passwords, MFA, and pay account should only be used for billing.
* CloudTrail is per account and enabled per region
  + Can consolidate logs using S3 bucket
    - Turn on CloudTrail in paying account.
    - Create bucket policy that allow cross account access
    - Turn on cloudtrail in other accounts and use bucket in paying account.

Lab AWS Organizations Lab 1:

**Cross Account Access** – allows you to separate resources cleanly. Can make it easier to work productively.

* Identify account numbers
* Create Group in IAM – Dev
* Create a user in IAM – Dev
* Log in to Production
* Create the read-write-app-bucket policy
* Create the updateApp cross account role
* Apply the newly created policy to the role
* Login to developer
* Create a new inline policy
* Apply it to developer
* Login as John
* Switch accounts

Accountnumber.signin.aws.amazon.com/login

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "s3:ListAllMyBuckets",

"Resource": "arn:aws:s3:::\*"

},

{

"Effect": "Allow",

"Action": [

"s3:ListBucket",

"s3:GetBucketLocation"

],

"Resource": "arn:aws:s3:::productionapp"

},

{

"Effect": "Allow",

"Action": [

"s3:GetObject",

"s3:PutObject",

"s3:DeleteObject"

],

"Resource": "arn:aws:s3:::productionapp/\*"

}

]

}

{

"Version": "2012-10-17",

"Statement": {

"Effect": "Allow",

"Action": "sts:AssumeRole",

"Resource": "arn:aws:iam::PRODUCTION-ACCOUNT-ID:role/MyDevelopersAccess"

}

}

Tagging & Resource Groups:

**Tags** – key value pairs attached to AWS resources. Metadata

* Can be inherited

**Resource groups** – make it easy to group resources using tags that are assigned to them.

* Contain region, name, health checks, specific information.
* Can have nested tags.

**VPC Peering** – connection between two VPCs that enables you to route traffic using priavet IP address. **Within a single region.** No single point of failure. Can connect between another AWS account.

* Overlapping CIDR blocks can cause conflicts. Transitive Peering NOT supported.
* Can’t create VPC peering between VPC’s in different regions.

**Direct Connect** – easy way to establish a dedicated network connection from premises to AWS.

* Reduces cost with large volumes of traffic
* Increase reliability
* Increase bandwidth
* Doesn’t involve the internet
* Dedicated Line -> Cross Connect -> Direct Connect
* 10 Gbps or 1Gbps or Sub 1 Gbps (only purchased through Direct Connect Partners)
* Uses VLAN trunking 802.1Q.

**Security Toke Service (STS) –** grants users limited and temporary access to AWS resources

* **Federation (AD)**
  + Uses SAML
  + Doesn’t need to be a user in IAM
  + SSO without assigning IAM
* **Federation with Mobile Apps**
  + FB/Amazon/Google or other OpenID
* **Cross Account Access**

**Federation** – combining or joining a list of users in one domain with a list of users in another domain

**Identity Broker** – a service that allows you to take an identity from point A and join it to Point B

**Identity Store** – Services like AD, FB, Google

**Identities** – user of a service like FB.

1. Employee enters their username and password
2. Application calls and Identity Broker. The broker captures the username and password
3. The Identity Broker uses the organizations LDAP directory to validate the employee’s identity
4. The identity Broker calls the new GetFederationToken function using IAM credentials. The call must include an IAM policy and a duration (1 – 36 hours), along with a policy that specifies the permissions to be granted to the temporary security credentials
5. The STS confirms that the policy of the IAM user making the call to GetFederationToken gives permission to create new tokens and then returns four values to the application: an access key, a secret access key, a token, and a duration(tokens lifetime).
6. The Identity Broker returns the temporary security credentials to the reporting application.
7. The data storage application uses the temporary security credentials (including the token) to make requests to Amazon S3.
8. Amazon S3 uses IAM to verify that the credentials allow the requested operation on the given S3 bucket and key
9. IAM provides S3 with the go-ahead to perform the requested operation
10. Develop an Identity Broker to communicate with LDAP and AWS STS
11. Identity Broker always authenticates with LDAP first, THEN with AWS STS
12. Application then gets temporary access to AWS resources
13. Develop an Identity Broker to communicate with LDAP and AWS STS.
14. Identity Broker always authenticates with LDAP first, gets an IAM role associate with a user.
15. Application then authenticates with STS and assumes that IAM Role
16. Application uses that IAM role to interact with S3

How to federate AD with AWS console:

1. Flow initiated when a user puts in his domain. ADFS is installed on a VM
2. SSO against AD
3. Browser receives SAML assertion from authentication response from ADFS
4. Bob’s browser posts the SAML assertion to AWS endpoint for SAML. API request temporary security credentials
5. Browser receives the sign-in URL.

Authenticate against ADFS and then receive security temporary key. ADFS authenticates with SAML

**Workspaces** – cloud based replacement for traditional desktop. Bundle of compute resources that allow users to perform day –to-day tasks. Can connect to any supported devices if AD is integrate by the existing domain.

* Don’t need access in the console
* Windows 7 experience by Windows Server 2008 R2.
* Personalize settings and can lock it down and install your own applications for local administration.
* Persistent.
* Data on D:\ is backed up every 12 hours.

Reread summaries and schedule exam.