AWS Cloud Practitioner Master Sheet for Exam

What is Cloud Computing?

- On-demand delivery of compute, database storage, applications, other IT resources through cloud platform via Internet
- Pay-as-you-go pricing

6 Advantages of Cloud Computing

1. Trade capital expense for variable expense
    - Only pay for what you use
2. Benefit from massive economies of scale
    - You won't have same purchasing power as Amazon
    - They get cheaper prices to purchase servers, hardware
3. Stop guessing about capacity
    - You'll buy too much or too little. Too much = wasted money, too little
4. Increase speed and agility
    - Websites/apps can scale infinitely with demand
5. Stop spending money running/maintaining data centers
    - Focus on what you're good at, not managing infrastructures
6. Go global in minutes
    - Deploy apps in minutes
    - Provide lower latency and better experience at minimal cost

3 Types of Cloud Computing

1. Infrastructure as a Service (IAAS)
    - You manage the server (physical or virtual) as well as the operating system
    - Data center provider has no access to your server
2. Platform as a Service (PAAS)
    - Someone else manages the underlying hardware and operating systems, you just focus on your applications
    - You upload your code and it just executes
    - Think of GoDaddy
3. Software as a Service (SAAS)
    - Think of Gmail
    - All you do is interact with the application, manage the software and how you want to use it
    - Someone else takes care of the infrastructure and everything related to it

3 Types of Cloud Computing Deployments

1. Public Cloud – AWS, Azure, Google Cloud Platform
2. Hybrid – Mix of public and private
    - May want to keep some sensitive data on-premise
3. Private Cloud (or on-premise) – You manage it in your data center. Openstack or Vmware

Around the World with AWS

Region

- Geographic area consisting of 2 or more availability zones

Availability Zone

- A data center

Edge Location

- CDN Endpoints for CloudFront
- Many more edge locations than regions

Let’s Log Into AWS

Support Plans

1. Basic
2. Developer
    - Experimenting with AWS
    - $29/month
    - One person can ask technical questions through support center, 12-24 hour support rate
3. Business
    - 24/7 support by phone
    - Full access to AWS Trusted Advisor
    - $100/mo
4. Enterprise
    - $15,000/month
    - Everything in business + technical account manager
    - 15 min response time for critical support cases

Create Billing Alarm

- Click Name at top-right, click My Billing Dashboard
- Enable Billing Alert
- Add the threshold in Cloudwatch and add e-mail address

Identify Access Management

Tick off five marks on Security Status:

1. Delete root access keys
    - We can skip this as a new user
2. Enable MFA (multi-factor authentication)
    - Set up MFA with Google Authenticator
3. Create individual IAM users
    - Access types:
        - **Programmatic access:** Access via command line

        - **AWS Management Console access:** Login to AWS console and make changes
        - **AWS SDK access:**
    4. Create IAM groups
        - Once you decide access, you need to add to a Group
        - Choose a policy access type (or multiple types) and add the group name
    5. IAM password policy
        - A password policy is a set of rules that define the type of password an IAM user can set
        - Change upper/lowercase required, number required, min password length, etc.

Policies

- How to define permissions to users, groups, roles
- You can click on a policy to get details about what it gives people access to
- Detailed JSON allows you to define a **statement** comprised of an effect (Allow, Disallow), Action (what will happen), and Resource (to what resource)

Exam Tips

3 ways to access AWS platform:

1. Via AWS Console
2. Programatically using command line
3. Using Software Development Kit (SDK)

Root account

- Full admin access
- Should never give away
- Instead, create a user for each individual in your organization and secure root account with MFA

Groups

- Place to store your users
- Users will inherit all permissions that group has
- To set permissions for a group, need to set policies for that group using JSON

S3 (Simple Storage Service)

Overview

- Provides developers and IT teams with secure, durable, highly-scalable object storage
- Safe place to store your files

- Object-based storage (pictures, Word files, videos), not operating system or database
- Files can be anywhere from 0 bytes to 5 TB in size
- Unlimited storage, but you pay by the gig
- Files are stored in **buckets**
- Bucket is a folder in the cloud
- Buckets are universal namespace. Must be unique globally.
- URL looks like **https://s3-eu-west-1.amazonaws.com/acloudguru**
    - Region + Region # + amazonaws.com + bucket name
- You will receive HTTP 200 code if file upload is successful

Data Consistency Model

- Read after Write consistency for PUTS of new objects
    - If you read a file as soon as you upload it, you'll be able to read the file
- Eventual consistency for overwrite PUTS and DELETES (can take some time to propogate)
    - If you update a file and overwrite the old version, you may get the old file or the new file. It will eventually show up.
    - You may be updating to one availability zone, may take time to propagate to other availability zones

S3 Key-Value Store

- S3 is object based, objects consist of:
    1. Key (name of object, e.g. *hello.txt*
    2. Data in file (sequence of bytes)
    3. Version ID (important for versioning)
    4. Metadata (data about data, e.g. tags)
    5. Subresources:
        1. Access Control List
        2. Torrents

Other Points

- Built for 99.99% availability
- Amazon guarantees 99.99999999999% (11 x 9s) durability
    - If you upload x amount of files, 99.99999999999% of those files will actually be uploaded (you won't lose any files)
- Tiered storage availability
- Lifecycle management
    - If a file is over 30 days old, move it from one storage tier to another and eventually archive to Glacier
- Versioning
    - Multiple versions of a file
- Encryption
- Secure data with Access Control Lists and Bucket Policies
    - Bucket policies: Policy is for a specific bucket
    - ACL: Individual file level, control who can access a specific file

S3 Storage Tiers/Classes

1. S3 Standard
    - 99.99% availability

- 99.99999999999% durability
- Stored redundantly across multiple devices (multiple disks) and multiple facilities (multiple availability zones)
- Designed to sustain loss of 2 facilities concurrently

2. S3 – IA (Infrequently Accessed)
    - Data accessed less frequently but requires rapid access when needed
    - Lower fee than Standard but charged a retrieval fee
3. S3 One Zone – IA
    - Same as S3 – IA but do not require multiple availability zone data resilience
    - Only stored in one availability zone
4. Glacier
    - Used for archival only
    - Cheapest
    - Expedited, standard, or bulk
        - Expedited: Restored within few mins, high fee
        - Standard: 3-5 hours for restore
        - Bulk: 5-12 hours

- No retrieval fee for Standard, only for the other three

S3 – Charges

Charged for:

- Storage
- Requests
- Storage management pricing
    - Tags that define who owns an object
- Data transfer pricing
    - Transferring from one region to another
- Transfer acceleration
    - Fast and easy secure transfers across long distances

S3 Transfer Acceleration

- Users upload to an edge location instead of directly to S3 bucket
- Once it goes to an edge location, it automatically gets distributed to the S3 bucket
- File goes across Amazon's backbone to transfer much faster

Read the S3 FAQ before taking the exam!

Creating an S3 Bucket

- Buckets must have unique names
- **Note:** Interface for S3 is Global (similar to IAM), but buckets created can be deployed in any region
- Bucket names must be DNS compliant (3-63 characters, no invalid characters)
- By default, buckets are **private** (recommended)

- You can change storage class and encryption on the fly by using the More menu

Setting public access

- Trying to open a file through a URL won't work by default because public read access is not enabled. Need to enable when uploading.
- Click box next to file > More > Make public
- Another way: Click into file > Permissions tab > Everyone (under public access) > Read Object

Transfer acceleration

1. Click Properties in bucket
2. Advanced Settings > Transfer acceleration > Enable
- S3 has a feature to allow you to test your transfer speeds to different regions around the world

Cross Region Replication

- Management > Replication
- Allows you to replicate bucket in one region to bucket in another region in the world
- Useful for disaster recovery
- Any object upload to first bucket is automatically replicated to second bucket

S3 for Web Pages

- S3 can host **static** web pages (not dynamic like WordPress or PHP)
- It will scale automatically, will scale with demand. Useful for large number of requests.
- You can use bucket policies to make an entire bucket public (used for static websites)

CloudFront

- Amazon's CDN network
- Used to deliver entire website, including dynamic, static, streaming, and interactive content using edge locations
- Requests for content automatically routed to nearest ege location so content is delivered with best performance possible

CDN

- Content-delivery network
- System of services around the world that deliver web pages or web page content to a user based on user geographic location, origin of the webpage, and content delivery server
- Works with origin types listed below
- Also works with non-AWS origins

Edge Location

- Location where content will be cached
- Similar to AWS Region/AZ
- As close to user as possible

Origin

- Origin of files that CDN will distribute
- S3 bucket, EC2 instance, Elastic Load Balancer, or Route53

Distribution

- Name given to the CDN which consists of a collection of edge locations
- First time a user goes to a website, it'll check a local edge location to see if website asset is there
- If not, it will download the asset from the origin and cache it to the edge location
- Next time someone tries to access, they will get the cached version from a local edge location
- Reduces stress on web servers and increases speed to download large files

Distribution Types

1. Web distribution – Used for websites
2. RTMP – Used for media streaming

Setting up CloudFront

1. Choose Web distrbution
2. Origin Domain Name: Choose an S3 bucket
3. Origin Path: You can choose subdirectories for your origin
- Once it's deployed, you will see a domain name. Use that and the name of a file in your bucket to access.

Exam Tips

- Content comes from Origin
- Cached at a local Edge Location
- Takes awhile for the first person to access, much quicker every time after that because it's cached geographically close to you

EC2 (Elastic Cloud Compute)

Setup

- **VPC:** Virtual data center in the cloud
    - Deploy all EC2 instances into a VPC
- **AMI:** Using Amazon Linux AMI because it includes stuff to connect to AWS
- **Instance:** Choosing t2.micro because it's usually used to test in dev
- **Instance Details:**

        - **Network:** Keep default VPC
        - **Subnet:** Choose which availability zone you want to be put into
        - **Auto-assign Public IP:** Allows you to assign a public IP so you can SSH into instance
        - **Shutdown behavior:** Choose what happens if your EC2 instance turns out (stop or have Amazon terminate for you)
        - **Enable termination protection:** Prevents people from accidentally shutting down your instance
    - **Storage:**
        - 8GB is default
        - **Volume Type:** General purpose is most common, Provisioned IOPS lets you choose a very fast disk (database server), Magnetic is a very slow disk (file server)
    - **Tags:** Allow us to add tags like Department and Employee ID to help with cost tracking later on
    - **Security Group:** Virtual firewall in the cloud
        - Open ports like 22 for SSH or 3389 for RDP (Windows) or 80 for HTTP

Connect to the EC2 server

- Open Terminal
- `chmod 400 MyVirginiaKP.pem` – Protects file from accidental overwriting
- `ssh ec2-user@54.242.147.206 -i MyVirginiaKP.pem` to connect
- `sudo su` for root
- `yum update` to update security patches

Exam Tips

- EC2 is compute-based, it's not serverless. It is a server!
- Use private key to connect to EC2
- Security groups are virtual firewalls in the cloud. Need to open ports in order to use them (22 for SSH, 80 for HTTP, 443 for HTTPS, 3389 for RDP)
- Always design for failure, have one EC2 instance in each availability zone

AWS Command Line

- Use `aws configure` on command line to set up login details
    - Enter Access Key and Secret Access Key
    - Region name: us-east-1
    - No output format
- `aws s3 ls` to view s3 buckets
- `aws s3 mb s3://myacloudgurubucket2018` to make a bucket
- `aws s3 cp hello.txt s3://myacloudgurubucket2018sheil` to upload EC2 file to S3 bucket

Tips

- Interact with AWS in 3 ways:
    1. Using the console
    2. Using the command line interface (CLI)
    3. Using the software development kits

Using Roles

- Prevent account from getting hacked
- `cd /~.aws` and `rm -rf credentials` to remove credentials file
- Roles are a secure way to grant permission to entities that you trust
- AWS Console > IAM > Roles
- Create new EC2 role, choose S3 Admin access, and name the role
- In EC2, find the instance, choose Actions > Instance Settings > Attach/Replace IAM Role
- No Role to My Admin S3 Access (the role I created in previous step)
- This process allows me to access S3 via CLI without having to store credentials on the EC2 instance itself

Tips

- Roles are much more secure than using access key IDs and secret access keys
- Much easier to manage
- Can apply roles to EC2 instances at any time (not just when it boots up)
- Changes take place immediately
- Roles are universal, no need to specify region. Similar to users

Build a Website

- Connect to EC2 with CLI
- Web servers need either Apache (Linux) or IIS (Windows)
- `yum install httpd -yes` to install
- `service httpd start` to start server
- Anything you put into `/var/www/html` will be on the website
- `aws s3 cp s3://myacloudgurubucket2018sheil /var/www/html --recursive` to copy files from S3 to web server on EC2

Databases

Relational Database Service (RDS)

Types:

1. SQL Server
2. Oracle
3. MySQL Server
4. PostgreSQL
5. Aurora (Amazon's own database)
6. MariaDB

Two key features:

1. Multi availability zones for disaster recovery
2. Read replicas for performance improvement

- **Multi AZ:** Exact copy of your database in case the primary goes down
    - Disaster recovery
- **Read replica:** Spread read access across five databases, only one is for writing
    - Scaling out / performance

Nonrelational Databases

1. Collection (table)
2. Documents (row)
3. Key-value pairs (fields)

- Allows you to add in extra fields all the time
- **Amazon DynamoDB** is Amazon's nonrelational/NoSQL database
    - Fast, flexible
    - Scales with your application

Aurora

- Relational, Amazon's own
- 6 copies of itself
- 5 times better performance than MySQL, 1/10 price point
- Choose Aurora if you have an RDS
- Choose DynamoDB if you have nonrelational

Data Warehousing

- Used for business intelligence
- Used to pull in large and complex datasets
- Used by management to do queries (current performance targets, etc)
- **Redshift** is Amazon's data warehouse in the cloud for business intelligence
    - Start with a few hundred GB of data, scale to petabyte or more

Autoscaling

- Review: EC2 connects to one database that is duplicated to a second database (redundancy).
- No redundancy on the EC2 itself. Autoscaling group will fix this.
- You can set up how many instances you want with Autoscaling. When one fails, it will automatically create a new one
- You can set a startup script to run when each new instance starts

Route 53

- Amazon's DNS service
- Domain registration

Elastic Beanstalk

- Allows you to deploy everything (provisions everything like EC2 and RDS and everything else) all at one button
- Creates load balancers, auto-scaling groups, security groups, etc.

- Provisioning EC2 instances, installs PHP

CloudFormation

- Way of scripting out infrastructure
- Turning infrastructure into code
- Codify creating EC2 instances, security groups, etc
- JSON that describes your cloud environment – this is a template
- Elastic Beanstack and CloudFormation are free, but you pay for the resources that are provisioned as a result of using EB and CF

Architecting for the Cloud – Best Practices

Why Cloud Computing?

- IT assets becoming programmable resources
- Global availability and unlimited capacity
- High-level managed services, incl call center functionality, text to voice, machine learning, etc
- Security built in (AWS manages security)

Design Principles – Scalability

1. Scale Up – Start with a small virtual machine and increase size
2. Scale Out – Start with an elastic load balancer, add more virtual machines as your project gets bigger
    1. Stateless Applications – Lambda (no state is stored)
    2. Stateless Components – Instead of storing state on server, it stores state on cookies on user's browser
    3. Stateful Components – Can store some stuff with databases that can scale with you (add replicas or increase size)
    4. Distributed Processing – Break your data into pieces and have EC2 instances work on them separately in parallel (Elastic MapReduce)

Design Priciples – Disposable Resources

- Treat your services like cattle, not pets
- If a server dies, just replace with another one
    1. Bootstrapping – Scripts allow you to set up an instance automatically, setup Apache
    2. Golden images – Take an Amazon Machine Image (AMI) and use it for autoscaling
    3. Hybrid of the two

Design Principles – Infrastructure as Code

- Cloudformation

- Allows you to deploy infrastructure to many clients very easily without manually setting anything up

Design Principles – Automation

- Use alarms, events to automate creation/maintenance of infrastructure
- Loose coupling: Make sure failure in one component doesn't affect other pieces of infrastructure
    - Well defined interfaces: Use RESTful API
    - Service discovery: Don't use fixed IP addresses. Instead use DNS names/endpoints.
    - Asyncronous integration: Messages (actions) remain in queue so if one EC2 goes down, the actions are stored in queues for the next EC2 to pick up
    - Graceful failure: If something breaks, nicely tell the user and report to developers

Design Principles – Serverless not services

- Managed Services (other compnanies like Paypal)
- Serverless Architectures (Lambda, DynamoDB, etc)

Design Principles – Databases

- Relational: Aurora
    - High scalability
    - High availability (6 copies of data at any given time)
    - Data needs joins or complex transactions
- Nonrelational: DynamoDB
    - High scalability
    - High availability
    - Data does not need joins or complex transactions
- Data Warehouse: Red Shift
    - Meant for data for business analysis
    - Red Shift is highly scalability and available
    - Red Shift not meant for online transaction processing (not production database)

Design Principles – Search

- Cloud Search or Elastic Search
- Cloud search: less control, easier
- Elastic search: more control
- Both are very scalability

Design Principles – Misc

- Remove single points of failure, everything should have redundancy
- Detect failure with monitoring (Health checks)
- Durable data storage
    - Don't store all on an EC2 instances
    - Store instead in S3 or Dynamo
- Automate multi-center resilience (multiple Availability Zones)
- Introduce fault isolation and horizontal scaling

Design Principles – Financial

- Optimize for cost
    - Elasticity: More servers when busy, less when not busy with auto saling
    - Purchasing options:
        - Reserviced Capacity
        - Spot Instances

Design Principles – Caching

- Application Caching
- Edge Caching

Design Principles – Security

- Offload security to AWS
- Reduce privileged access
- Treat security as code

Tips

- Understand the basic services:
    - Databases – RDS, DynamoDB, Red Shift
    - Compute – EC2 vs Lambda
    - Storage – S3 (great for static hosting)

Summary of Cloud Concepts and Tech Summary

General

1. 6 Advantages of Cloud
2. 3 Types of Cloud Computing
    1. Infrastructure as a Service (IAAS) – Lightsail
    2. Platform as a Service (PAAS)
    3. Software as a Service (SAAS)
3. 3 Types of Cloud Computing Deployment
    1. Public Cloud (AWS, Azure, Google Cloud)
    2. Hybrid (mix)
    3. Private Cloud (managed locally)
4. Difference between:
    1. Regions – London, Frankfurt, N. Virginia
    2. Availability Zones – Collections of data centers, geographically distributed
    3. Edge Locations – Caching
5. Access AWS Console by:
    1. Via AWS console
    2. Programatically using command line
    3. SDKs

6. Root account has full admin, never give out. Create user for each individuals and secure with multi-factor auth.
7. Groups are places to store users
8. Set permissions in group with policies with JSON

S3

1. S3 bucket is a place to store objects
2. S3 unique namespace
3. Object based only, 200 status code when complete
4. Storage places:
    1. S3 – Current data
    2. Glacier – Archival (2-5 hour retrieval)
5. Restrict access with bucket policy
6. Restrict access to indiv objects with access control lists
7. S3 transfer acceleration – Upload to edge locations. Edge locations then send to central place.
8. Cross-region replication – Replicate to other buckets
9. S3 hosts static websites
10. Scales automatically to meet demand (movie preview)

Cloudfront

1. Edge Location: Location where content is cached
2. Origin: Origin of files that CDN will distribute (S3, EC2, Elastic Load Balancer, Route53)
3. Distribution: Name given to CDN, consists of edge locations
    1. Web – Websites
    2. RTP – Media streaming
4. Can write to edge locations (S3 transfer acceleration)

EC2

- NOT SERVERLESS, compute-based
- Private key to connect
- Security Groups: Virtual firewalls in the cloud, open ports to use
- Design for failure, have one EC2 instance in each Avail Zone
- Pricing models
- Types of EC2 depending on the purpose of EC2
- EBS: Elastic block storage where you install operating system and file
- 4 kinds of EBS:
    - General Purpose SSD
    - Provisioned IOPS SSD
    - Throughput Optimized HDD
    - Cold HDD
- Roles much more security and easier to managethan using access and secret access keys
- Roles are universal, no need to specify users

RDS

1. Multi Avail Zone: Disaster Recovery
2. Read replicas: Scaling out or performance
3. DynamoDB for nonrelational, Aurora for relational, Red Shift for data warehousing

Billing

- Philosophy on pricing: Pay for what you use, start or stop using product at any time. No long-term contracts required.
- Free Tier to help new AWS users get started

Pricing policies

- **Pay as you go: **EC2 used to be pay by hour, pay by second as it's used
- **Pay less when you reserve:** If you reserve time ahead of time, you get a discount
- **Pay even less by unit when using more: **If you use more, you pay less per GB
- **Pay even less as AWS grows**
- Custom pricing for enterprise

What's free?

1. Amazon VPC
2. Elastic Beanstalk (services it provisions are not free)
3. CloudFormation (services it provisions not free)
4. Identity Access Management (IAM)
5. Auto Scaling (EC2 instances it uses are not free)
6. Opsworks
7. Consolidated Billing (add all AWS accounts into one bill)

3 Fundamental Charges

1. Compute
2. Storage
3. Data Out to Internet (Data In is free)

What determines price?

1. Clock hours of server time (time server is running)
2. Machine configuration (more resources consumed = more paid)
3. Machine purchase type (some instance types cost more)
4. Number of instances
5. Load balancing
6. Detailed monitoring (monitor EC2 by minute instead of 5-min intervals)
7. Auto scaling (EC2 instances cost money)
8. Elastic IP Addresses
9. Operating systems (Windows) and software packages

- Elastic Compute Cloud can reserve instances ahead of time, even cheaper if you pay upfront

S3 – What determines price?

1. Storage class (Standard or IA)
2. Storage amount
3. Number of requests
4. Data transfer (data transfer out)

RDS – What determines price?

1. Number of hours RDS is running
2. Database characteristics (licensed?)
3. Database purchase type (huge, nano?)
4. Number of instances
5. Provisioned storage (how big?)
6. Requests made to database
7. Deployment type (multi A-Z, read replicas)
8. Data transfer out

Cloudfront – What determines price?

1. Traffic distribution
2. Requests
3. Data transfers out

Billing: Support Plans

1. Basic
    1. Free, no tech acct mgt, no open cases
2. Developer
    1. $29/mo, business hr access via email, no TAM, 1 person can open unlim cases
    2. General guidance: < 24 business hours
    3. System impaired: < 12 business hours
3. Business
    1. $100/mo, 24×7 email, chat, and phone support, no TAM, unlimited cases for support
    2. General guidance: < 24 business hours
    3. System impaired: < 12 hours
    4. Prod system impaired: < 4 business hours
    5. Prod system down: < 1 hour
4. Enterprise
    1. $15,000/mo, 24×7 email chat and phone, TAM, unlimited cases for support
    2. General guidance: < 24 business hours
    3. System impaired: < 12 hours
    4. Prod system impaired: < 4 business hours
    5. Prod system down: < 1 hour
    6. Business critical down: < 15 mins

- Pricing can be higher if you use AWS a lot

Billing: Resource Groups

- Tags are key-value pairs attached to resources
- Tags can be inherited (created by one service, moves to another service)
- **Resource groups:** Make it easy to group resources based on tags assigned to them
- Resource groups contain info like:
    - Region
    - Name
    - Healthchecks

        - EC2 – Public/Private IP Addresses
        - ELB – Port Configs
        - RDS – Database Engine
    - You can search for resources by a specific tag (used by a particular department, user ID, etc)
    - Tag Editor allows you to find resources not tagged and add tags

Billing: Consolidated Billing

- **AWS Organization:** Enables you to consolidate multiple AWS accounts into an organization that you create and centrally manage
- **Consolidated billing:** One monthly bill (paying account) for all linked accounts in organization
- 20 linked accounts for consolidating billing
- Easy to track charges and allocate costs
- Volume pricing
- You can also reserve EC2 instances and if one group isn't using them, you can carry them over to another group to save money
- Best practices:
    - Always enable multi factor auth
    - Strong and complex factor
    - Restrict root access
- Billing alerts

Exam Tips

- Consolidated billing allows you to get volume discounts for all your accounts
- Unsused reserved instances for EC2 are applied across group
- CloudTrail is on per-account and per-region basis , can be aggregated into single bucket in paying account

AWS Quick Starts

- Allow you to enable a particular type of technology very quickly
- Templates to get you started with a server that runs a particular technology
- Uses CloudFormation based on a template URL

AWS Cost Calculators

Simple Monthly Calculator

- Allows you to quickly add the resources you're going to use and the types of resources and it'll tell you the cost of each and total monthly bill
- Not comparing what you have on premise and in cloud

Total Cost of Ownership Calculator

- Compares against your current costs for total cost of ownership
- Takes into account:

- Server costs (hardware & software)
- Storage costs (hardware & storage admin)
- Networking costs (network hardware & network admin)
- IT labor costs

Billing & Pricing Summary

- Remember the free services!
- AWS Support Plans and features of each
- What are tags?
- What are resource groups? Group resources based on tags
- What is the benefit of consolidated billing?
- What's the benefit of AWS Quick Starts?
- Two different AWS calculators

AWS Compliance

Certifications / Attestations

AWS certified with:

1. ISO 27001
2. PCI DSS Level 1
3. SOC 1
4. SOC 2
5. SOC 3

Laws, Regulations, Privacy

1. HIPAA compliant – Meets standards to store health information

Alignments / Frameworks

1. G-Cloud (UK) – Frameworks for government customers to meet these requirements in UK

Shared Responsibility Model

- AWS manages security of cloud, security in cloud itself is responsibilty of customer. Customers are responsibility for security of how AWS is set up. AWS is responsible for the infrastructure
- Do you have the ability to stop something from happening? If you dont have the ability to stop it, it's Amazon's responsibility
- You have control over encryption, customer data

AWS Web Application Firewall and AWS Shield

AWS WAF

- Application firewall that helps protect your web apps from common web exploits that could affect availability, compromise security, or consume excessive resources
- AWF can read data hacker is sending and can intervene on your behalf
- Prevents common attacks
- Goes down to Layer 7

AWS Shield

- Managed DDOS service
- Provides safeguards for web apps running on AWS Two tiers:

1. Standard – Free, avail automatically
2. Advanced – Advanced protection for $3000/mo

AWS Inspector vs AWS Trusted Advisor

AWS Inspector

- Automated security assessment service
- Automatically asses apps for vulnerabilities or deviations from best practices
- Assessment done, provides detailed list of security findings prioritized by leve of severity

AWS Trusted Advisor

- Optimizes AWS environment to reduce cost, increase performance and improve security

1. Cost Optimization (do you have an EC2 with nothing happening on it or an empty DB?)
2. Performance
3. Security
4. Fault Tolerance (are you using multiple avail zones?)

- Two options:

1. Core checks and recommendations 2 Full trusted advisor – business/enterprise only

Security Summary

- Name some of the compliance that AWS meets (above)
- Define what shared responsibility means
- AWS WAF reads data and blocks traffic if it will cause problems
- AWS shield blocks DDOS attacks. Two tiers: Standard and Advanced
- Inspector looks for vulnerabilies on your EC2 instances.
- Advisor gives suggestions for improvement, advanced one requires business subscription