CCP Exam Tips (2019 version) [Editable]

Section 1: Cloud Concepts

6 advantages of Cloud:

Trade Capital Expense for Variable Expense (Pay only for what you use)
Benefit from massive economies of scale (Amazon is such a big power for providing powerful services instead of building everything by your own)
Stop guessing about capacity (auto scaling, no need guess)
Increase speed and agility (auto scaling, auto increase when you need it)
Stop spending money on running and maintaining data centers (That's why we need AWS Cloud)
Go global in minutes (AWS services helps you to deploy your apps to the cloud and share with the world)

3 types of cloud computing:

Infrastructure As A Service (IAAS): EC2 (Amazon Web Server)
Platform As A Service (PAAS): Go daddy, Elastic BeanStalk (You upload code they will provision online resources for you)
Software As A Service (SAAS): Gmail

3 types of cloud computing deployments:

Public Cloud: AWS, Azure, GCP
Hybrid Cloud: Mixture of public and private cloud
Private Cloud: you manage it which is in your company/local data center, such as Openstack or VMware

Region vs Availability Zone vs Edge Locations:

Region: is a physical location which contains 2 or more availability zones
Availability Zone: refers to 1 or more data center, each availability zone has redundant facilities or powers for backup purpose
Edge Location: basically are endpoints which used for caching contents, such as CloudFront (AWS CDN)

Support packages:

Basic - Free
Developer $29/month (scale based on usage)
Business: $100/month (scale based on usage)
Enterprise: $5000/month (scale based on usage) [Get a TAM: Technical Account Manager (AWS Technical Support: serve for you)]

Section 2: Technology

IAM (Identity Access Management)

IAM (Identity Access Management) [Free service and no need to consider region]
When you create a user or group, it will be public and globally !!!

3 ways to access / interact with AWS platform

AWS Console
CLI (Command Line): not safe, coz once secret key and access key were exposed to someone else, then disaster will happen, the 'someone' can access your AWS platform and use any of your services
SDK (Software Development Kit)

Using roles which are much more secure than using the CLI to access AWS platform, coz you configure everything on cloud instead in your local PC !!! (We can update role policies to have better and secure experience to use different services, such as some roles can access database, apply roles to EC2 instance any time you want)

Roles are universal, similar with users, you DON'T need to specify what regions they are in

Root Account

Root Account: is the email address you used to set up for your aws account, and root account always has FULL admin access.

For security reasons, just create another user account within your organization for other users, instead of sharing your root account, (NEVER SHARER Root account Please)

Should always secure root account with multi-factor authenticator (Google Authenticator)

Group VS User

Group is simply a place where stores your users, your users will inherit all permissions that group has (Examples of group: Developers, Admins, Finance, Human Resources)

To setup permissions in a group, you need to set policy to a group, and policy is basically a JSON object, which refers as {key: value} pairs, such as {“name”: “service”}

User: is user account for accessing other services

S3

S3: AWS storage service (Not Free !!!): is object-based (allow to upload files)

File size: 0 Byte to 5TB

Files stored inside buckets (unlimited storage)

Bucket name must be unique (S3 is a universal namespace)
Example bucket url: https://s3-region-name.amazonaws.com/unique-bucket-name

Not suitable to install operating system on

After file uploaded successfully, will give http 200 status code

Also S3 is a key: value pairs, key is object name, and value is the actual data

When uploads a new file, S3 will give consistency which is 200 status code
When update (PUTS/DELETE) or overwrite the old file, the update may not be immediately, it may stay a while and then updated to the latest file, S3 takes time to propagate

Can be used to STATIC website hosting, not database based website hosting, only STATIC

S3 supports auto-scale based on demand for serving your static site with better performance

6 different S3 storage classes

S3-Standard: 99.99% availability, and 99.999999999% durability, and which save files in multiple devices over multiple facilities

S3-Infrequent-Access (S3-IA): for data less visited, but when needed, needs to rapid access

S3-One-Zone-Infrequent-Access (S3 One Zone IA): save less frequent files, and not need to save multiple devices over multiple facilities

S3-Intelligent-Tier (S3-IT): Using machine learning technology to move your files to different storages

S3 Glacier: is a lower cost storage for data achieving, and retrieval time from minutes to hours, which is configurable

S3 Glacier Deep Archive: is the S3 LOWEST cost storage class and retrieval time of 12 hours is acceptable

Bucket

Bucket name must be unique, non-repeatable (CANNOT have same bucket name)

You can view buckets globally and you also can have buckets in individual regions

Can replicate contents of bucket from one region to another by using **Cross Region Replication** functionality

S3 transfer acceleration: can upload files to edge location, so different regions visitors can access the edge locations to get the files much quicker (Also S3 bucket functionality !!!)

Can use bucket policies to make entire S3 bucket being PUBLIC

CloudFront

What is CloudFront? It is a CDN (Content Delivery Network) service provided by AWS, serve your app for different regions users in a faster way (load web pages faster for different regions users, such as hosted on Australia, and display in EU, please using CloudFront)

--- 3 terms:

Edge Location: The location where the contents will be cached. (This is separate to AWS region or AZ (Availability Zone))

Origin: This is the origin of all the files that the CDN will distribute, it could be S3 bucket, Route 53, Elastic Load Balancer and EC2 instances

Distribution: This is the name given the CDN which contains collections of edge locations

--- Two types:

Web Distribution: typically used for websites
RMTP: used for media streaming

--- Other tips:

You can write edge locations (such as put an object on edge locations)

Object cached for the life of TTL (Time To Live)

You can clear your cached object, you need to PAY

EC2

EC2: is a web server provider, user can create different web server instances, such as Linux, Windows and etc (basically is a virtual server in the cloud)

--- 4 pricing models:

On Demand: allows to pay hourly fixed rate without commitment

Reserved: provides you with a capacity reservations, and got hourly rate discount with the contract term between 1 to 3 years

Spot: Enable to bid price for instance capacity, providing for great savings if your app have flexible start and end times (If AWS terminate instance you won't get charged, if you terminate spot instance by yourself, you are going to PAY the BILL)

Dedicated Hosts: Physical EC2 server dedicated for your use, reduce costs by allowing you to use your existing server-bound software licenses.

--- Instance Types: FIGHTDRMCPXZ

F: For FPGA

I: For IOPS

G: Graphics

H: High Disk Throughput

T: Cheap General Purpose (Think T2 micro)

D: For Density

R: For RAM

M: Main choice for general purpose apps

C: For Compute

P: Graphics (think Pics)

X: Extreme Memory

Z: Extreme Memory AND CPU

--- EBS: it just the virtual hard-disk in the cloud

SSD:

Type 1: General Purpose SSD [GP2]: normal usage balance between price and performance

Type 2: Provisioned IOPS SSD [IO1]: used for high performance purpose only

Magnetic:

Throughput Optimized SSD [ST1]: low cost HDD designed for frequent access workloads, such as data warehouse

Cold HDD [SC1]: lowest cost for less frequent accessed workloads, such as file servers

Magentic: (Previous Generation)

EC2 Security Groups

EC2 Security Groups: is a virtual firewalls in the cloud and you need to open the port in order to use them, such as https: 443, http: 80, RDP (Remote Access Protocol): 3389 and SSH: 22

Always design for failure: better have one EC2 instance in each availability zone !!!!!

Elastic Load Balancers

Elastic Load Balancer: it has 3 types, which contains load balancers, network load balancers and classic cloud balancers

Application Load Balancer (ALB): Layer 7 helps to make intelligent decisions

Network Load Balancer (NLB): Extreme performance with static IP addresses

Classic Load Balancer (CLB): Test and develop purpose which keeps costs lower

AWS Databases

RDS: Relational Databases: it has few tools, SQL, MySQL, Oracle, Aurora and MariaDB

DynamoDB (No SQL) [Non-relational databases] -> tool: DynamoDB

RedShift OLTP(OLTP: On Line Transaction Processing) (From infrastructure layer): is AWS data warehouse solution, which is used to store all the data [suitable for data warehousing & business intelligence solution]

Elasticache: it has 2 types: Memcached and redis, the purpose of using Elasticache is if we need to speed up for the existing databases (Frequent Identical Queries)

RDS has 2 key features: Multi-AZ: for disaster recovery & Read Replicas: For performance !!

Route 53

Route53 is AWS DNS service, it is GLOBAL which is similar with IAM and S3

You can use it to direct traffic all over the world and use it for domain registration purpose !!

Elastic Beanstalk

Elastic Beanstalk: you can deploy and manage applications quickly to the AWS cloud without running infrastructure on those applications. You can simply UPLOAD your application to Elastic Beanstalk, and which automatically helps you to handle the details of capacity provisioning, load balancing, auto-scaling and application health monitoring (super powerful!)

CloudFormation

CloudFormation: is a service that helps you model and setup your AWS resources so that you can spend less time managing those resources and more time focusing on applications that run in AWS.

You create a TEMPLATE that describes all AWS resources that you want to use for your application (such as AWS EC2, RDS DB instances and so on), and CloudFormation takes care of provisioning and configuring those resources for you automatically. (Magic !!!!)

You don't need to individually create and configure AWS resources and figure out what dependent on what … AWS CloudFormation will handle all of those headaches for you !!!!!!!!

Similar and difference between Elastic Beanstalk and CloudFormation

Similar: Elastic Beanstalk and CloudFormation both are FREE services, however, the resources they provision (such as EC2, RDS ..) are NOT free !!!

Difference: Elastic Beanstalk is limited in what it can provision and is not programmable, and CloudFormation can provision almost any AWS service and is completely programmable.

Architecture For the Cloud : Best practices !!! (Please read white paper provided by AWS before join the AWS CCP exam, cheers !)

1). Traditional Computing vs Cloud Computing (6 areas comparison):

[IT Assets as Provisioned Resources] (Traditional computing asked devops need to configure servers, installing operating systems, that's time and human resources wasting …, which is also low efficiency)

[Global Available and Scalable Capacity] (You can deploy your app everywhere, such as Tokyo , London and share your app with the whole of the world, and when your visitors become more and more you can enhance the server capacity to handle more requests !!)

[Higher Level Managed Services] (AWS Cloud Computing supports for doing machine learning inside the databases)

[Built-in Security] (AWS Cloud is quite secure compared with your local server (traditional computing, such as multiple-factors authentication))

[Architecting For Cost] (You can design your infrastructure in the cloud which is quite cost efficient, save your money pocket doing powerful things, such as building serverless app and public to the www and only cost little, compared with setup local server and doing everything manually. Obviously, AWS is much better)

[Operations on AWS] (Such as move virtual machines from local server virtual machine to EC2 cloud server and also remove MySQL from local virtual machine and move it to the AWS RDS database services) - This example is called INFRASTRUCTURE REFACTORING

2). Scalability:

Scale Up: Eg: increase the size of your micro instances [it increases RAM, CPU inside the virtual machine] (EC2 instance scale up)

Scale Out: eg: add more virtual machines for load balancing purposes, in order to provide more stable services to your end users !!!

[Stateless Applications] (such as lambda functions (Serverless Architecture)) !!

[Distribute Load To Multiple Nodes] (WordPress Example: we have multiple EC2 servers and another example is RDS read replicas)

[Stateless Components] (such as store user cookies for session authentication, instead putting everything on the server, browser stores those information, so nothing can be tracked on server, server does not know anything, which means stateless)

[Stateful Component] (Such as store users behaviours inside the database, because you want to save yours end users information for your business good, which means, you need server to handle of storing those state based information, eg: RDS database saves user transaction state information into database !!!)

[Implement Session Affinity] (you stuck to a particular EC2 instance with a sticky session cookie), every time a user visits websites, load balancer will detect the cookie which stored inside the user's browser, and will send this session cookie back to the EC2 instances, so EC2 knows which user visits the server !!!

[Distributed Processing] (used an example of Elastic Map Reduce (AWS EMR))

[Implement Distributed Processing] (The AWS EMR is allowed the user to have a whole bunch of EC2 instances (such as 1000 EC2 instances) and process a really huge amount of data, calculating some big data and give some results)

3). Instantiating Compute Resources:

It means automatically doing something complex jobs for you, such as

[Bootstrapping] (you don't need to manually create EC2 instances every single time, we can write some bootstrap scripts to automatically tell EC2 instance to configure for us, such as WordPress Bootstrapping scripts)

[Gloden Images] (create an image of already configured EC2 instances, so next time, we can deploy the new EC2 instance by calling this image and directly use it, no need another configuration job !!!!)

[Hybrid] (a combination of containers as well as EC2 instances)

4). Infrastructure As Code:

CloudFormation: we can use it for Provision our WordPress website (it will automatically configure the wordpress server in avery quick time)

5). Automation:

[Serverless management and deployment] (CI/CD pipeline to do the deployment job)

[Infrastructure Management and Deployment]

AWS Elastic Beanstalk
Amazon EC2 auto recovery
AWS Systems Manager
Auto Scaling

[Alarm and Events]

Amazon CloudWatch Alarms: billing alert

Amazon CloudWatch Events: setup functionality on detecting someone upload images resources to S3 Bucket

AWS Lambda scheduled events (auto report for the behaviour of lambda function running)

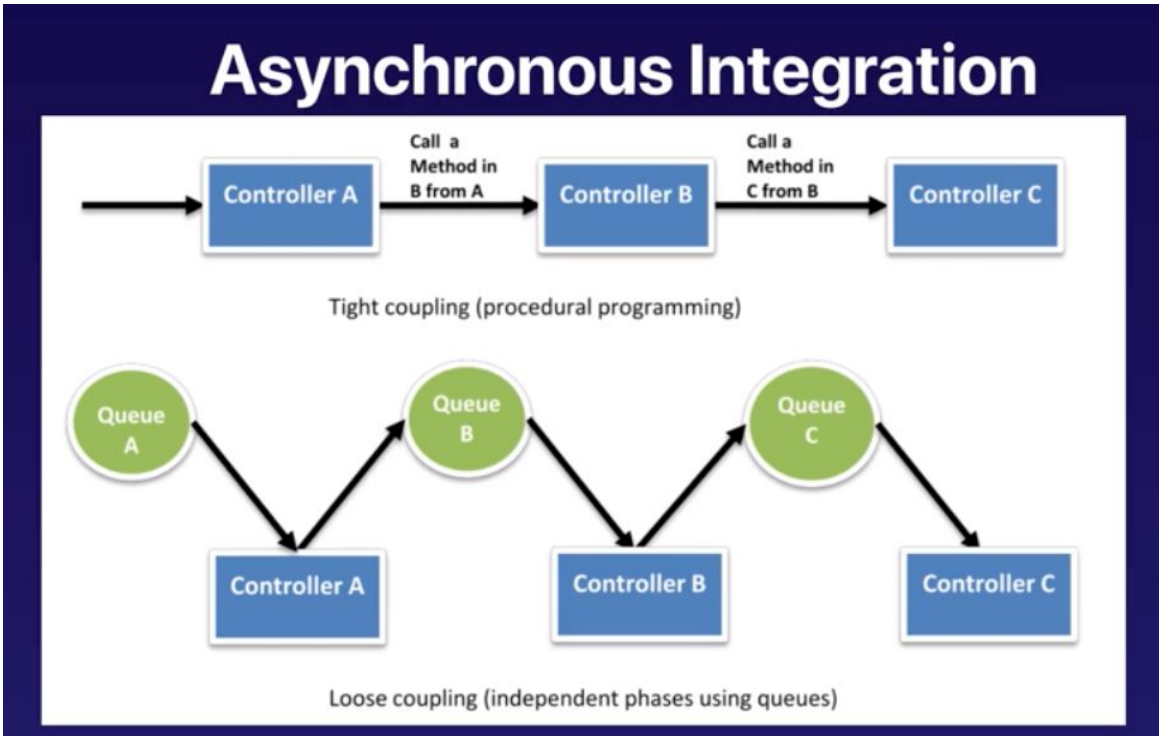
AWS WAF security automations (Firewall and detect danger and report to AWS system admin to notice, such as XSS attack)

6). Loose Coupling:

[Well defined interfaces] (Amazon API gateway (allow to create your own APIs))

[Services Discovery] (Implement Service Discovery (allow one AWS component/service to discover another AWS component/service, such as EC2 uses one public IP address, and RDS cannot use the same address, otherwise will alert system admin the failure))

[Asynchronous Integration] (always have loose coupling in your part of the environment, because we don't want anything failed. For instance, we always want our EC2 instances ON instead of tight coupling, which is synchronous which does not have queue service to handle the process !!!) What is the Queue service? (SQS [Simple Queue Service]) will do the queue service job !!! Check with this chart:



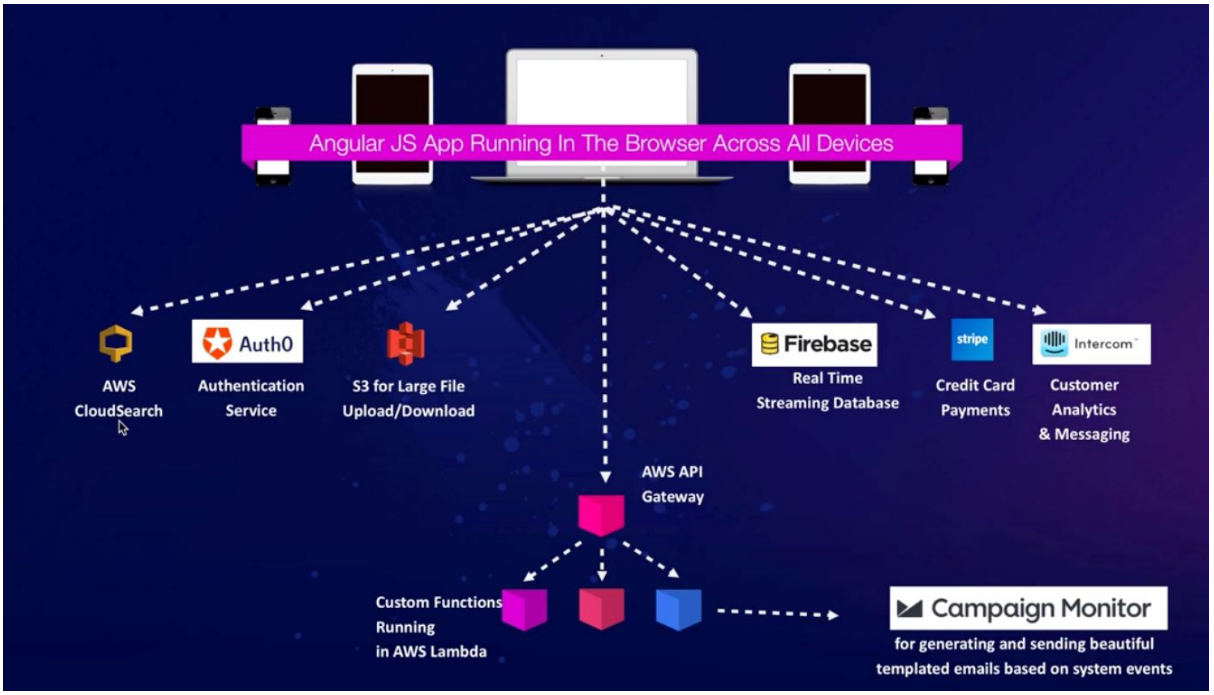
[Distributed Systems Best Practices] (If S3 static web site fails, then we can render the error page to the users, which means Graceful Failure in Practice)

7). Services Not Servers

[Managed Services] (Lambda, S3, route 53, basically anything NOT replied on physical servers)

[Serverless Architecture]: You don't have to manage servers

Check this picture:



8). Databases:

Relational Databases (Aurora):

Scalability: always have 6 copies of your data

High Availability - Multi-AZ (3 availabilities zones or more)

Anti-Patterns: no need for joins or complex transactions, use No-SQL

Non-Relational Databases (DynamoDB):

Scalability: always have 6 copies of your data

High Availability - Multi-AZ (3 availabilities zones or more)

Anti-Patterns: requires joins or complex transactions, use relational databases (Aurora). If you have large binary files (Video, Audio and etc), consider storing the files in S3

Data Warehouse (Redshift):

Scalability: always have 6 copies of your data

High Availability - Multi-AZ (3 availabilities zones or more)

Anti-Patterns: not meant for On Line Transaction Processing (OLTP)

Graph Databases (Neptune):

Scalability: always have 6 copies of your data

High Availability - Multi-AZ (3 availabilities zones or more)

9). Managing Increasing Volumes of data:

A data lake: store massive massive data in a data center and the data can be processed, categorized, analysed and consumed by different groups of people within your organization.

10). Removing Single Points of Failure:

[Introducing Redundancy]
[Detect Failure]
[Durable Data Storage]
[Automated Multi-data Center Resilience]
[Fault Isolation and Traditional Horizontal Scaling]
[Shading] (Split among multiple shades) {shade means a cool place}: advantage: process data a lot faster

11). Optimize For Cost:

[Right Sizing]
[Elasticity] (Flexible: if you have a sales websites, you can open more resources, such as create more load balancers for handling customers who purchase your products, then after the sales days (Black Friday) are over, then turn down those load balancers, which means more flexible)
[Take advantage of the variety of purchasing options] (Reserved Capacity & Spot Instances)

12). Caching:

[Application Caching] (Website caching)
[Edge Caching] (CDN caching)

13). Security:

[Use AWS features for defence inDepth]
[Share security responsibility with AWS] (Understand who is responsible for which part of security)
[Reduce Privileged Access]
[Security as Code]
[Real time Auditing]

Section 3: Billing & Pricing

Read the AWS Price Overview PDF file !!!

AWS Free Services

Amazon VPC
Elastic Beanstalk
CloudFormation
IAM
Auto Scaling
Opsworks
Consolidated Billing

AWS Pricing Models

1) Pay as you go
2) Pay less when you reserve (Reserved, such as spot instance)
3) Pay even less per unit by using more (more resources are using, the less will pay)
4) Pay even less as AWS grows
5) Custom pricing

Tags

Key value pairs attached to AWS resources
Metadata (data about data)
Tags can sometimes inherited (CloudFormation tags)

Resource groups make it easy to group your resources using the tags that are assigned to them. You can group resources that share one or more tags.

We can use it for name, health check region via different AWS services

Using resource groups you can apply automation to resources tagged with specific tags. For example, we stopped all EC2 instances in the Stockholm Region.

Resource Groups in combination with **AWS Systems Manager** allow you to control and execute automation against entire fleets (ship by ship / team) of EC2 instances, all at the push of a button (resource group basically control some tags and those tags can do some actions for interacting with services , such as execute automation against entire fleets (ship by ship) of EC2 instances)

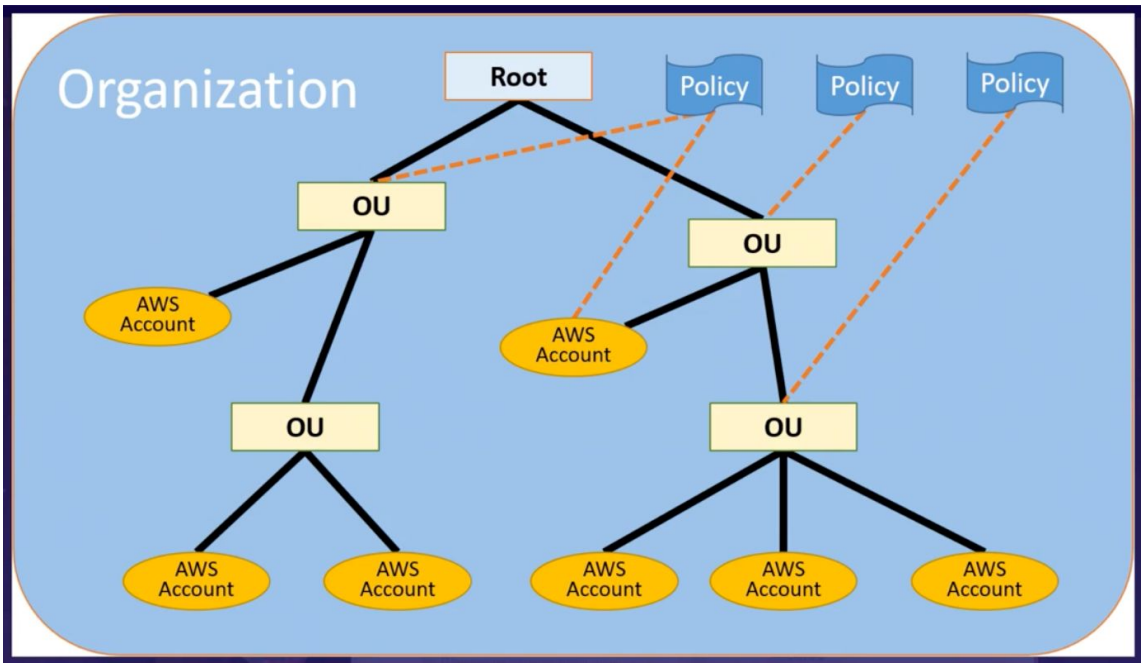
Tag Editor is a global service that allows us to discover resources and to add additional tags to them as well. Newer regions may take some time to be compatible with tag editor.

Consolidated Billing

Consolidated Billing allows you to get volume discounts on all your accounts
Unused reserved instances for EC2 are applied across the group

AWS Organizations

Remember this accounts charts:



Using root account can create few organizational units (OU), and then create policies to attach on organizational units (OU), and then inside each OU, we can create different aws accounts for each unit's user. We also can directly attach the policy to the AWS account as well.

Best practices with AWS organizations:

Always enable multi-factor authentication on root account
Always use a strong and complex password on root account
Billing account should be used for billing purposes only. **NEVER** deploy resources into paying account

Maximum 20 linked accounts ONLY

Billing alert contains all lined accounts bills information and you can create billing alerts for each individual linked account !!!

CloudTrail: a way of auditing what people are doing in AWS platform

- Per AWS account and is enabled per region
- Can consolidate logs using an S3 bucket:
  - Turn on CloudTrail in paying account
  - Create a bucket policy that allows cross-account access
  - Turn on CloudTrail in the other accounts and use the bucket in the paying account

CloudTrail is on a per account and per region basis, but can be aggregated (joined/summed) into a single bucket belonging to the paying account.

AWS Quick Start VS AWS Landing Zone

AWS Quick Start is a way of deploying environments quickly, using CloudFormation templates built by AWS solutions architects who are experts in that particular technology.

AWS Landing Zone is a solution that helps customers more quickly setup a secure multi-account AWS environment based on AWS best practices.

AWS Calculators: (PLEASE GIVE IT A SHOT !!!)

Calculate the cost of each service (How much you need to pay before use those payable services)

AWS Simple MOnthly Calculator: calculate the service monthly payment via an online calculator provided by AWS

AWS TCO (Total Cost Ownership) is used to compare costs of running your infrastructure on premise vs in the AWS cloud. It will generate report that you can give to your C-level execs (manager) to make a business case to move to the cloud.
AWS Snowball: is used for PB-scale data transport, just remember it uses to transfer a massive data out of AWS cloud

Section 4: Security

AWS Compliance: Such as Shared Responsibility Model

Shared Responsibility Model

Must read and fully understand the whole chart and must be 100% clear about who is responsible for which part of security, exam will be cover a hell a lot of questions on this part. (https://aws.amazon.com/compliance/shared-responsibility-model/)

AWS WAF vs AWS Shield

AWS WAF: is a web application firewall, designed to stop hackers doing bad things

AWS Shield: is a DDOS mitigation service designed to stop DDOS attacks

AWS inspector vs AWS Trusted Advisor

AWS Inspector: is used for inspecting EC2 instances for vulnerabilities

AWS Trusted Advisor: inspects your AWS account as a whole (Not just EC2). It does more than just security checks. It also does cost optimization, performance check and fault tolerance!

AWS CloudTrail (A CCTV monitor inside AWS)

It increases visibility into your user and resource activity by recording AWS management Console actions and API calls. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred.

Thanks for reading ^_^