AWS Review Sheet links & notes - Solutions Architect

Sections

**Random Notes/Other Key Services:**

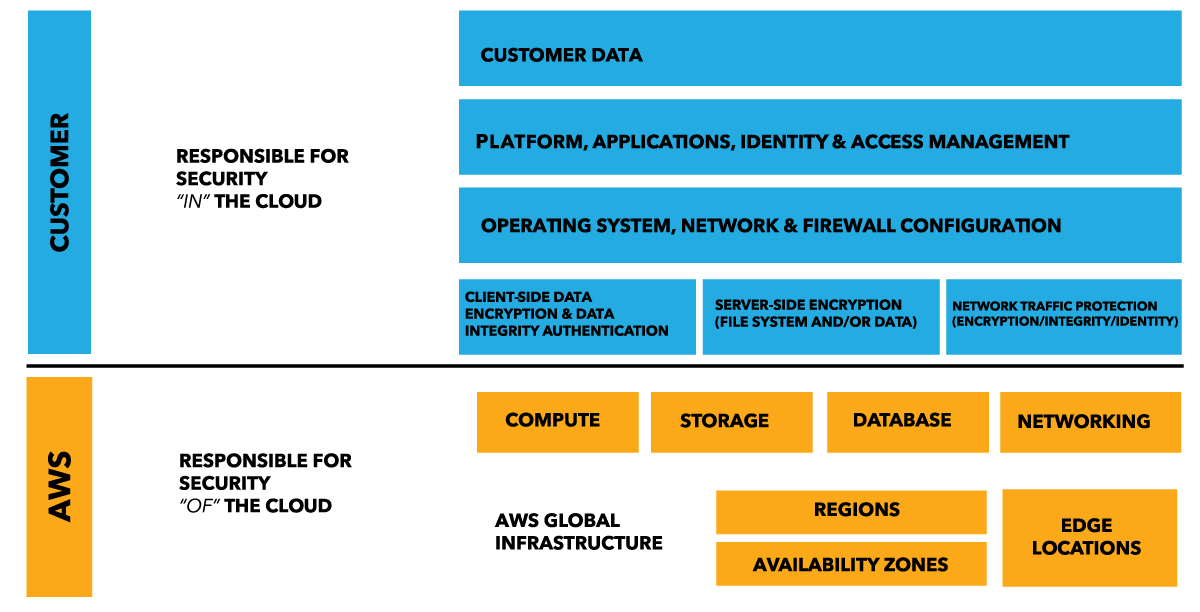
Links:

* optional whitepaper: <http://media.amazonwebservices.com/AWS_Disaster_Recovery.pdf>
* https://aws.amazon.com/blogs/aws/new-whitepaper-use-aws-for-disaster-recovery/

Best Practices/Acceptable Use

* **‘penetration testing’ (port scans) can only be done ON YOUR EC2 INSTANCES, WITH APPROVAL FROM AWS**

**Shared Responsibility Model**



CIDR Blocks

* xx.xx.xx.xxx/y
  + /y tells us HOW MANY ADDRESSES we are allowing
    - 32 🡪 2^0 🡪 1 address
      * Restrict to IP of client
    - 31🡪 2^1 🡪 2 address
    - 30 🡪 2^2 🡪 4 addresses
    - …….
    - 16 🡪2^16 🡪 65536 addresses
    - 0 🡪 any address in the network

Kinesis

* **AWS Solution used for real time processing of streams of data records**
* Service Types
  + Kinesis Streams
    - uses **shards** to store data
    - **Data retained for default 24 hrs up to 7 days**


     Kinesis Streams High-level Architecture Diagram
    

* + Kinesis Analytics
    - Send to stream and analyze real time
    - Run sql queries on stream data
  + Kinesis Firehose
    - **No shards**
    - No data retention window
      * As soon as data comes in it is either moved or processed in real time
* Actors
  + Kinesis Producer
    - Collect data and send to stream

Disaster Recovery Keywords:

* **RTO = Run Time Objective**

-time it takes to restore business process to service level after disruption

* **RPO = Recovery Point Objective**

-acceptable amount of data loss measured in time

-if a disaster occurs at 12:00 PM (noon) and the RPO is one hour, the system should recover all data that was in the system before 11:00 AM. Data loss will span only one hour, between 11:00 AM and 12:00 PM (noon).

AWS Code Pipeline

* Continuous delivery service that enables you to model visualize and automate steps required to release software.
* Builds, tests, and deploys app according to pre-defined work flow

AWS Directory Service

* Types
  + MS active directory
  + Simple AD
  + AD connector

AWS Budgets

* Uses data from ‘cost explorer’ to see usage-to-date and current estimated charges
* Up to 20,000 budgets per acct

CloudTrail

* Used for auditing/logging access to and actions taken on, aws resources
* Stores logs in s3
* Records API calls
  + ID of caller
  + Time of API call
  + Name of api
  + Requested parameters

AWS OpsWorks

* Configure and operate applications with Chef
* Workflow automation from template that can define:
  + App architecture
  + Package installations
  + Software configs… etc

AWS Direct Connect

* Direct connection b/w on-prem environment and AWS (not used for just for large file transfers)
* Extend environment like VPN
* Types
  + Public
  + Private

Amazon Import/Export

* Used to increase bandwidth for large data uploads to AWS

**HTTP STATUS CODES**

200=success

300=redirect

400= client side error

* 400 bad request
* 403 forbidden
* 409 conflict

500= server side error

Tagging

* Limits
  + **Max Tags/Resource = 50**
  + **Supported Services for tags**
    - AMI
    - CGW
    - DHCP Option
    - EBS Snapshot
    - EBS Volume
    - Instance
    - IGW
    - nACL
    - route table
    - SGs
    - Subnet
    - VPG
    - VPC
    - VPC peering connection
* **Use with IAM to create conditions in your policy that allow access only to specific tags**

AZ vs Edge Location

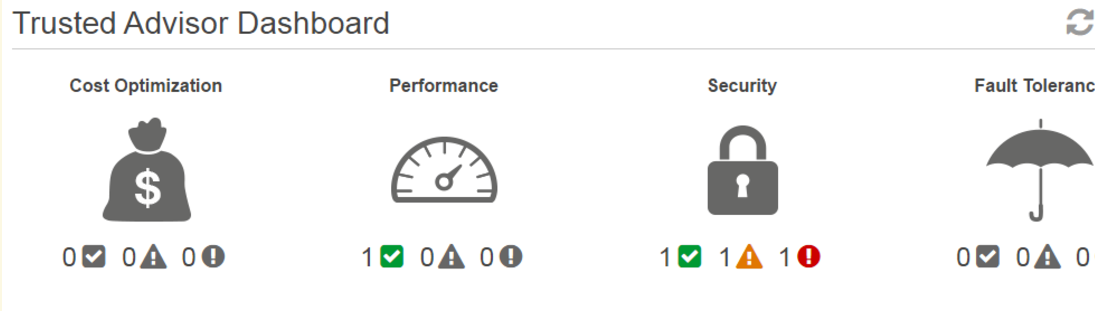
* AZ – AMZN resource within an AWS region
* Edge Loc – will deliver cached content to the closest location to user, to reduce latency

Web Tier vs DB tier

* Web tier refers to EC2 instances
* DB – RDS

Trusted Advisor

* **Can check against service limits\*\***



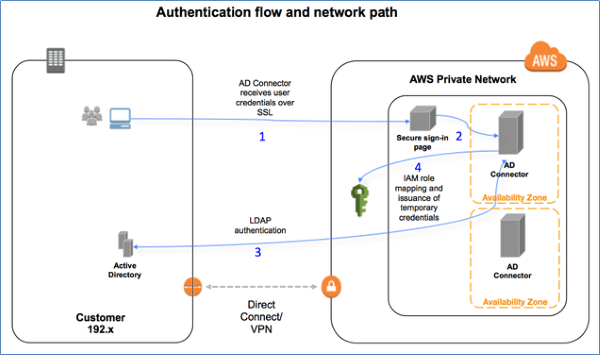
**IAM**

* Principals (IAM entities)
  + Root User
    - Primary account user; has full privileges
  + IAM Users
    - Identities set up in IAM
    - When created, **has NEITHER access key or password**
    - URL given to IAM user to log into the console
      * https://**aws-account-id-or-alias**.signin.aws.amazon.com/console
  + Roles/Temp Security tokens
    - **Roles**
      * **Groups of policies you can apply to people**
      * **ROLES ARE GLOBAL**
      * Grant specific privileges to specific actors **for a specific time period**
      * When an internal/external actor assumes role, AWS provides temp security token from STS (Security Token Service) which lasts anywhere from **15mins – 36 hours**
    - **Use cases**
      * EC2 roles – grant permissions to apps on EC2 Server
      * Cross-account access
      * Federation – granting permissions to users authenticated by external system
        + **OIDC (OpenID Connect)**

Fb, google, login w/amazon etc

* + - * + **Security Assertion Markup Language (SAML 2.0)**

Used with active directory of LDAP



* Authentication Means
  + Username/Password – CONSOLE ONLY
  + Access Key (Access key ID and Secret access key)
    - Used for programmatic access (API or SDK)
  + Access Key/Session Token
* Authorization
  + Attach policies to users/groups/roles
  + **Policy = JSON to define access permissions**
    - **Effect** – allow or deny
    - **Service** – which service am I controlling access to
    - **Resource** – which specific resource w/in that service is being used
      1. ARN (amazon resource name)
         * “Arn:aws:service:region:account-id:[resourcetype:]resource”
    - **Action**
      1. List of actions that can be taken (e.g. s3:GetObject…)
    - **Condition** (optional) – any other limitation like limiting to a specific IP
* Important Features
  + MFA
  + Rotating Keys
    - **Allows two access keys at a time**
      1. Create new access key for user
      2. Reconfig all apps with new key
      3. Disable original key
      4. Test new key
      5. Delete original key
  + Resolving Multiple Permissions
    - Initially request denied by default
    - Policies evaluated. If there is explicit ‘deny,’ evaluation stops and request denied.
    - if there is no deny && there is an explicit allow, the request is allowed
    - If no explicit deny && no explicit allow => default deny

**S3**

Links:

* <http://docs.aws.amazon.com/AmazonS3/latest/API/RESTCommonRequestHeaders.html>
* <http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingBucket.html>

Limits/Properties

* **Object based storage**
* **100 buckets/acct**
* **File size 0 bytes – 5TB**
* **Buckets are private by default**
* **Bucket names universal**
* **Data is NOT encrypted by default, but SSE can be enabled**
* **Can use multi-object delete to delete large numbers of objects in S3**
* **THERE IS NO COST TO TRANSFER FILES FROM s3 and ec2 if theyre in the same region**

Buckets:

* Accessing Bucket
  + URLs
    - http://**bucket**.s3*-aws-region*.amazonaws.com
    - <http://s3-aws-region.amazonaws.com/bucket>
  + Website urls
    - <http://bucket.s3-website-aws-region.amazonaws.com>
    - **Bucket Name must be the same as the domain name when using route53**
* Naming conventions
  + Between 3 – 63 characters long
  + **ONLY use lowercase letters, numbers, periods, and dashes**
  + Must start with lowercase letter or number
  + No '\_’ , cannot end in dash, have consecutive periods, or use dashes next to periods
  + **CANNOT be formatted like IP address**
* **Sub-resources**
  + ***Location* – region s3 bucket is in. one region per bucket**
  + ***policy* and *ACL* (Access Control List) – ways to secure bucket**
    - **Bucket Policy**
      * **Components**
        + **Resources – buckets and objects, identify with ARN**
        + **Actions – operations you allow or deny**
        + **Effect – allow or deny**
        + **Principal – account or user who is allowed access**
  + *cors* (cross-origin resource sharing)
  + ***website* - boolean, is website configured for static website hosting.**
  + ***Logging*** – Boolean is logging enabled?
  + event notification -You can enable your bucket to send you notifications of specified bucket events.
  + ***Versioning* – Boolean, is versioning enabled?**
    - **Any object stored in the bucket before versioning is enabled will have a version ID of *null***
  + ***Lifecycle* – content lifecycle policy (JSON?)**
  + ***cross-region replication* - Boolean**
  + *tagging*
  + *requestPayment*
  + ***transfer acceleration***
    - Transfer Acceleration enables transfers of files over long distances between your client and an S3 bucket taking advantage of Amazon CloudFront’s globally distributed edge locations.
    - Upload to transfer acceleration endpoint (edge location) which will then upload to original bucket

**Consistency**

* **Read after Write consistency for PUTS of new Objects**
* **Eventual consistency for DELETES and overwrite PUTS**

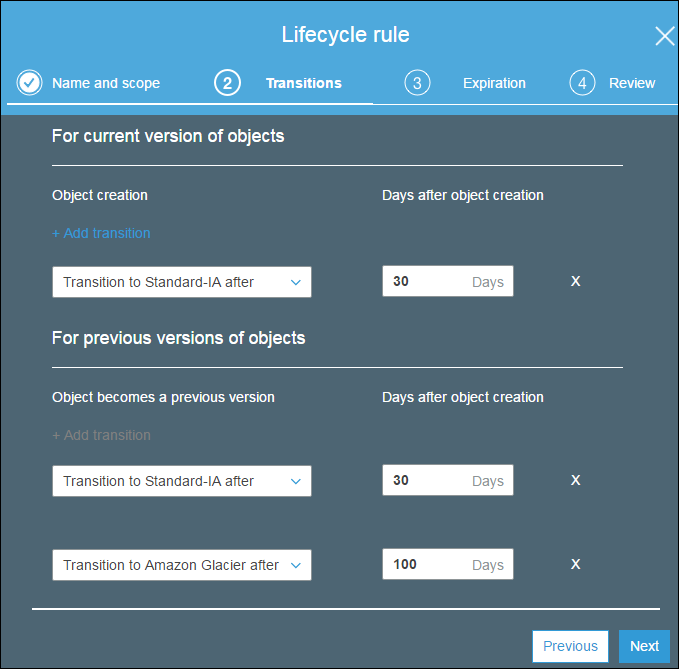
Encryption

* In Transit
  + SSL/TLS
* At rest
  + Server side encryption
    - SSE-S3 : S3 managed keys
      * Everything handled by amazon
    - SSE-KMS
      * Added protection with envelope key, and audit trail, have option to manage key yourself
    - SSE-C
      * You, not amazon, manage the key
  + Client Side Encryption
    - Encrypt data before uploading

**Storage Gateway**

* File interface
  + NFS protocol
* Volume Interface – iSCSI based
  + Gateway Cached Volumes
    - **S3 primary data store, freq accessed data cached on prem**
    - Each volume up to 32TB in size; 32 volumes supported, up to 1PB
  + Gateway stored Volumes
    - **Cache data to s3 asynchronously , main on prem**
    - Each volume up to 16TB; 32 volumes supported, up to 512 TB
* Tape Interface
  + Limitless collection of virtual tapes
  + Virtual Tape Library vs Virtual tape shelf
    - **Library – immediate access—max 1500 virtual tapes (1PB)**
    - **Shelf – based on glacier, up to 24 hour retrieval**
* Networking requirements
  + Open port 443 on firewalls
  + Internally allow port 80, 3260, UDP 53

Life cycle policies



Multi-part upload

|  |  |
| --- | --- |
| **Item** | **Specification** |
| **Maximum object size** | **5 TB** |
| Maximum number of parts per upload | 10,000 |
| Part numbers | 1 to 10,000 (inclusive) |
| **Part size** | **5 MB to 5 GB, last part can be < 5 MB** |
| Maximum number of parts returned for a list parts request | 1000 |
| Maximum number of multipart uploads returned in a list multipart uploads request | 1000 |

Charged for:

* Storage
* Requests
* Storage management pricing
* Data transfer pricing
* Transfer acceleration
  + uses cloudfront edge locs to route data to s3 over optimized network path for faster file transfer

**CORS**

* **allow static websites in s3 to access resources in a separate bucket**
* **enable on resources bucket and state URL for origin that will be calling the bucket**
* **in order to access file that successfully loads you must use the origin URL (in this case the website URL that we added to the sample CORS policy**

Glacier

* **Can download up to 10GB for free**
* **Data is encrypted by default**
* Archives are **IMMUTABLE**

Events S3 can use to trigger msgs (SNS mainly)

* New object
* Object deletion
* RRS object

API Reference:

* Common ***Request*** Headers
  + Authorization
  + Content-Length
  + **Content-Type**
  + Content-MD5
  + **Date**
  + Expect (Valid Values: 100-continue)
  + **Host**
  + x-amz-content-sha256
  + x-amz-date
  + x-amz-security-token
* Common ***Response*** Headers
  + Content-Length
  + Content-Type
  + Connection
  + Date
  + ETag
  + Server
  + x-amz-delete-marker
  + x-amz-id-2
  + x-amz-request-id
  + x-amz-version-id

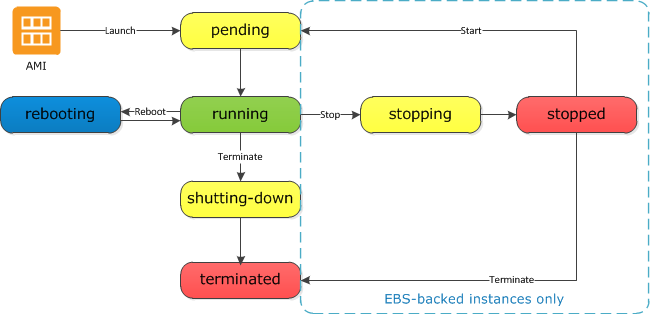
**EC2/EBS:**

Links:

* <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-instance-addressing.html#public-ip-addresses>
* <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/dedicated-instance.html>
* <https://aws.amazon.com/blogs/aws/new-ec2-spot-instance-termination-notices/>
* <http://docs.aws.amazon.com/autoscaling/latest/userguide/as-instance-termination.html#custom-termination-policy> 🡪might wanna look at this again

random notes:

* AZ names unique per acct (my us-east-1 might not be the same location as ganesh’s us-east-1)
* **Improve website load times**
  + **Cache some media with elasticache**
* **Limit 20 EC2 instances running per region, across the instance family**
* **Limit 20 load balancers/region**
* **Instance Termination**
  + Underlying host is changed
  + All *instance store* data is lost
* To attach EBS volume to EC2 instance they must be in the **\*same AZ\***
* Place to input bashscript is called “User Data”
* **Instance store backed volumes CANNOT be restarted or stopped**
* **AMI Stuff (here for now)**
  + Golden Image = AMI constructed from customized image



* + AMI’s can be made public by modifying permissions
* **Public data sets provided by EC2 are free**
* Elastic Network Interface (ENI)
  + Set of attributes that you can attach and re-attach to different instances to carry those attributes
    - A primary private IPv4 address
    - One or more secondary private IPv4 addresses
    - One Elastic IP address (IPv4) per private IPv4 address
    - One public IPv4 address
    - One or more IPv6 addresses
    - One or more security groups
    - A MAC address
    - A source/destination check flag
    - A description
  + Types of attachment
    - **Hot** – attach when instance is **running**
    - **Warm** – attach when its **stopped**
    - **Cold** – when instance is **being launched**

Pricing models

* On-Demand
* Reserved
  + Modifying **Reserved instances** – Can modify the following attributes:
    - AZ
    - Scope
    - Network platform
    - **Instance size (must be same instance type)**
  + When selling reserved instances that are no longer needed on the marketplace, instances are grouped according to:
    - **Duration of the term remaining**
    - **Hourly price**
      * AS A RESULT: BEFORE SELLING, **TERMINATE** (vs. Stopping) THE INSTANCES IN ORDER TO MAXIMIZE THE REMIANING TERM
  + Lowers cost of ownership (1 or 3 year deals and discounted hourly rate)
* Spot
  + “Spot Instance Termination Notice” – **two minute** spot termination warning
  + Good for batch processing and workloads that can be interrupted

IP addressing

* Private IPv4
  + If not specified for an instance being launched, chosen by AWS
  + Instances also given internal DNS hostname that resolves to private IP, by default
* Public IPv4
  + assigned to your instance from Amazon's pool of public IPv4 addresses
  + not associated with your AWS account.
    - When address is disassociated from your instance, it is released back into the public IPv4 address pool, and you cannot reuse it
    - **When does AWS release my IP?**
      * When instance is **stopped** or **terminated**
      * When you associate instance with an Elastic IP
* Elastic IPs
  + http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html
  + Address allocated to your account
  + Can associate to and from instances
  + When am I charged for an EIP
    - If EIP is created and NOT allocated to any instance
    - If EIP is attached to a stopped instance

API Commands

* Revoke-security-group-ingress
  + Remove one or more rules from SG

Tenancy:

* **Dedicated instances** - physically isolated at host hardware level, may share hardware with other instances
* **Dedicated host** - physical server dedicated for your use

**Attributes:**

Value Description

default Your instance runs on shared hardware.

dedicated Your instance runs on single-tenant hardware.

host Your instance runs on a Dedicated Host, which is an isolated server with configurations that you can control.

**\*\*"You cannot change the tenancy of a default instance after you've launched it. You can change the tenancy of an instance from dedicated to host after you've launched it, and vice versa.\*\***

**EBS Volumes**

* **EBS volume CANNOT *always* tolerate an AZ failure** 
  + volume replicated to physical hardware w/in AZ
* When Instance is terminated
  + root volume DOES get deleted
  + snapshots maintained
  + other attached volumes (NOT ROOT), are preserved
* **Existing unencrypted instance CANNOT be encrypted**
  + To encrypt: create and mount new EBS volume and copy data
    - Create new instance after creating a snapshot of existing instance and further making an encrypted copy, and restoring from that encrypted snapshot
      * Unencrypted instances CANNOT create encrypted snapshots, instead use copy creation process describes above
* **EBS volumes can still be used while snapshot creation is in progress**
* **You can change EBS volume size and it can be changed when it is still attached to an instance**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Volume Type** | **General Purpose SSD (gp2)\*** | **Provisioned IOPS SSD (io1)** | **Throughput Optimized HDD (st1)** | **Cold HDD (sc1)** |
| **API Name** | gp2 | io1 | st1 | sc1 |
| **Volume Size** | * 1 GiB - 16 TiB | * 4 GiB - 16 TiB | * 500 GiB - 16 TiB | * 500 GiB - 16 TiB |
| **Max. IOPS\*\*/Volume** | 10,000 | 20,000 | 500 | 250 |
| **Max. Throughput/Volume†** | 160 MiB/s | 320 MiB/s | 500 MiB/s | 250 MiB/s |
| **Max. IOPS/Instance** | 75,000 | 75,000 | 75,000 | 75,000 |
| **Max. Throughput/Instance** | 1,750 MB/s | 1,750 MB/s | 1,750 MB/s | 1,750 MB/s |
| **Dominant Performance Attribute** | IOPS | IOPS | MiB/s | MiB/s |
|  |  |  |  |  |

* **EBS ONLY Instance Families**
  + T2
  + M4
  + C4
  + R4
  + P2
  + G3

Instance Store backed volumes

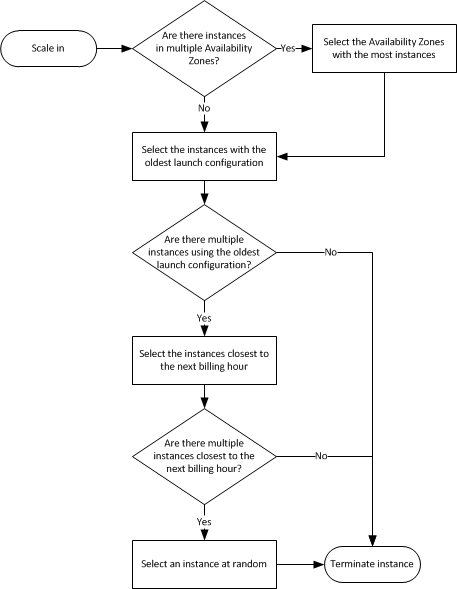
* Limit 10GB
* Cannot be stopped/restarted
* Data is lost if instance goes down
* Attributes are fixed for life of the instance

**Elastic Load Balancing**

* Types of load balancers (also application layer 7, and classic (transport layer) 4)
  + **Internet Facing load balancer**
    - Takes requests from internet
  + **Internal load balancer**
    - Route traffic to instances in a private VPC
  + HTTPS load balancer
    - Encrypts data b/w clients that initialize HTTPS and load balancer
* **You are never given IP for an ELB, only DNS name**
* **Listeners** – configured to process connection requests with port and protocol
  + Supported protocols
    - http/s
    - tcp
    - ssl
* **Health Checks**
  + Report up-time status of instances (InService or OutOfService)
  + Configuration
    - Ping protocol (default HTTP)
    - Ping port
    - Ping path – file path to the file you want to be used for the health check
    - Advanced details
      * Response timeout
        + Time to wait when receiving response from healthcheck
      * Interval
        + Time between health checks (5-300 secs)
      * Unhealthy threshold
        + How many consecutive health check failures before instance is deemed out of service
      * Healthy threshold
        + Consecutive passes required to be considered healthy again=
* ELB access logs
  + Load balancers provide ability to enable access logs which contain:
    - time access request was received
    - client IP address
    - latencies… etc

**Auto Scaling**

* (generally for web applications)
* Can be configured to send notifications via SNS
* **Can only add existing instance to an autoscaling group (target group) if**
  + AMI used for this instance still exists
  + Instance is NOT a member of another Auto Scaling group
  + Instance is in SAME AZ as the autoscaling group
* Instance Termination policies for ‘scaling-in’
  + Default

(good example of BMPN)\*\*\*

* + Customize
    - Can customize policy to respond to different thresholds being crossed, triggered ultimately by CloudWatch
  + Auto-scaling plans
    - Maintain Current Instance Levels
    - Manual Scaling
    - Scheduled Scaling
    - Dynamic Scaling
* Configuration
  + **Connection draining**
    - **Specify max time for load balancer to keep connections alive before reporting instance as “de-registered” (1-3600 seconds)**
    - **Enables completion of ‘in-flight’ (existing) requests**
  + Proxy Protocol
    - Enable to add human-readable header to request headers with connection info like IP addresses and ports
  + Sticky Sessions
    - Allows load balancer to bind users session to a specific instance
  + Health Checks
    - Ping, connection attempt, or page load to check if ec2 instance is up
* Launch configurations
  + Template from which Auto-scaling group will create new EC2 instances from
  + **what you need to create one: Name, AMI, instance type (ALL ELSE IS OPTIONAL)**

**RDS**

* Types of DB systems
  + **OLTP (online transaction processing)**
    - Transaction oriented applications
    - Frequently writing and changing data
      * Ex) e-commerce, data entry
    - Can be updated thousands of times per second
    - **Primary Service : RDS**
  + **OLAP (online analytics processing)**
    - Data warehouses
    - Reporting/analyzing big data sets
    - Updated in scheduled batches
    - **Primary Service : Redshift**
* Offered DB Engines
  + Licensing (MSFT, Oracle)
    - License Included
    - BYOL

|  |  |  |  |
| --- | --- | --- | --- |
| Engine | Notes | Multi-AZ (Y/N) | Read Replicas (Y/N) |
| MySQL | NumReplicas: 5 | Y | Y |
| Oracle |  | Y | N |
| PostgreSQL | NumReplicas: 5 | Y | Y |
| MS SQL Server |  | Y | N |
| MariaDB | More scalable than MySQL  NumReplicas: 5 | Y | Y |
| Aurora (AMZN) | * MySQL Compatible * DB Clusters   + Primary Instance     - Both read and write access, you modify this when modifying any data   + Aurora Replica     - Secondary, read-only instance     - **Upto 15 replicas/cluster** | Y | Y |

* Storage options
  + **Magnetic**
    - **Reg Magnetic (CAN BE BOOT DRIVE)**
    - Cold storage
    - Provisioned throughput
  + **General purpose SSD**
    - **Up to 10,000 IOPS**
    - **3 IOPS/GB**
    - Can throttle in peak times with IOPS credits from previously unused IOPS
  + **Provisioned IOPS SSD**
    - **10,000+ IOPS; up to 20,000**
    - **50 IOPS/gb ratio when provisionin**
  + **Only certain instance classes support encryption**
* Automated Backups
  + **One day of backups retained by default, can modify retention period up to 35 days**
  + **When you delete a DB instance, all automatic backups are deleted** (manual snapshots retained, [in console or with CreateDBSnapshot command])
  + **Occur daily in 30-minute maintenance window**
* Failover (multi-AZ)
  + When does it happen?
    - Loss of availability in primary AZ
    - Loss of network connectivity to primary DB
    - Compute unit failure on primary DB
    - Storage failure on primary DB
  + DNS name stays the same, amazon just changes CNAME to point to the standby
* Scalability
  + Vertical – change instance class (CPU, memory), or increased storage
    - ModifyDBInstance or schedule change in next maintenance window
    - Storage expansion supported for all DB engines except SQL Server
  + Horizontal
* Other Notes
  + **Limit 40 RDS DB instances per customer**
    - Limit 10 for Oracle or MS SQL Server
  + Min DB instance: db.t2.micro
  + **Max DB instance: db.r38xlarge, 244GB memory**
  + **DB Parameter Group**
    - Container for configuration values that can be applied to multiple instances
    - **Can change for existing instance; reboot required**
  + **DB Option Group**
    - Container for engine features (empty by default)
  + **CANNOT access underlying OS in RDS instance (e.g. CANNOT SSH)**
  + **Read Replicas = Asynchronous replication (boots RR w/snapshot of existing instance)**
    - Can created RRs across multiple regions
  + **Multi-AZ = Synchronous replication (automatically copies to backup instance in another AZ)**
    - **Multi AZ deployments CANNOT use backup for read AND write operations**
  + Linux tool = EC2 rescue
  + **When deploying RDS instance into VPC you must have at least 1 subnet in at least 2 of the AZs (I assume this is because of the automatic backup)**

**EFS**

* **Features**
  + Supports NFS v4 (network file system)
    - Supports thousands of concurrent NFS connections
  + Only pay for storage you need; no need to pre-provision
    - $0.30/Gb
  + Can scale to petabytes
  + Data stored across multiple AZs w/in a region
  + **BLOCK BASED STORAGE**
* Demo notes
  + Create FS and chose a specified VPC and which subnets (AZs; 1 subnet = 1 az) you want it to be available in
  + **When attaching EC2 instances, make sure they are in the SAME SG as the EFS**

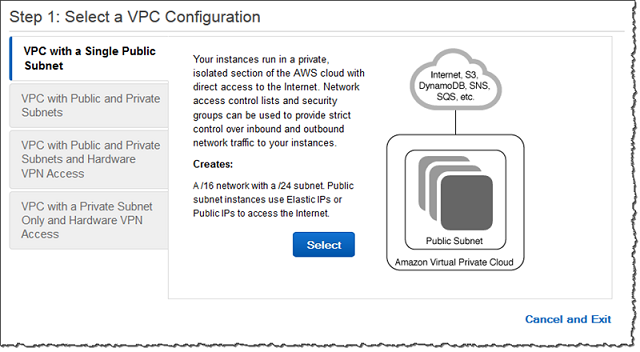
**Route53 DNS**

* Record Types
  + A record (address record)
  + AAAA (IPv6)
  + **CNAME or alias**
    - **CANNOT create record at ‘zone-apex’ lvl**
    - Example: M.amazon.com
    - **You will be charged for these requests**
  + **Alias record is an A record, that *acts like* a CNAME**
    - **When creating alias record, you CANNOT set TTL for records as R53 will use the TTL for the associated AWS resource**
    - **Not charged**
  + TTL record
    - Length that a DNS record is cached on the resolving server or users local PC
    - Lower TTL = faster changes to DNS record propagation
  + MX – mail exchange record
  + **NS – name server**
  + **SOA**
  + PTR
  + SRV
  + TXT
* Routing Policies
  + **Simple**
  + **Latency Based**
    - Chose based on which instance will give the least latency
  + **Geo-location**
    - Route based on location of user
  + **Weighted**
    - Choose % to each of two
    - Commonly used when testing a new environment
      * Route smaller % to new environment
  + **Failover**
    - Primary in one region
    - Secondary set up in another and will auto failover if prim goes down
* DNS Resolution steps
  + My browser asks resolving server what the IP address is for yomama.com
  + Is there a cached IP address?
    - Yes – access that IP
    - No – ask route server managed by registrar managed by ICANN

**VPC:**

Links:

* <http://docs.aws.amazon.com/AmazonVPC/latest/PeeringGuide/invalid-peering-configurations.html>
* <http://docs.aws.amazon.com/AmazonVPC/latest/PeeringGuide/vpc-peering-basics.html#vpc-peering-limitations>
  + <http://docs.aws.amazon.com/AmazonVPC/latest/PeeringGuide/invalid-peering-configurations.html>



Limits/Properties

|  |  |
| --- | --- |
| Resource | Default limit |
| VPC/region | **5** |
| Subnet/VPC | **200** |
| IGW/region | 5 (only 1 attached per VPC) |
| VPG/region | 5 (only 1 attached per VPC) |
| CGW/region | 50 |
| VPN connections/Region | 50 |
| Route tables/VPC | 200 (50 for non-propagated) |
| Elastic IPs/region | **5** |
| Security Groups/VPC | **500** |
| Network ACLs/VPC | **200** |
| Rule/nACL | **20** |
| VPC endpoints/region | 20 |

Peering:

* Invalid setups/Limitations – **CANNOT** set up VPCs that:
  + Have overlapping CIDR blocks
  + Have transitive peering
    - (A 🡪 B) + (B 🡪 C) **!=** (A 🡪C)
  + Edge to edge routing through gateway or private connection (DirectConnect)
    - CANNOT extend peering to:
      * VPN connection or AWS Direct connection to corporate network
      * Internet connection through IGW
      * Internet connection in priv subnet through NAT
      * VPC endpoint to aws service
  + **Are in different regions**

**VPN Connections**


            Multiple VPN layout
          

Tenancy:

"Each VPC has a related instance tenancy attribute. **You can't change the instance tenancy of a VPC after you create it**. This attribute has the following values."

**Attributes:**

Value Description

Default An instance launched into the VPC runs on shared hardware by default, unless you explicitly specify a different tenancy during instance launch.

dedicated An instance launched into the VPC is a Dedicated instance by default, unless you explicitly specify a tenancy of host during instance launch. You cannot specify a tenancy of default during instance launch.

Security

* SG
  + Stateful – return traffic automatically added with 1 way rule
* nACL
  + Stateless – return traffic must be explicitly allowed
  + Default nACL is default ALLOW all; when you make your own nACL, its default DENY all
  + Rules evaluated in number order, starting with lowest numbered rule
  + **Subnet can only be associated with ONE nACL**
  + Rule components
    - Number
    - Protocol
    - Source/destination CIDR range (inbound rules/outbound rules, respectively)
    - ALLOW or DENY

**DynamoDB:**

Links:

-https://aws.amazon.com/blogs/aws/optimizing-provisioned-throughput-in-amazon-dynamodb/

-http://docs.amazonwebservices.com/amazondynamodb/latest/developerguide/BestPractices.html

General notes:

* **ALL WRITE REQUESTS ARE APPLIED IN THE ORDER IN WHICH THEY WERE RECEIVED**
* “DynamoDB offers scalable throughput and storage by **horizontally partitioning your data across a sufficient number of servers** to meet your needs.”
* AWS strategies to ensure performance/uptime
  + **Store data on SSDs**
  + **Spread across 3 geographically distinct areas (you cannot chose where your db is deployed)**
  + DB is partitioned across multiple partitions
* **Limit of 1 byte - 400KB per item**
* Can access DB with HTTP/S and API accepts JSON
* Supported Data Types:
  + String, Number, Binary, Boolean, sets (num, str, binary), hetero list, hetero map
  + **INDEXABLE: String, Number, Binary, Boolean**
* Supports key-value and document store (document store = JSON, XML, HTML)
* **MUST contact amazon if you want to exceed throughput rates of 10,000 writes or reads / sec**

**Use cases for storage**

* **Store METADATA of BLOB**
* **User state information**
* **JSON documents**
* **Managing web sessions**
* **real time tabulations**
  + (collecting many live votes)

Consistency Models

* Eventual consistent reads (default)
  + Consistency across all copies of data (on all 3 of the distinct locations dyanmodb is hosted on), is **usually reached within a second**
  + **Best read performance\***
* Strongly Consistent reads
  + Returns result that reflects all writes that received a successful response prior to the read (if data needs to be data synced)

The Basics

* Tables - collections of JSON items
* Items – collections of key value pairs
* Attributes – key value pairs
  + **One will be the primary key**

Pricing

* **Read units are rounded to 4KB while Write items are rounded to 1KB**
* Provision throughput capacity (calculate price with read and write capacity units)
  + **Free Tier gives: 25 read capacity unit and 25 write capacity units / month**
  + **Cost:**
    - **Write Throughput Capacity: $.0065/hour for every 10 units (writes per second)**
    - **Read Throughput: $.0065/hour for every 50 units (Reads per second)**
* Storage
  + **Free Tier gives: 25 GB storage / month**
  + **Cost**
    - **$.25/gb/month**
* **EXAMPLE**
  + Assumptions
    - App needs 1,000,000 reads and 1,000,000 writes/day
    - Stores 28 GB data
  + Calculations
    - Writes and reads per second = 1,000,000 **\*** **\*** **\***
      * **11.56 round up to 12**
    - **Total throughput cost**
      * Write capacity cost per day
        + ((.0065)/10units )\* 12 writecapunits \* 24 hrs = **$.1872/day**
      * Read capacity cost per day
        + ((.0065)/50units )\* 12 readcapunits \* 24 hrs = **$.0374/day**
    - **Total Storage Cost/month**
      * **25 GB /28 GB free, pay for 3GB/month storage**
      * **.25\*3gb = $.75/ month**
    - **TOTAL = $7.49/month**

**Keys and Indexes**

* Two types of **Primary Keys**
  + **Single Attribute (unique id)**
    - Called the **partition key (hash key)**, composed of 1 attribute
    - Partition key value used as input in internal hash function, output partition (physical location where data is stored)
    - With this pk, no to items in the table can have the same partition key
  + **Composite**
    - **Partition key & sort key (hash and range)**, 2 attributes
    - Partition key also used as input in hash function, output partition
    - Two items can have the same partition key, but **MUST HAVE DIFFERENT SORT KEY**
    - All items with the same partition key are stored together, sorted by sort key value
* **Indexes**
  + Local Secondary Index
    - Same partition key, different sort key
    - **Can ONLY be created when first creating the table, and CANNOT be removed or modified**
  + Global Secondary Index
    - Has DIFFERENT partition key, and different sort key
    - Can be created at table creation or added later

Streams

* Used to capture any kind of modification to dynamodb tables
  + If new item is added, stream captures image for item and attributes
  + If item is updated, stream captures before and after image
  + If item is deleted, stream captures copy of item
* Holds data for up to 24 hours

Operations

* **Query** 
  + Finds item in a table only using pk attribute values (partition attribute name, and value)
    - Optional sort key attribute name and value
  + By default returns all data attributes for items with specified pks
    - Can us **ProjectionExpression** parameter to get query to return only some of attributes
  + Results sorted by sort key (if there is one)
    - Results come in numeric/asci order for numbers/strings respectively
    - To reverse the order (flip from ascending to descending and vice versa), set **ScanIndexForward** param to **false**
* Scan
  + Examines every item in the table, by default all attributes but can use ProjectionExpression
  + You’ll probably rarely ever use this
  + **Limit of 1 MB per operation** (only returns results upto 1MB)

Calculate throughput units

* Read Throughput
  + Givens:
    - KB/item
    - number of items
  + Round KB/sec to nearest multiple of 4, divide by 4KB, multiply by number of items
  + For strongly consistent reads this is your answer, divide by 2 for eventual consistency
* Write throughput
  + Givens:
    - Number of items
    - KB per item each second
  + Multiply the two
* Response when you reach/pass read/write capacity
  + 400 HTTP Status code error **ProvisionedThroughputExceededException**
  + Exceeded maximum allowed provisioned throughput for a table or one or more GLOBAL secondary indexes

Web Identity Providers

* Process
  1. User authenticates with ID provider
  2. ID Provider passes a Token to the user
  3. User code calls AssumeRoleWithWebIdentity API Request, provides providers token and specifies ARN for the IAM role that will be assumed
  4. STS will send back a temp credentials; app can now access DynamoDB from 15 mins – 1 hour (default = 1 hr)
* Implementation
  + Each table will allow you to generate policies based on your selected provider and requested action
  + Create new policy with generated code, and attach it to the new role

Conditional Writes

* Only write to the DB if a certain condition about the attributes of an item, is met
* They are ‘**idempotent**’
  + You can send same conditional write request multiple times, but it will have no effect on an item after the first (if two users want to update ID:1’s price only if price is $10 dollars, only first update will be made)
* Define ConditionExpression
  + Boolean functions: ATTRIBUTE\_EXIST, CONTAINS, and BEGINS\_WITH
  + Comparison operators: =, <>, <, >, <=, >=, BETWEEN, and IN
  + Logical operators: NOT, AND, and OR.

Atomic Counters

* Use **UpdateItem** operation to increment or decrement values without interfering with other write requests (essentially, attribute++ or -- )(remember writes are applied in order they are received)
* These are **NOT indempotent** (each request WILL increment the value)
  + Only use this if you can have a margin of error with the data, if you must have it 100% accurate (e.g. banking), use conditional writes

Operations

* **BatchGetItem**
  + **can retrieve up to 16MB in data, up to 100 items**
  + Single request can pull from multiple tables
* GetItem
* PutItem
* BatchWriteItem
* **UpdateTable - (does not consume capacity units)**
  + **Modifies throughput settings, global secondary indexes, and DynamoDB streams settings for a given table**
  + **While updating you cannot issue another UpdateTable reques**

LIMITS

|  |  |  |
| --- | --- | --- |
| **ITEM** | **LIMIT** | **Limit Increase? How?** |
| **Tables/region** | **256** | **Yes, request from AWS** |
| **Global secondary indexes/ table** | **5** | **NO** |
| **local secondary indexes/ table** | **5** | **NO** |
| **Item size** | **400KB (1byte min)** |  |
| **Size of item collection** | **10GB** |  |
| **SMALLEST reserve capacity** | **100 cap units** |  |
| **Attributes/item** | **NO LIMIT** |  |

Data Spread and partitions TIPS:

* **“When storing time series data in DynamoDB**, it is recommended that you **spread your data across multiple tables – one per time period (month, day, etc).”**
* “The provisioned throughput associated with a table is also divided evenly among the partitions, with no sharing of provisioned throughput across partitions. Consequently, to achieve the full amount of request throughput you have provisioned for a table**, keep your workload spread evenly across the hash key values. Distributing requests across hash key values distributes the requests across partitions**.”
* Use **Pagination** (limit number of results per query)
* Separate Hot and Cold Data
  + **Hot data** = accessed frequently (e.g. recent replies in the example forums application)
  + **Cold data** = accessed infrequently or never (e.g. replies from several months ago)

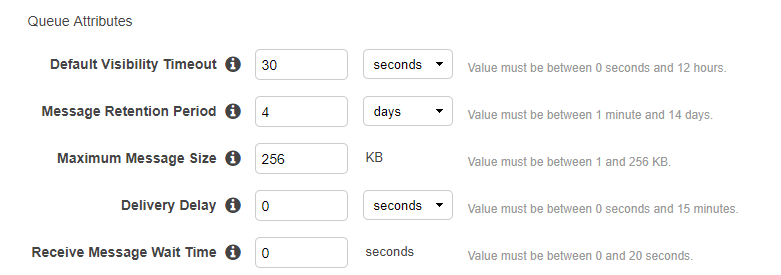
**CloudWatch**

* Properties
  + Functionality
    - Collect/track metrics
    - Send notifications
    - Respond to threshold changes
      * can automatically stop, terminate, reboot or recover EC2 instances
    - **Receives/aggregates metrics every 1 minute by default**
  + Monitoring intervals
    - Basic = 5 minutes
    - Detailed = 1 minute
  + Retention Schedules
    - **1 minute data points available for 15 days**
    - **5 minute data points available for 63 days**
    - **1 hour datapoints available for 455 days**
* Limits
  + **Max alarms/account = 5,000**
  + **Data retained for 2 weeks by default**
  + **Cannot aggregate data across regions** (but can across AZs)
* Logs
  + Supported formats
    - JSON
    - Any text based common log

**SQS**

* **Properties/Notes**
  + **Used to decouple applications to facilitate horizontal scaling**
  + **No limit to number of queues you can have**
  + **Max size for message = 256 KB of text**
    - To send messages that are larger, use **SQS Extended Client Library for Java**
    - This will contain a reference to the message payload in s3\*\*
  + Visibility Timeout
    - Time message is hidden before being re-released to queue for other instances to pick it up
    - **Max = 12 hours**
    - Can change with the **ChangeMessageVisibility** action
  + Best to use **IAM ROLE** to poll from EC2 instance, **NOT** user credentials
    - Users MUST be granted access to retrieve messages from the queue
  + **To distribute messages to different queues (that might do different things), couple with SNS; subscribe the queues to the SNS topic**
  + Identifying Messages
    - **Identifiers to know\***
      * Queue URLs
        + Must be provided if you want to perform action on queue
      * Message ID
      * Receipt handles
        + when you receive message from queue, response include *receipt handle* which you must provide when deleting the message
  + Configuring anonymous access IS possible.
* Message lifecycle
  + Comp 1 sends message to queue
  + Comp 2 consumes message from queue when ready to process
    - **While message is processed, a copy remains in the queue but is hidden for duration of the visibility timeout**
    - If instance crashes mid-processing, message remains in queue still assigned to same EC2 instances, when instances come back online, *within visibility timeout window*
      * if the instances do not come back online w/in this window, the message will become visible in the queue for other instances to pick up
  + Comp 2 *deletes* message from the queue
* **Queue Types**
  + Standard Queue
    - Nearly unlimited throughout
    - ‘at-least-once delivery’
      * message will definitely be delivered once but can be delivered multiple times
    - Best effort ordering
      * Occasionally messages delivered out of order
  + FIFO Queue
    - Throughput limited to **300 TPS**
    - ‘exactly-once’ processing
      * Duplicate messaged removed
    - First in First out delivery
  + (delay queues)
    - Any queue where DelaySecond attribute > 0 (max 900 secs)
    - **Allows postpone delivery of new messages for a specific num of secs**

**IMPORTANT Default MIN - MAX**



* Long polling
  + **If nothing in the queue, ReceiveMessage function waits for 1-20 seconds, if nothing is returned, call again (avoid constant pinging of queue)**
* **Pricing**
  + **First 1 million SQS requests/month are FREE**
  + **$.50/ 1million requests/month**
  + Single request can have 1-10 messages, but still max payload = 256kb
  + **Each 64KB chunk of payload is billed as one request (full 256kb payload will be billed as 4 requests)**

**SWF and SNS**

* **SWF**
  + Workflow tool with **task based API**
  + **Up to 1 year retention** for tasks
  + **TASK IS ONLY ASSIGNED ONCE, and NEVER duplicated**
  + **Actors**
    - Worker
      * Programs that interact with AWS to get tasks, process them, and return results
    - Decider
      * **Program (not human)** that controls coordination of tasks (ordering, concurrency etc)
  + **Domains**
    - Isolates types, executions, task lists etc
    - Must register the domain
  + **Limits**

|  |  |
| --- | --- |
| **ITEM** | **LIMIT** |
| **Domains** | **100** |
| **SWF task execution time** | **1 year** |
| **Activity types** | **10,000** |
| **Open activity tasks** | **1000** |

* **SNS**
  + **Push based** messaging service
  + **Publishers publish messages to Topics**, which use protocols to send to **subscribers**
  + Subscriptions must be confirmed first, **3 days avail for confirmation**
  + **CANNOT recall message once it is sent**
  + **Max 256 characters for topic title**
  + **Ma 100,000 topics**
  + Supported Protocols
    - HTTP/S
    - SMS
    - Email/**email-JSON**
    - SQS - Does guarantee message delivery to SQS
    - Lambda
    - **Mobile device push notifications**
      * **Example Providers**
        + **GCM (Google Cloud Messaging (android))**
        + **APNS (apple push notification service)**
      * **Steps to set up**
      1. **Register Mobile app with AWS** 
         1. **Provide notification service platform credentials**
      2. **Create endpoint for app/mobile device**
  + **PRICING**
    - **$0.50 / 1 million SNS requests**
    - **$0.06 / 100,000 deliveries over HTTP**
    - **$0.75 / 100 deliveries over SMS**
    - **$2.00 / 100,000 deliveries over email**
  + **NAME, TYPE and VALUE fields must NOT be empty or null.**
  + Components of SNS message
    - **"Type"**
    - **"MessageId"**
    - **"Token"**
    - **"TopicArn"**
    - **“Subject”**
    - **"Message”**
    - **"SubscribeURL”**
    - **"Timestamp"**
    - **"SignatureVersion"**
    - **"Signature”**
    - **"SigningCertURL"**

**RedShift**

**Relational DB designed for OLAP or large data sets \* data warehouse\***

* **Cluster**
  + Leader node and one or more compute nodes
  + Client interacts directly ONLY with the leader node
    - Node Types
      * Dense Compute
        + Up to 326 TB w/SSDs
      * Dense Storage
        + Up to 2 PB w/large *magnetic* disks
  + Nodes will execute parallel query execution (slice CPU into 2-16 pieces, distribute compute evenly over nodes/slices)
  + **Spreads data based on user-defined *distribution strategy***
    - **Can resize at will. When performing this operation, new cluster will be created, DBs will become read only until operation finishes**
    - Uses **columnar storage**

**Caching (CloudFront, Elasticache)**

* **CloudFront**
  + **Distribution Types Supported**
    - S3 buckets – can use s3 as origin
    - **Custom Origin**
      * Any HTTP server (EC2 webserver, on prem webserver etc) **NOT RDS**
  + **You can control how long objects stay in the distribution. Expiration value\***
  + Key Terminology
    - Web distribution (websites)
    - RTMP (media streaming)
  + Exam tips
    - Edge locs are not just read only; you can write to them
    - Objects cached in edge locs for life of TTL
    - Charged for clearing cache
* **Elasticache – In Memory cache**
  + Most commonly used for storing **session data** for webbased apps

**AMAZON CLI/SDKs**

* Used to programmatically access AWS resources
* **Security best practice is to use ROLES to federate permissions to the instances that will interact with AWS instead of using *aws-configure* and storing access key id and secret access key locally**
* Tips
  + When using s3 commands, get in the habit of using **the *- - region* flag** for commands to ensure completion
  + Good practice to use *aws configure* to configure default region
  + Learn to understand the types of words used for different command types\*\*
    - <http://docs.aws.amazon.com/cli/latest/reference/ec2/>
    - **Examples**
      * **associate**-route-table
      * **cancel**-spot-instance-requests
      * **create**-network-acl
      * **delete**-network-acl-entry
      * **describe**-availability-zones
      * **detach**-volume
      * **modify**-instance-attribute
* **Commands you should know**
  + ***Describe-images***
    - Describes all images (AMIs) that are available to us
  + Describe-instances
    - What images do I have running
  + ***Run-instances***
    - **Provision ec2 instance from the command line**
    - aws ec2 run-instances --image-id ami-1a2b3c4d --count 1 --instance-type c3.large --key-name MyKeyPair --security-groups MySecurityGroup
* **Available SDKs**
  + Android, iOD, JS(browser)
  + Java
  + .net
  + Node.js
  + Php
  + Python
  + Ruby
  + Go
  + C++

**Lambda**

* **Event-driven compute service where lambda runs your code in response to events (e.g. changes to date in an s3 bucket or DynamoDB tabl**
* **Calls made to lambda must complete execution in 300 secs**
* Supported languages
  + C#
  + Node.js
  + Python
  + Java
* Which services can TRIGGER lambda events
  + **API gateway**
    - **Anytime a user sends a request to APi gateway which should trigger lambda, it triggers a NEW lambda function (i.e. if two users make the request, two lambda functions will be invoked**
  + AWS IoT
  + **Alexa** Skills kit/ Smart home
  + **CloudFront**
  + **CloudWatch events/ logs**
  + **CodeCommit**
  + Cognito Sync Trigger
  + **DynamoDB**
  + **Kinesis**
  + **S3**
  + **SNS**
* **Pricing based on:**
  + **NUMBER OF REQUESTS**
    - First 1 million requests are free
    - $0.20 per 1 million requests thereafter
  + **DURATION**
    - From time the code begins executing until it returns or otherwise terminates (rounded to the nearest 100ms)
    - Price depends on amt of memory you allocate to your function
      * $0.00001667 for every GB-second used
    - **Function cannot execute over 5 minutes**
* Exam tips
  + Lambda functions are independent; 1 event = 1 function
  + Serverless
  + Lambda functions can trigger other lambda functions, 1 event can = x fxs, if fxs trigger other fxs

**Elastic Beanstalk**

* Stores data in s3
* Can have multiple versions of your apps
* Apps can be split into tiers (web tier/ app tier/ db tier)
* Allows developers to deploy and manage applications in the cloud
  + Devs upload application (code?)
  + beanstalk automatically handles
    - deployment details
    - provisioning
    - load balancing
    - auto-scaling
    - health monitoring
* Have Access to underlying infrastructure
* **Available platforms**
  + **.net**
  + **Java**
  + **Node js**
  + **PHP**
  + **Python**
  + **Ruby**
  + **Tomcat**
  + **Go**
  + **Packer**
* App Deployment policies – how will beanstalk update your applications? **– can do configuration updates (properties of environment) or application updates**
  + All at once
  + Rolling
    - Percentage: X% of fleet at a time
    - Fixed: x instances at a time
  + Rolling with additional batch – before rolling updates, launch a new batch of instances to ensure full capacity during deployment process
  + Immutable – deploy new code with a fresh set of instances
* **Database Configuration (RDS) options**
  + **Let Beanstalk deploy your own RDS instances**
    - **When you delete a beanstalk deployment, RDS instances will be deleted**
  + **Deploy your own RDS instances and attach them to elastic beanstalk**
    - **RDS instances will be maintained when beanstalk deployment is deleted**
* **Elastic beanstalk itself is free to use, but you will pay for provisioned resources**

**CloudFormation**

* **General Notes**
  + **No charge, but you pay for resources that are provisioned**
  + **Can automate deployment**
  + Describe AWS environment with code
  + After resources are deployed, you can modify and update them (in effect version control for AWS infrastructure)
  + by default **“automatic rollback on error”** is **enabled**
    - if there is an error mid-deployment, all resources provisioned before the error will be deleted
    - You are charged for errors
  + Have full root access to resources
  + Stacks can wait for apps to be provisioned using “Wait Condition”
  + Route53 and IAM role creation/assignment are both supported
  + Chef and Puppet both supported
  + **API Calls:**
    - List-stacks = lists all stacks running / deleted
    - Describe-stacks = lists all running stacks
  + **Where can I use intrinsic functions?**
    - resource properties, outputs, metadata attributes, and update policy attributes
* **Components**
  + **Template**
    - Architectural diagram
    - **http://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/template-anatomy.html**
    - JSON or YAML format
    - **Example:**

{

"AWSTemplateFormatVersion" : "version date", //(if not given amazon will assume version from latest template)

"Description" : "JSON string",

"Metadata" : {

template metadata

},

"Parameters" : {

set of parameters

},

"Mappings" : {

set of mappings

},

"Conditions" : {

set of conditions

},

"Transform" : {

set of transforms

},

**"Resources" : {**

**set of resources (Only required field)**

},

"Outputs" : {

set of outputs

}

}

* + **Stack**
    - result of the diagram (what was provisioned)
* **LIMITS**

|  |  |
| --- | --- |
| **ITEM** | **LIMIT** |
| **Stacks/account** | **200 (request higher limit with AWS)** |
| **templates** | **NO LIMIT** |
| **Parameters** | **60** |
| **Outputs** | **60** |

* Intrinsic Functions
  + Fn:GetAtt
    - Returns value of an attribute from a resource in the template
    - Examples:
      * { "Fn::GetAtt" : [ "logicalNameOfResource", "attributeName" ] }
      * "Fn::GetAtt" : [ "myELB" , "DNSName" ]
  + Ref
    - Returns value of specified *parameter*  or *resource*
      * If you specify a parameters logical name, Ref returns value of the parameter
      * If you specify resource logical name, it returns value that you can use to refer to that resource (like physical ID
    - Example

"MyEIP" : {

"Type" : "AWS::EC2::EIP",

"Properties" : {

"InstanceId" : { "Ref" : "MyEC2Instance" }

}

}

If you make an AMI public, this AMI is immediately available across all regions, by default. - false

You can have multiple SSL certificates (for multiple domain names) on a single Elastic Load Balancer. - falseCloud formation and elastic beanstalk resources (allowed)

Acess keys are used for CLI not console

Resources talking within a VPC (ELB, EC2)

Test on Cloud Academy

IAM

Dynamo DB

VPC

Monitoring and debugging

ADD SET dynamo db CLI if\_not\_exists to increment or modify a price (adding a new attribute)

Slow down with S3 versioning enabled

Millions of versions of an object

AWS Sheild (for DDOS attack protection), trusted advisor (managing cost and security recommendation tool)

AWS VPN CloudHUb

-Direct connect

-Cloudformation at run time customize aspects of your template certified

SWF look at task level details status

Trusted advisor

S3 object versioning is it good for security?

Is there a limit on cloudformation templates (NOPE)