|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Stores | | | | | |
| Concepts | | | | Persistent data store: data durable and available after reboot, restarts. – Glacier, RDS  Transient data store: data temporarily stored and passed to another process/persistent store. – SQS, SNS  Ephemeral data store: data lost when stopped. – EC2 Instance Store, Memcached  Input/Output Operations per Second (IOPS): how fast can read and write to a device  Throughput: how much data can be moved at a time  ACID: Atomic, Consistent, Isolated, Durable. BASE: Basically Available, Soft-state, Eventual Consistency | |
| S3 | Object Store. – Max obj size = 5TB. - Largest object in a single PUT = 5GB  - Use multi-part uploads if > 100MB. - S3 eventually consistent. – Versioning (old version still billed)  - Lifecycle management. – Cross-Region Replication  Security: Resource-based (Obj ACL, Bucket policy) & User-based (IAM-policies). – Optional MFA before delete | | | | |
| Storage classes: 1) Standard. 2) Standard-IA (min storage 30 days). 3) One Zone-IA (min store 30 days).  4) Reduced redundancy (frequently accessed, non-critical data). 5) Intelligent-Tiering (min store 30 days)  6) Glacier (min store 90 days, retrieval mins-hrs). 7) Glacier Deep Archive (min store 180 days, retrieval < 12 hrs) | | | | |
| S3 Analytics: - Data Lake: Athena, Redshift Spectrum, QuickSight. – IoT Streaming Data: Kinesis Firehose  - ML and AI Storage: Rekognition, Lex, MXNet. – Storage class analysis: S3 Management Analytics | | | | |
| Encryption at Rest: 1) SSE-S3: use S3 encryption key (AES-256). 2) SSE-C: use own AES-256 key.  3) SSE-KMS: use key generated and managed by AWS KMS. 4) Client-side: encrypt obj before upload to S3 | | | | |
| Transfer Acceleration: speed up data uploads by using CloudFront  Requester Pays: requester rather than bucket owner pays for requests and data transfer  Tags: Assign tags to objs for use in costing, billing, security, …  Events: Trigger notifications to SNS, SQS, Lambda. Static Web Hosting: simple & scalable static website hosting  BitTorrent: use BitTorrent protocol to retrieve any publicly available obj by auto generating a .torrent file | | | | |
| Permissions in S3 | | | | Bucket Policies: defined at bucket resource. Give access to a principal access to all/subset of objects in bucket. – Explicit allow or deny. – Resource based policy takes precedence over identity-based policy  IAM: give temp permissions in the form of a role to a given user/service/app | |
| For cross account: Resource based bucket policy is the same  If bucket policy neither explicit allow or deny: In the S3 bucket account, establish the Permissions & Trust policy (permission policy grant access, trust policy allow entity from outside AWS to assume that role)  Also, in IAM user account, create a role which gives permissions to assume the role that’s been granted to that acct in the trust policy (AssumeRole Permissions) | |
| S3 Gateway endpoint / Elastic Network Interface: allows traffic traverse on AWS private intranet  - Set appropriate route in route table for VPC | |
| Glacier | | | | - Glacier Vault: similar to S3. - Archive: File, zip, tar, … Max size 40TB, Immutable (similar to S3 obj)  - Glacier Vault Lock: policy that enforce rules like no deletes or MFA, Immutable  - IAM: access permissions  For Vault Lock: within 24 hours, can abort or complete vault lock (permanently applied)  - After 24 hours, vault lock aborts | |
| EBS | | | - Only use w EC2. – Tied to single AZ. – Choices for IOPS, Throughput, Cost. – Snapshots. – Not locked to a specific EC2  vs Instance Stores: - Temp. – Ideal for caches, buffers, work areas. – Data remove when EC2 stop or terminate. - Locked to that specific EC2 | | |
| EBS Snapshots: cheap and easy backup strategy. – To share datasets w other users or accounts  - To migrate to a new AZ or region. – To convert unencrypted volume to encrypted one  - Successive snapshots only record the changes from previous snapshot | | |
| Amazon Data Lifecycle Manager: schedule snapshots for volumes or instances every X hours  - Retention rules to remove stale snapshots | | |
| EFS | | Implementation of NFS file share. – Can mount to multiple EC2. – Elastic storage capacity, pay for only what you use (in contrast to EBS where u specify amt of space for volume). – Multi AZ in a given region.  – Configure mount points in ≥ 1 AZ. - Can be mounted on-prem (alternatively, use DataSync to keep storage on-prem in-sync w EFS or S3). - EFS cost > EBS > S3 | | | |
| FSx | File share service, for jobs EFS can’t solve. – Distributed file system. – Has non-Network File System options for file sharing. – Commonly used for Windows file services integration w AWS | | | | |
| 1) FSx for NetApp ONTAP. 2) FSx for OpenZFS. 3) FSx for Windows File Server (Windows, SMB-based file shares).  4) FSx for Lustre (for high-performance distributed apps, HPC clusters, big data & ML)  - Lustre also use S3 to store snapshots and as cold storage for least accessed files | | | | |
| FSx ENI EC2 mount targets. – To grant access, use IAM or managed Microsoft AD  - All drives backing the file system are SSD, so most frequently accessed data can be accessed fastest | | | | |
| Storage Gateway | | Virtual machine that can be run on-prem w VMWare or HyperV or special Dell hardware appliance OR run on EC2. – Provides local storage resources backed by S3 and Glacier  - Often used in disaster recovery to sync to AWS. – Useful in cloud migrations | | | |
| |  |  |  |  | | --- | --- | --- | --- | | New name | Old name (AWS Exam) | Interface | Function | | File Gateway | None | NFS, SMB | On-prem / EC2 to store objs to S3 via NFS/SMB mount points | | Volume Gateway Stored Mode | Gateway-stored Volumes | iSCSI | Async replication of on-prem data to S3 | | Volume Gateway Cached Mode | Gateway-cached Volumes | iSCSI | Primary data stored in S3 w freq access data cached locally on-prem | | Tape Gateway | Gateway-Virtual Tape Library | iSCSI | Virtual media changer and tape library w existing backup software | | | | |
| Has feature = bandwidth throttling. Use Vol Gateway Cached Mode to provide sync files for remote offices | | | |
| DB on EC2 | | Can run any DB w full control and flexibility. – Must manually manage everything like backups, redundancy, patching, scale. – For DB not supported by RDS, like IBM DB2 or SAP HANA | | | |
| RDS | | Managed DB for MySQL, Maria, PostgreSQL, Microsoft SQL Server, Oracle, MySQL-compatible Aurora  For structured, relational data store.- Automated backups, patching in customer-defined maintenance windows. – Push-button Scaling, replication, redundancy | | | |
| Don’t use RDS: 1) many BLOBs: S3. 2) Automated scalability or Name/Value Data structure or data not well structured: DynamoDB. 3) Other DB platforms like IBM DB2 or SAP HANA or Complete Control: EC2 | | | |
| - Multi-AZ RDS (sync replication). – Read-replicas (async replication)  - If master DB die/1 AZ fails, stand-by in another AZ assume role of Master  - If whole region fails, read replica promoted to Stand-Alone (single AZ). Single-AZ reconfigured to multi-AZ | | | |
| Aurora | | Fully managed RDS services for MySQL and PostgreSQL db. – Multi-AZ, auto-scaling, easy multi-region replication. – Up to 15 read replicas. – Main instance write to all data copies, usually available for read within 100 ms. – Cluster endpoint to write -> main instance. – Reader endpoint -> ELB -> read replicas | | | |
| Can set read replicas across regions but replication is asynchronous. – Up to 5 secondary regions | | | |
| Use Aurora standard for predictable, consistent workload. Use Aurora Serverless for unpredictable workload  ACU (Aurora Capacity Units): ≈ 2 GiB of memory, CPU, networking. – Can set max ACU for a cluster from 0.5 to 128, then will auto scale for Aurora Serverless | | | |
| DynamoDB | | | | Managed, multi-AZ NoSQL data store w Cross-Region Replication option. Stores data as name/value pair  Default eventual consistency reads, but can request strongly consistent read via SDK param  - Priced on throughput, rather than compute. – Provision read/write capacity in anticipation  - Autoscale per configured min/max levels. – On-demand capacity for flexible capacity at higher cost  - DynamoDB Transactions: to achieve ACID. – Don’t store data in JSON format | |
| Can do composite primary key = partition key + sort key  1) Global Secondary Index: partition key and sort key can be diff from those on table  - For fast query of attributes outside the PK, w/o having to do table scan  2) Local Secondary Index: same partition key as table but diff sort key  - Already know partition key and want to quickly query on some other attribute  - Cannot create new index on existing table | |
| Sparse Index: index only exists for those records with field. If field don’t exist for record, won’t be included, hence index is smaller  Can replicate table via Secondary Indexes: separate read write | |
| DynamoDB Accelerator (DAX): in-memory cache in front of DynamoDB, to speed up reads | |
| DocumentDB | | | | | AWS-native Document Data Storage: stores NoSQL JSON data in form of document s  - Compatible w MongoDB APIs. – Auto-Scaling. – Writes-then-read consistency  - Retrieve entire document/record at once. – 1 main instance + 15 read replicas in diff AZ/region |
| RedShift | | Fully managed, clustered peta-byte scale data warehouse. – PostgreSQL compatible w JDBC and ODBC drivers. – Parallel processing and columnar data stores. – Redshift Spectrum: query from files on S3 | | | |
| Prepare Data for Analysis | | AWS Glue: - Discover data w Glue Crawlers. – Prepare data w Glue Data Catalogs and Glue ETL Jobs  - Integrate data from diff sources. – Serverless. – Glue Data Quality: predefined rules on data (set alerts)  - Glue Data Catalogs can be used by EMR, RedShift, Athena | | | |
| Athena: serverless big data querying service. – Designed to create DBs from S3 objs. – Use SQL syntax OR run Spark. – Define Athena Federated queries to analyze data from source other than S3 | | | |
| Final consumer can be QuickSight or SageMaker | | | |
| Neptune | | Fully managed graph DB. – Supports open graph APIs for Gremlin, SPARQL | | | |
| Elasticache | | | | Fully managed implementation of in-memory data stores – Redis and Memcached  - Push-button scalability for memory, writes, reads. – Key/value stores, no default persistence  - Billed by node size and hours of use. – Memcached for basic, Redis if need more features | |
| Other DBs | | 1) Athena: SQL Engine overlaid on S3 base on Presto. – Convert to Parquet for performance boost  - Use Athena if data is in S3 w/o need to perform joins w other data sources BUT use RedShift Spectrum if need join S3 w existing RedShift tables or create union products  2) Amazon Quantum Ledger DB (QLDB): based on blockchain. – Provides immutable and transparent journal as a service w/o setup and maintain entire blockchain framework  - Centralized design (oppose to decentralized for most blockchain) allows higher performance & scalability  3) Amazon Managed Blockchain: fully managed blockchain framework supporting open source Hyperledger Fabric and Ethereum  - Distributed consensus-based concepts consisting of network, members (AWS accts), nodes (instances) and potentially apps. – Uses QLDB ordering service to maintain complete history of all transactions  4) Amazon Timestream DB: fully managed DB services for storing and analyzing time-series data  - Alternative to DynamoDB or RedShift and includes built-in analytics like interpolation and smoothing  5) Amazon Elasticsearch/OpenSearch: search engine and document store (not main purpose)  - Service components sometimes referred to as an ELK stack (Elasticsearch, Logstash, Kibana)  6) Amazon MemoryDB: durable, in-memory managed DB for workloads that require ultra-fast, primary DB  - Stores successful write operations durably in a distributed Multi-AZ transaction log  - Can seed MemoryDB cluster w snapshot from existing Redis cluster | | | |
|  | | 1) MSCK REPAIR TABLE: update metadata for partitions added after table was created for Hive  For partitions not compatible w Hive, use ALTER TABLE ADD PARTITION  2) Migrate HPC workload to AWS. Data stored in single S3 bucket, and use EC2 to process data. Multiple apps will modify, overwrite, access data. Requires file locking:  - FSx for Lustre is file system for HPC. Can scale linearly and provision TBs of throughput and millions of IOPS. Supports file locking and can import and export data in S3  3) Store objs in S3. Mk access only access mk prefix. Sa to sa prefix. SM access to mk and sa. Fa all prefix  - Create 1 S3 bucket. Place all objs in bucket. Create 3 S3 access points based on obj prefixes for mk, sa and both. Attach appropriate IAM resource-based policies to S3 access points and S3 bucket. Give appropriate S3 access point aliases to Ma and SM. Give URL for S3 bucket to Fa.  4) S3 bucket created. Need ensure no user delete or overwrite obj for min of 5 years  - S3 versioning can enable for existing bucket. But Object Lock only can enable for new bucket. Set Object Lock retention to compliance mode (governance mode still allow user w permission to remove lock and delete). Move existing obj to new S3 bucket  5) EC2 w EBS. Need run 2 copies of app to be HA:  - Deploy 2nd EC2 in another AZ. Create EFS and configure both EC2 w shared access. EBS Multi-Attach only work if both instances in same AZ  6) EC2 w AMI w unencrypted root volumes. Need shut down when not in use and boot process is slow.  - Create an encrypted snapshot of a root volume. Create a new AMI from encrypted snapshot. Terminate existing instances. Launch new instances by using new AMI w hibernation turned on. Hibernate instances when not in use  7) CloudFront w 2 S3 buckets as origins. 1st S3 should allow access through CloudFront and other AWS resources w appropriate permissions. 2nd S3 should only accessible through CloudFront for all except S3 bucket owner.  - Create separate origin access control (OAC) for CloudFront dist for ea S3. Create S3 bucket policy for first S3 bucket that allows access to appropriate AWS resources and OAC as principals. Create S3 bucket policy for 2nd S3 allowing access to appropriate OAC as principal. OAC = special CloudFront user that is associated w S3 origins to secure all or some S3 content. | | | |

|  |  |  |
| --- | --- | --- |
| Networking | | |
| Concepts | |  |  |  |  | | --- | --- | --- | --- | | Layer | Name | Example | Who handle | | 7 | Application | Web Browser | Customer | | 6 | Presentation | TLS/SSL, Compression | | 5 | Session | Setup, Negotiation, Teardown | | 4 | Transport | TCP, UDP | | 3 | Network | IP, ARP | | 2 | Data Link | MAC | AWS | | 1 | Physical | CAT5, fiber optic, 5GHz | | |
| |  |  |  | | --- | --- | --- | | Protocol | Characteristics | Uses | | TCP | Connection-based, stateful, acknowledges receipt | Web, Email, File Transfer | | UDP | Connectionless, stateless, simple, no retransmission | Streaming media, DNS | | ICMP | Use by network devices to exchange info | traceroute, ping | | |
| Ephemeral Ports: short-lived transport protocol ports used in IP communications. – Dynamic ports  - Above the restricted IP ports (above 1024). – Suggested range is 49152 to 65535 (but Linux kernels use 32568 to 61000, Windows default from 1025) | |
| Reserved IP addresses in AWS VPC (eg: 10.0.0.0/24): Usable: 10.0.0.4 to 10.0.0.254  1) 10.0.0.0: Network addr. 2) 10.0.0.1: Reserved for VPC router. 3) 10.0.0.2: Reserved for Amazon DNS.  4) 10.0.0.3: Reserved by AWS for future use. 5) 10.0.0.255: VPC don’t support broadcast so AWS reserves | |
| Network to VPC Connect-ivity | AWS Managed VPN: simplest way to secure connection from on-prem to AWS  - Use AWS Managed IPsec VPN connection over your existing internet  - Pros: supports static routes, BGP peering and routing. – Cons: Dependent on your internet connection  1) Designate an appliance to act as customer gateway (usually on-prem router)  2) Create Virtual Private Gateway & VPN connection in AWS and download config file for customer gateway  3) Configure customer gateway using config file.  4) Generate traffic from customer side of VPN to bring up VPN tunnel. 5) Configure BGP routing (if needed)  On-prem Customer Gateway VPN connection Virtual Private Gateway AWS | |
| AWS Direct Connect: dedicated network connection over private lines straight into AWS backbone  - Pros: more predictable network performance, BGP peering and routing.  – Cons: require additional telecom and hosting provider r/s and/or new network circuits  - Work w existing Data Networking Provider; Create Virtual Interfaces (VIF) to connect to VPCs (private VIF) or other AWS service like S3 or Glacier (public VIF)  - Requires 802.1Q VLAN support | |
| AWS Direct Connect + VPN: IPsec VPN connection over private lines  - If need added security of encrypted tunnel over Direct Connect. – Pros: more secure than Direct Connect alone. – Cons: more complexity. – Need work w existing Data Networking Provider | |
| AWS VPN CloudHub: connect locations in a Hub and Spoke manner using AWS Virtual Private Gateway  - Link remote offices for backup or primary WAN access to AWS resources and each other  - Pros: reuse existing Internet connection; Supports BGP routes to direct traffic  - Cons: Dependent on Internet connection; No inherent redundancy  - Assign multiple Customer Gateways to a Virtual Private Gateway, ea w own BGP ASN & unique IP ranges | |
| Software VPN: Provide own VPN endpoint and software.  - You must manage both ends of VPN connection for compliance reasons or use VPN option not supported  - Pros: Ultimate flexibility. – Cons: You must design for any needed redundancy across the whole chain  - Install VPN software via Marketplace appliance or EC2 instance | |
| Transit VPC: to connect geographically disperse VPCs and locations in order to create a global network transit center. – Locations and VPC-deployed assets across multiple regions need to communicate w one another. – Pros: ultimate flexibility but also AWS-managed VPN hub-and-spoke btw VPCs  - Cons: you must design for any needed redundancy | |
| VPC-to-VPC Network-ing | VPC Connection Options: VPC Peering, AWS Transit Gateway, Software Site-to-Site VPN, Software VPN-to-AWS Managed VPN, AWS managed VPN | |
| VPC Peering: uses AWS Networking infra to connect VPCs within an acct, across accts, across Regions  - No transitive peering: A peered w B and C ≠ B can peer to C | |
| Transit Gateway (TG): Hub-and-spoke. – Single transit gateway can act as regional hub, connecting up to 5000 attachments, across accts, within 1 Region  - To achieve cross-regional VPC connections, peer multiple transit gateways tgt  - Within ea AZ/subnet of VPC, add a TG attachment in order for subnet to connect to TG | |
| CIDR considerations: when calculating num of available IP addr for subnet, subtract 5 for AWS reserved IP addresses. Netmask can be from /16 to /28  - Cannot connect 2 VPCs with overlapping CIDR blocks | |
| AWS VPCs Transit Gateway Direct Connect (DX) Gateway Direct Connect On-prem | |
| Internet Gateways (IGW) | IGW: Horizontally scaled, redundant and HA component allowing connection btw VPC and Internet  - No availability risk or bandwidth constraints. – If subnet is associated w a route to Internet, then it is a public subnet. – Supports IPv4 and IPv6  - Provide route table target for Internet-bound traffic. – Perform NAT for instances w **public** IP addresses | |
| Egress-Only IGW: provides outbound Internet access for IPv6 instances  - IPv6 addr are globally unique and public by default. – Prevents inbound access to those IPv6 instances  - Stateful-forwards traffic from instance to internet and send back response  - Must create custom route for ::/0 to Egress-Only IGW. – Use Egress-Only IGW instead of NAT for IPv6 | |
| NAT Instance: EC2 instance from AWS-provided AMI. – Translates traffic from many private IP instance to a single public IP and back. – Doesn’t allow public Internet initiated connections into private instance  - Not supported for IPv6. – NAT instance must be on public subnet w route to IGW  - Private instances in private subnet must have route to NAT instance, default route dest of 0.0.0.0/0 | |
| NAT Gateway: Fully managed NAT services that replaces NAT Instance. – Must create in Public subnet  - Use Elastic IP for public IP for life of Gateway. – Created in specified AZ w redundancy in that zone  - For multi-AZ redundancy, create NAT Gateways in ea AZ w routes for private subnet to use local Gateway  - Private instances in private subnet must have route to NAT instance, default route dest of 0.0.0.0/0  - Up to 5Gbps bandwidth that can scale up to 45 Gbps.  – Can’t use NAT Gateway to access VPC peering, VPN or Direct Connect, so need include specific route to those in route table (most specific route is selected first) | |
| Routing | VPC Context: 1) Route Tables: Ea route table contains local route for the CIDR block  - Most specific route for addr wins | |
| VPC Context: 2) Border Gateway Protocol (BGP): routing protocol for Internet  - Propagates info about network to allow for dynamic routing. – Required for Direct Connect, optional for VPN. – Alternative of not using BGP w AWS VPC is static routes  - AWS supports BGP community tagging as way to control traffic scope and route preference  - Required TCP port 179 + ephermeral ports. – Autonomous System Number (ASN) = unique endpoint ID  - Weighting is local to router, higher weight = preferred path for outbound traffic | |
| Route53 uses Routing Policies. ELB uses Request Routing | |
| Enhanced Network-ing | Used for HPC. – Uses single root I/O virtualization (SR-IOV) to deliver high performance than traditional virtualized network interfaces. – Need install driver if not Amazon Linux HVM AMI  - Intel 82599 VF Interface = 10 Gbps. – Elastic Network Adapter = 25 Gbps | |
| Placement Groups:  1) Clustered: instances placed in low-latency group within single AZ. – Lower latency, harder to scale  2) Spread: Instances spread across underlying hardware. – Reduce risk of simultaneous failure if infra fails, can span multi AZ, but max 7 instances running per group per AZ  3) partition: instances grouped into partitions spread across racks. – Reduce risk of correlated hardware failures, better for larger distributed or replicated workloads than Spread, not for Dedicated Hosts | |
| Hybrid & Cross Acct Network-ing | 1) Direct Connect (DX): private route from on-prem to Cloud (physical line). Takes time to provision  On-prem Customer Gateway (enable BGP routing) DX endpoint (at DX location) (private virtual interface) Virtual Private Gateway (VPG) | |
| 2) Site-to-Site VPN: fastest way to connect on-prem to Cloud. – Secure connection using IPSec tunnels using Internet. – Connect directly to VPCs or transit gateways. – 2 tunnels per connection for HA | |
| 3) Establishing HA: 2 data centers can communicate over internet or DX Gateway  - DX Gateway has feature Sitelink: connect 2 data centers  - Each DX Gateway can only connect to ≤ 3 Transit Gateways | |
| PrivateLink | | Securely connect to VPC endpoints outside your VPC. – HA and scalable.  – Traffic don’t traverse public Internet. - Connect to endpoints within same region as your VPC |
| Can connect to 1) Marketplace solutions, 2) Endpoint services owned by your organization in a separate VPC. 3) Peering alternative (for limited, secure connection to resources in another VPC). 4) 3rd -party apps |
| Global Accelera-tor | Move AWS endpoints closer to users. – Allows end users to use AWS edge locations to reach your app faster. – Provides global static public IPs to access app endpoints. – Improves networking performance, security and reliability. – Can be used for failover in multi-region architectures | |
| 1) Public apps: lower latency.  2) Hybrid Networking: lower latency for site-to-site VPN, great middle ground if DX is too pricey  3) Origin Masking: protection from DDoS attacks, static entry point protected by AWS Shield by default  4) Multi-Region Failover: static IP addrs route to endpoints in up to 10 diff regions, picking closest to user | |
| Route 53 | - Register domain names. – Check health of domain resources. – Route internet traffic for your domain  Record types: 1) A/AAAA: return IPv4/IPv6 addr. 2) Alias: URL to an AWS resource  2) CNAME: map DNS queries for current record to another domain or subdomain  3) MX: mail server. 4) NS: name server | |
| Routing policies: 1) Simple: returns dest. 2) Failover: if health check fail, return backup dest  3) Geolocation: route based on location dist. 4) Geoproximity: route based on location resources (bias)  5) Latency: direct to lower latency. 6) Multivalue Answer: return multiple IP addrs  7) Weighted: route according to weight assign | |
| Cross-Acct Route 53 | Hosted Zone: collection of DNS records for a domain. To centralize management | |
| Directing to a Subdomain: 1) Create hosted zone in account associated w your subdomain  2) Copy the NS records from newly created hosted zone  3) Access the parent DNS hosted zone in appropriate acct. 4) Create a new NS record set in parent domain’s hosted zone. 5) Paste the NS records from the child domain’s hosted zone | |
| CloudFront | | Distributed content delivery service for simple static asset caching  - Integrated w Amazon Certificate Manager and supports SNI (Server Name Indication) |
| Certificate Manager: to use SSL/TLS certificates  SNI: allows client to specify which host it’s trying to connect to and server can present multiple certs on same IP otherwise only can serve 1 cert for ea Ip addr  Only Amazon issued SSL/TLS certificates can be auto renewed by AWS Certificate Manager. Custom certs that you import would need to be manually renewed |
| ELB | Distributes inbound connections to 1 or many backend endpoints  1) Application Load Balancer (layer 7). 2) Network LB (layer 4). 3) Classic LB (Layer 4 or 7)  - Can be used for public or private workloads. – Consume IP addrs within a VPC subnet  - Zonal failover, Health checks, Cross-zone LB, CloudWatch Metrics, SSL Offloading, Resource-based IAM permissions. – ALB & NLB work in VPC; CLB work in EC2-Classic or VPC | |
| |  |  |  |  | | --- | --- | --- | --- | |  | ALB | NLB | CLB | | Protocols | HTTPS, HTTP | TCP, UDP, TLS | TCP, SSL, HTTP, HTTPS | | Path or Host-based Routing | Yes | No | No | | WebSockets | Yes | Yes | No | | SNI | Yes | No | No | | Sticky Sessions | Yes | No | Yes | | Static IP, Elastic IP | No | Yes | No | | User Authentication | Yes | No | No | | |
| Routing: 1) ALB: host-based/path-based/HTTP header-based/HTTP method-based/Query string parameter-based/Source IP addr CIDR-based routing,  2) NLB route based on Port number. TCP connections to backend are persisted for duration of connection | |
| 1) To allow VPC instances to resolve using on-premises DNS:  - Configure DHCP Option Set to issue your on-prem DNS IP to VPC clients  2) Domain name registered w Route53 in acct A. To register a subdomain in acct B. You have already created the Hosted Zone for the subdomain in Account B. How to query forward from parent DNS acct to new subdomain?  - Retrieve NS record for hosted zone associated w subdomain in acct B. Then, within hosted zone of parent domain in acct A, create a new NS record specifically for the subdomain and input the NS record values retrieved from acct B.  3) Implement HA hybrid connectivity btw on-prem data center and AWS Region. Company use transit gateway and AWS DX connection. Wants to set up a secondary connection for its most critical latency-sensitive app. These apps don’t have high data-transfer activity:  - Set up accelerated AWS Site-to-Site VPN connection (which use AWS Global Accelerator to route traffic from on-prem to AWS edge location). Enable NAT-traversal (NAT-T) for accelerated VPN connection  4) App in 2 subnets in VPC. Employees connect through on-prem data center to connect to app. 2 subnets currently accessible only from on-prem through a DX connection. Now need allow access from internet. App access from company on-prem DC must continue to occur exclusively through DX connection:  - Create private VIF in DX. Configure ea subnet w a route to a newly created internet gateway. Propagate specific on-prem routes through BGP.  5) HPC cluster on EC2. Want incr throughput and reduce network latency  - Run HPC cluster in single AZ. Select EC2 instance type w Elastic Fabric Adapter (EFA) to accelerate HPC and ML apps. Disable CPU hyperthreading on EC2  6) Web app on EC2 behind ALB. Company deploy same app to new region and want customer to use apex domain name to access app:  - Create Amazon Route 53 public hosted zone w record for apex domain name. Set up a DNS active-active failover config that uses latency-based routing. Set “Evaluate Target Health” to “Yes” for set of alias resource records (CNAM cannot map to apex domain name)  7) Web servers in VPC must be available on Internet. App must have HA architecture:  - Place all web servers behind ELB. Configure Route53 CNAME to point to ELB DNS name. Assign EIP to all web servers. Configure Route53 record set w all EIPs, w health check and DNS failover  8) DX connection. Link up and routes advertised from customer’s end, but customer unable to connect from EC2 instances inside its VPC to on-prem servers. How to fix?  - Enable route propagation to virtual private gateway (VGW).  – Modify Instances VPC subnet route table by adding route back to customer’s on-prem env  9) ELB support proxy protocol: use if apps need to know IP address of clients  10) Company has multiple AWS accts as part of an organization created w AWS Organizations. Ea acct has a VPC in us-east-2 Region and used for either production or development workloads. EC2 across production accts need to communicate w e.o and EC2 across development accts need to communicate w e.o, but production and development should not be able to communicate w e.o.  To facilitate connectivity, company created a common network acct. Company used AWS Transit Gateway in us-east-2 Region In network acct and shared transit gateway w entire organization by using AWS Resources Access Manager. Network admins then attached VPC in ea acct to transit gateway. How to stop production from communicating w development?  - Create separate route tables for production and development traffic. Delete ea acct association and route propagation to the default AWS Transit Gateway route table. Attach development VPCs to the development AWS Transit Gateway route table and production VPCs to the production route table, and enable automatic route propagation on each attachment  11) Benefits of using IPSec tunnel.  - Data encryption across Internet. – Protection of data in transit over Internet.  - Peer identity authentication btw VPN gateway and customer gateway. – Data integrity protection across the Internet  12) Steps to use Direct Connect:  - a) Submit AWS Direct Connect Connection Request. b) Complete the Cross Connect (optional; for cross region connectivity). c) Configure redundant connections w AWS Direct Connect. d) Create a Virtual Interface. e) Download Router configuration. f) Verify your Virtual Interface  13) Steps to connect to AWS VPC using Direct Connect:  a) Provide a private Autonomous System Number (ASN) to identify your network on the Internet.  b) Amazon then allocates a private IP addr in the 169.x.x.x range to you  c) Create a virtual private gateway and attach it to your VPC | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Security | | | |
| Concepts | Shared Responsibility Model. IAM: Principle of Least Privilege  Identity: IAM user or role, root acct user, temp security credentials  Authentication: prove you are who you say. MFA, client-side SSL certificate  Authorization: are you allowed to do this? IAM policies  Trust: do other entities I trust say they trust you? Cross-acct access, SAML-based/ web identity federation  Identity -> Identity Broker -> Identity Store (to authenticate identity) OR Federation (3rd party identity provider to validate identity of user. Google, Cognito) -> Provide key or token to authorize identity to take specific actions -> Service provider (AWS)  Malicious Packets to VPC or DDoS on application  Trust policy: defines which principals can assume the role, and under what conditions  Identity-based policies: define permissions user of role is able to perform and on which resources  Permission boundary: feature for managed policy to set max permissions for a role/user (not group) | | |
| AWS Organiza-tions | Centralizes management of multiple accts. Billing is also consolidated, hence larger discounts  In an Organization: 1) Management acct. 2) Member accts: contains AWS resources and workloads  3) Organizational Units (OU): group of accts that serve an app or service | | |
| Control Tower: AWS service to automate multi-acct management  - Centralize & automate guardrails across organization: apply service control policies or AWS Config rules  - Provides organization dashboard to improve visibility  - Landing zone: What you provision when you start using Control Tower  - Baseline: combination of blueprints (CloudFormation stacks) and guardrails applied to member acct  - Default accts created: 1) Core OU: a) Log Archive acct. b) Audit acct: contain cross-acct audit roles  2) Custom OU: can create ≥ 1 custom OU to provision member accts in | | |
| Control Access in Org | Service Control Policies: applied to org, OU, accts. – Use IAM policy syntax, but never grant permissions (only DENY). – Effects inherited by all accts below the SCP’s target (cascade down)  - Best practice not to apply SCP to individual acct  - Deny list: explicit deny specific actions. Allow list: implicit denies all actions not listed. Orgs are assigned SCP FullAWSAccess by default  - An action is enabled only if allowed by IAM policy and not denied by an SCP | | |
| AWS Config: monitor resources config across organization  - Define detective controls and check if accts are compliant. – Discover noncompliant resources  - Keep history of actions that caused noncompliance | | |
| IAM Identity Center (formerly SSO): recommended way to give users access to AWS accts  - Maps users and groups from an identity provider to IAM users and groups  - Integrates w identity providers that leverage SAML 2.0, like Azure AD  - Can act as an independent user directory.  User (in Identity Provider) -> IAM Identity Center -> Grant role defined in IAM -> Access resources in acct | | |
| IAM Users: - static access keys  - Grants 1 user access to 1 acct  - 1 permissions set per user  - Federated Identity not supported | | Identity Center: - roles w rotating access keys  - Grants 1 user access to many accts  - Many roles assumed by user  - Designed for Federated Identity |
| AD Connector: uses Microsoft Active Directory  - Use existing users and groups to grant permissions in AWS accts. - Integrates w Identity Center  - Allows for single source for credential management | | |
| MFA: can enable in Identity Center to require for all users in Organization | | |
| Review User Activity: Access Log Archive acct in organization. – Uses CloudTrail logs  - Logs cannot be deleted. – Can monitor CloudTrail logs w CloudWatch  - Use GuardDuty for intelligent threat detection | | |
| NACLs & SGs | SG: Virtual firewalls for individual resources (EC2, RDS, AWS Workspaces)  - Controls inbound and outbound traffic for TCP, UDP, ICMP or custom protocols. – Port or port ranges  - Inbound rules are by Source IP, Subnet, other SG. – Outbound rules are by Dest IP, Subnet, other SG  NACL: Security for VPC that acts as firewall. – Apply to entire subnets rather than individual resources  - Default NACL allows all inbound and outbound traffic. – NACLs are stateless  - Remember ephemeral ports for Outbound if you need them  NACLs provide backup mtd of security if you accidentally change your SG to be too permissive | | |
| Trouble-shooting | EC2 -> SG -> Subnet (NACL) -> Subnet (Route Table) -> IGW  Internet facing EC2 need public IP addr (Elastic IP) | | |
| AWS Directory Services | |  |  |  | | --- | --- | --- | | Directory Service Option | Description | Best for | | AWS Cloud Directory | Cloud-native dir to share & control access to hierarchical data btw apps | Cloud apps that need hierarchical data w complex r/s | | Amazon Cognito | Sign-up & sign-in features & federated to public social media services | Develop consumer apps or SaaS | | AWS Directory Service for Microsoft Active Directory | AWS-managed full Microsoft AD running on Windows Server | Enterprises that want hosted Microsoft AD or need LDAP for Linux apps | | AD Connector | Allows on-prem users to log into AWS w their existing AD credentials. Allows EC2 to join AD domain | SSO for on-prem employees and adding EC2 instances to domain | | Simple AD | Low scale, low cost AD implementation based on Samba | Simple user directory, or need LDAP compatibility | | | |
| AD: supports MFA vs Simple AD: MFA not supported | | |
| Credential & Access Manage-ment | AWS Security Token Service (STS): temporarily grant credential access to apps or users  - Can get credentials from IAM or federated options  Cognito: more for mobile apps  Apps -> Identity Broker -> AD/Cognito -> Identity Broker -> STS -> Apps -> AWS Services | | |
| Token Vending Machine concept: common way to issue temp credentials for mobile app development  - Anonymous TVM: provide access to AWS services only, don’t store user identity  - Identity TVM: used for registration and login, and authorizations  - AWS recommends mobile devs to use Cognito and SDK | | |
| Secrets Manager: store pw, encryption keys, API keys, SSH keys, PGP keys, …  - Alternative to storing pw or keys in a vault. – Can access secrets via API w fine-grained access control by IAM. – Auto rotate RDS credentials for MySQL, PostgreSQL, Aurora. | | |
| Sharing Roles across Accts | Resource Access Manager: centralized resource sharing w principals across many accts  - Share w individual accts or OUs within org. – Resources still owned by sharing acct  - Resource sharing must be enabled by organization  1) Choose resource to share. 2) Apply managed policy to resource share  3) Define principals for resource share. 4) Send share invite to principal acct(s) | | |
| Use Cases: 1) Foundational Infra: share VPC infra across accts in organization  - AWS Organization has infra acct w VPC. Individual accts cannot manage own network but must be able to create AWS resources within subnets  - Enable resource share in AWS Organization management acct. Create resource share in AWS RAM in infra acct. Select specific AWS Org OU that will use the shared network. Select ea subnet to associate w resource share  2) Certificate Authorities: centrally manage private cert auths and share to reduce cost and complexity  3) App Mesh Networking: allows networking btw disparate compute infra and share a mesh  4) Aurora clusters: share Aurora or RDS clusters across accts and allow cross-acct cloning | | |
| Most shared resources are regional (can only be shared within same region)  To share across organizations: use *enable-sharing-with-aws-organization* CLI command for trusted access | | |
| Encryption | | 1) Encryption at rest: data encrypted in storage  2) Encryption in Transit: data encrypted as it flows through network/process, such as SSL/TLS for HTTPS, or IPSec for VPN connections | |
| KMS: key storage, management, auditing. – Integrated into many AWS services like Lambda, S3, EBS, EFS, DynamoDB, SQS, …. – Can import own keys or use KMS generated ones  - Control who manages and accesses keys via IAM users and roles. – Audit use of keys using CloudTrail  - Differs from Secret Management as purpose-build for encryption key management  - Validated by many compliance schemes (PCI DSS Level 1, FIPS 140-2 Level 2) | |
| CloudHSM: dedicated hardware device for a single tenant.  – Must be within VPC and can access via VPC Peering. – Don’t integrate w AWS services like KMS  - Offload SSL from web servers, act as issuing CA, enable TDS for Oracle DB | |
| |  |  |  | | --- | --- | --- | |  | CloudHSM | AWS KMS | | Tenancy | Single-Tenant HSM | Multi-Tenant AWS Service | | Availability | Customer-managed durability & availability | HA & durable key storage & management | | Root of Trust | Customer managed root of trust | AWS managed root of trust | | FIPS 140-2 | Level 3 | Level 2 / 3 in some areas | | 3rd party support | Broad 3rd party support | Broad AWS Service support | | |
| AWS Certificate Manager: managed service to provision, manage, deploy public/private SSL/TLS certs  - Integrated into many AWS services like CloudFront, ELB, API Gateway  - Free public certs to use w AWS services; no need register via 3rd party cert auth (but can import 3rd party cert for use on AWS)  - Supports wildcard domains (\*.domain.com) to cover all subdomains. - Managed certificate renewal.  – Can create managed private cert auth for internal or proprietary apps, services or devices | |
| Secrets Manager | Auto rotate encryption keys to keep secrets safe. – Use KMS keys to encrypt  - Can use Lambda to auto rotate keys. – IAM roles grant services and apps access to specific secrets  - Can monitor and audit logs using CloudWatch and CloudTrail | | |
| Distributed Denial of Service (DDoS) attacks | | Use multiple compromised machines to overwhelm app/network w traffic/requests  1) Amplification/Reflection attacks: NTP (Network Time Protocol) service has command MONLIST (for monitoring; returns last 600 IP addrs it’s talked to). Bad actor send spoofed pkt to NTP server, who then send MONLIST output to target device  2) Application Attacks (Layer 7): Bad actor flood target w HTTP GET requests  To mitigate DDoS: 1) Minimize attack surface = NACLs, SGs, VPC design  2) Scale to absorb attack = Auto-Scaling, CloudFront, Static web content via S3  3) Safeguard exposed resources = Route 53, AWS WAF, AWS Shield  4) Learn normal behaviour: AWS GuardDuty, CloudWatch | |
| Managed Security Services | Security Hub: single-panel view of prioritized security insights  - Integrates w Inspector, GuardDuty, Firewall Manager, Macie to deliver security insights  - Inspector discover workloads and scans for software vulnerabilities and unintended network exposure  - Generates and prioritizes recommendations based on AWS best practices  - Helps integrate security findings in multi-acct envs. – Can integrate w 3rd party security services | | |
| Network Firewall vs WAF: Network firewall sits outside VPC and filter traffic from TGW, DX, VPN, IGW  WAF can filter traffic to and from ALB, CloudFront, API GW, AppSync (distributed or serverless apps) | | |
| |  |  | | --- | --- | | Shield for DDoS | Shield Advanced | | Protection against known infra attacks (layer 3 & 4) | more advanced DDoS attacks (include layer 7) | | Applied to all AWS services when activated | Specify resources to protect (ELB, CloudFront) | | Standard AWS support for DDoS assistance | Dedicated 24x7 access to DDoS experts | | Limited visibility on attacks | Detailed logs to analyze attacks | | | |
| Firewall Manager: manages many firewalls across an organization  - Standardise firewall rules across acct & org. – Deploy security tools at scale, spanning many VPC & accts. – Integrates w SG, WAF, Shield, Network Firewall, 3rd party tools. – Provide audit details to Security Hub | | |
| GuardDuty: use ML to intelligently detect threats and nefarious actions all throughout AWS acct  - Continuously inspect API logs that are generated and sent to CloudFront by your various apps/users  - Generate security alerts and recommendations (send to user, Security Hub, EventBridge to remediate) | | |
| Intrusion Detection System (IDS) and IPS (Prevention) | | IDS: watches network and systems for suspicious activity that might indicate someone trying to compromise a system  IPS: prevent exploits by sitting behind firewalls and scanning & analyzing suspicious content for threats  Normally comprised of a Collection/Monitoring system and monitoring agents on each system  Logs collected or analyzed in CloudWatch, S3 or 3rd party tools (Splunk, SumoLogic,… ) sometimes called a Security Information and Event Management (SIEM) system  - CloudWatch logs stored indefinitely. CloudTrail logs stored to S3 or CloudWatch indefinitely  - CloudWatch alarms history stored for 14 days, CloudTrail no alarms (use CloudWatch) | |
| Service Catalog | Framework allowing admins to create pre-defined products and landscapes for their users  - Granular control over which users have access to which offerings  - Makes use of adopted IAM roles so users don’t need underlying service access  - Allows end users to be self-sufficient while upholding enterprise standards for deployments  - Based on CloudFormation templates.  – Admins can version & remove products. Existing running product versions will not be shutdown | | |
| |  |  |  | | --- | --- | --- | | Type | What | Why | | Launch Constraint | IAM role that Service Catalog assumes when an end-user launches a product | W/o launch constraint, end-user must have all permissions within their own IAM credentials | | Notification Constraint | Specifies the SNS topic to receive notifications about stack events | Get notifications when products are launched or have problems | | Template Constraint | ≥ 1 rules that narrow allowable values an end user can select | Adjust product attributes based on choices a user makes (certain instance for DEV env) | | | |
| Can share Service Catalog Portfolio to multi accounts as baseline. Then individual acct add own Portfolio | | |
| Questions | Prevent root users from taking action in accts.  - Root user cannot assume IAM roles. Use SCP to deny all actions for root users and attach to root OU  OAuth 2.0 handles authorization  Mobile app needs secure access to AWS resources: use Cognito SDK to provide temp credentials | | |
|  | 1) In AWS Organization, 2 VPCs connected to transit gateway in dev acct. New prod acct w another VPC. How to attach prod VPC to transit gateway? - Use Organizations management acct to activate resource sharing. – Use AWS RAM to share transit gateway w production acct. – Attach prod VPC to transit gateway  2) EC2 instance under network attack. Need inspect network traffic message content  - VPC Flow Logs can capture IP traffic gg to and from network interfaces in VPC, BUT don’t capture message content. – Instead, use Traffic Mirroring to duplicate the data and analyze it  3) Cognito user pools: authentication user (sign in through user pool or federate through 3rd party)  Cognito Identity pools: authorization (create identities for users & give them access to AWS services)  4) IAM Access Analyzer: analyze CloudTrail events to identify actions and services that have been used by an IAM entity such as an IAM role. Policy generation then generates an IAM policy based on action-level and service-level info.  5) ARN for IAM: arn:aws:service:region:account:resource  - If e.g. region left blank = global  CloudFront Origin Shield: reduce latency by adding caching layer closer to origin  6) Website is single-page app w files stored in S3, delivered using CloudFront. Company get several fake postings every day that are manually removed. Security team identified most of the fake posts are from bots w IP add that have a bad reputation within the same global region.  - Associate an AWS WAF web ACL w the CloudFront distribution. Select the managed Amazon IP reputation rule group for the web ACL w a deny action  7) SG default settings:  - Allow no inbound traffic. Allow all outbound traffic and allow instances associated w this SG to talk to e.o  8) Max number of VPC allowed in a Region:  - 5 VPCs | | |

|  |  |
| --- | --- |
| Migrations | |
| Migration Strategies | |  |  |  | | --- | --- | --- | | Migration Strategy | Description | Eg | | Re-Host | Lift and Shift | Move on-prem MySQL DB to EC2 instance | | Re-Platform | Lift and Reshape | Migrate on-prem MySQL DB to RDS MySQL | | Re-Purchase | Drop and Shop | Migrate legacy on-prem CRM system to salesforce | | Rearchitect | Redesign app in cloud-native manner | Create serverless version of legacy app | | Retire | Get rid of apps not needed | End of life app | | Retain | Do nothing |  | |
| Cloud Adoption Framework | Business (strong business case, ROI), Platform (resource provision, architecture),  People (new skills and roles), Security (IAM, logging, audit),  Governance (mange portfolio to determine cloud eligibility), Operations (service monitoring, DR, HA) |
| Hybrid Architec-ture | Make use of cloud resources along w on-prem resources  Ideally, integrations are loosely coupled: each end can exist w/o extensive knowledge of other side  E.g. Storage Gateway w local cache on-prem and S3 in cloud |
| Migrating Apps | Workload Migrations: 1) AWS Batch or Fargate can be used to host migrated containerized workloads  2) Elastic Beanstalk or Lightsail could be used to host migrated web apps  3) Event Driven or Serverless Architecture could be used to migrate message handling microservices, decoupling queues, event handling workloads |
| Migration Hub: central hub for migration planning and tracking  - Leverages ADS to analyze and group servers. – Gives instances recommendations to help right-size workloads. – Provides migration strategy best practice advice.  - Keeps track of lift-and-shift and refactor-first migration progress w dashboard |
| App Discovery Service (ADS): audit on-prem env and find out which servers are involved in a given app  1) Install ADS agent on all on-prem servers . 2) Agents sends encrypted data over internet on which server being used to ADS dashboard in Migration Hub |
| App Migration Service (MGN): lift servers to AWS Cloud.- Migrate Windows & Linux servers to AWS Cloud. – Sources servers can be on-prem, AWS, other cloud providers  - Allows for test env and cutover w minimal interruption  - Install MGN agent on servers you intend to migrate to right-size EC2 instances  - MGN is free, only pay for resources provisioned. – Can also use to replicate servers across regions  1) Install MGN agent of servers to migrate. 2) Once connection established over VPN or DX, MGN agent will continuously replicate block storage of servers |
| Bringing AWS to your Data Center: 1) Runs EKS Anywhere or ECS Anywhere in Data Centers  - Takes advantage of AWS-managed container orchestration while doing process on-prem  2) AWS Outposts: physical servers or racks to order from AWS that are preconfigured to run AWS services  3) Snowball Edge: need order device, can use EKS Anywhere or EC2 to process data w import/export jobs |
| Migrating Data | DB Migration Service (DMS): migrate source relational DB to RDS, Aurora, EC2-based DB  - Built-in Schema Conversion Tool (SCT) allows for heterogeneous migrations  - Can be used for smaller homogeneous migrations and MongoDB, DynamoDB migrations  - SCT will be needed for complex data sets like data warehouses.  – Can replicate directly to AWS, S3, Snowball  1) Create Replication instance. 2) Create source and target endpoints pointing to DB  3) Create DB migration task |
| DataSync: migrate File systems or obj storage. – Compatible w SMB, HDFS, NFS fileshares  – Secure online service that automates moving data from on-prem to AWS  - Move self-managed obj store to S3. – Sync data to EFS or Amazon FSx  1) Install DataSync agent on-prem to sync to Snowball or S3 on Outposts  2) DataSync then sync from agent to AWS |
| Transfer Family: transfer large files to and from S3 and EFS. - Multi-AZ architecture for HA.  – Use SFTP, FTP, FTPS, AS2 protocols for file transfers. - Use Transfer Family Managed File Transfer Workflows (MFTW) to automate encryption, filtering, tagging, compression  - Authenticate external users to securely transfer files and obj to and from AWS |
| Network Migration and Cutovers | Ensure IP addr don’t overlap btw VPC and on-prem. – VPC supports IPv4 netmasks range from /16 to /28  - 5 IPs are reserved in every VPC subnet. |
| Most orgs start w VPN connection to AWS. – As usage grows, might use DX but keep VPN as backup  - Transition from VPN to DX should be seamless using BGP.  - Once DX set up, configure VPN and DX withing the same BGP prefix  - From AWS, DX always preferred, but need ensure DX path is preferred route from network to AWS and not VPN (through BGP weighting or static routes) |
| Snow Family | Move massive amt of data to and from AWS. – Data transfer speed depends on cost (UPS, DHL, FedEx)  - Encrypted at rest and in transit.  1) AWS Import/Export: Ship external hard drive to AWS. At AWS, plug it in and copy data to S3  2) AWS Snowball: NAS in a box which AWS ships to you. Copy over up to 80TB of data and ship back to AWS. They copy data over to S3  3) AWS Snowball Edge: same as Snowball, but w built-in Lambda, clustering, compute power  4) AWS Snowmobile: Shipping container full of storage (up to 100PB) and truck to transport it |
| Questions | 1) Replicate large amt of existing and new data in S3 bucket to another Region:  - S3 Batch Replication for existing objs. Configure a replication for new objects  2) Company migrate legacy Objective-C app and IBM Informix DB that company runs on fleet of physical Windows servers:  - Use AWS App Migration Service to lift-and-shift and rehost existing app and DB to EC2. Install the AWS Replication Agent on source servers. Configure App Migration Service by using AWS Replication Agent  3) Migrate 150 VMs to AWS w minimal downtime. Also need migrate 300 TB of data to AWS region. W current connection, can migrate in 5 days  - Transfer data by using S3 Transfer Acceleration. Use AWS App Migration Service to migrate VM  4) Migrate web app to AWS |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Architecting to Scale | | | | |
| Scaling in the Cloud | Loosely Coupled Architecture: components can work independently and require little to no knowledge of the inner workings of other components | | | |
| Scaling Up: add more CPU/RAM as demand incr  Requires restart to scale up or down  Require scripting to automate  Limited by instance sizes | | Scaling Out: add more instances as demand incr  No downtime required  Auto scaling available for compute services  Theoretically unlimited scaling potential | |
| Event-Driven Architecture: form of loosely coupled architecture that involves event producers, sending event routers and those routers sending events to event consumers | | | |
| Auto Scaling | 1) EC2 Auto Scaling: setup scaling groups for EC2; health checks to remove unhealthy instances  2) App Auto Scaling: API to control scaling for resources other than EC2 like Dynamo, ECS, EMR  - Provides a common way to interact w the scalability of other services  3) AWS Auto Scaling: centralized way to manage scalability for whole stacks; predictive scaling feature  - Console that can manage 1 and 2 from a unified standpoint | | | |
| EC2 Auto-Scaling groups: auto provides horizontal scaling.  – Availability, Cost, System Metrics factor in scaling  – Triggered by event or scaling action to launch or terminate instances  1) Maintain: keep a specific or min num of instances running  2) Manual: use max, min, or specific num of instances  3) Schedule. 4) Dynamic: scale based on real-time metrics of system  For Auto Scaling launch configs: - specify VPC and subnets for scaled instances  - attach to a ELB. – Define a Check Grace Period. – Define size of group or scaling policy | | | |
| For dynamic scaling: 1) Target Tracking Policy: scale based on predefined metric in relation to a target value (CPU utilization at 70%, scale up)  2) Simple Scaling Policy: Waits until health check & cool down period expires before evaluating new need  3) Step Scaling Policy: responds to scaling needs w more sophistication and logic | | | |
| Scaling Cooldown: configurable durations that gives scaling chance to boot up and absorb load  - Default cooldown period = 300s. – Auto applies to dynamic scaling and optional to manual scaling but not supported for schedule scaling. – Can override default cooldown via scaling-specific cool down | | | |
| App Auto-Scaling: 1) Target Tracking Policy. 2) Step Scaling. 3) Scheduled Scaling | | | |
| AWS Predictive Scaling: dynamically scale based on ML learning your load and calculating expected capacity. – Or can just use the data calculated to scale manually | | | |
| Compute Optimizer | Use ML to give recommendations on compute resources  - Incr cost efficiency by reconfiguring over-provisioned resources  - Improve performance by rightsizing under-provisioned resources. - Activate in 1 acct or across org | | | |
| In compute resources, send metrics to CloudWatch -> Compute Optimizer -> Calculate recommendations | | | |
| Compatible Resources: 1) EC2 instances (instance types, Auto Scaling groups), 2) EBS volume types and sizes, 3) ECS on Fargate task size and container size, 4) Lambda functions CPU or memory allocation | | | |
| Kinesis | Collection of services for processing streams of various data  - Data processed in shards, each shard able to ingest 1000 records per sec, 5000 reads per sec  - Default limit of 500 shards, but can request incr to unlimited  - Record consists of Partition Key, Sequence Num, Data Blob (up to 1 MB)  - Same partition key gets send to same shard. Seq Num monotonic increasing  - Can act as Transient Data Store: default retention of 24 hours, can configured up to 7 days | | | |
| Kinesis Video Streams: camera devices stream video to AWS using Kinesis Video Streams SDK | | | |
| KDS: ingest and stores data streams for processing  KDF: prepares and loads data continuously to destinations you choose  Kinesis Data Analytics: run standard SQL queries against data streams | | | |
| DynamoDB Scaling | | 1) Throughput: Read/Write Capacity Unites. 2) Size: max item size = 400 KB  Partition: physical space where data is stored. Partition/Hash key = unique ID.  Sort/Range key = define storage order. Used in combination w partition key  DynamoDB scales out by adding partitions. Formula:  - By Capacity = (Total RCU/3000) + (Total WCU/1000). – By Size = Total size / 10 GB  - Total partitions = Round up for MAX(By Capacity, By Size) | | |
| Auto Scaling for DynamoDB: when reach certain target utilization or read/write capacity, scale up  - Use Target Tracking mtd to stay close to target utilization.  – Does not scale down if table’s consumption drops to 0  - Workaround 1: Send requests to table until it auto scales down  - Workaround 2: Manually reduce max capacity to be same as min capacity  - Also supports Global Secondary Indexes (copy of the table) | | |
| On-Demand Scaling: alternative to Auto-Scaling  - Use if can’t deal w scaling lag or have no idea of the anticipated capacity requirements  - Instantly allocates capacity as needed w no concept of provisioned capacity  - Costs more than traditional provisioning and auto-scaling | | |
| DynamoDB Accelerator (DAX): in-memory cache sitting in front of DynamoDB table  - For microsecond level response time using DAX is DynamoDB millisecond not enough | | |
| CloudFront | | Deliver content to users faster by caching static and dynamic content at edge locations  - Dynamic content delivery is achieved using HTTP cookies forwarded from your origin  - Supports Adobe Flash Media Server’s RTMP protocol but have to choose RTMP delivery mtd  - Web distributions also support media streaming and live streaming but use HTTP or HTTPS  - Origins can be S3, EC2, ELB, or another web server. – Multiple origins can be configured  - Use Behaviors to configure serving up origin content based on URL paths | | |
| To invalidate/clear CloudFront Cache: 1) Delete file from origin and wait for TTL to expire  2) Use AWS Console to request invalidation for all content or a specific path  3) Use CloudFront API to submit an invalidation request  4) Use 3rd party tools to perform CloudFront invalidation (CloudBerry, Ylastic, CDN Planet) | | |
| - Supports zone apex DNS entries. – Supports Geo-Restrictions | | |
| Event-Driven Architec-ture | Event-Driven Architecture: integrate serverless services  - Trigger asynchronous events across AWS env w events.  – Event = change in state or update signalled by an event producer  - Event routers filter and push events and their payloads to event consumers  - Event consumers use payloads to update state and make backend changes | | | |
| Serverless: 1) No server management: patches and OS updates handled by provider  2) Flexible scaling: app scales automatically within bounds defined  3) HA: availability and fault tolerance built in by default. 4) Scales to 0: when services not in use | | | |
| 1) Lambda: serverless compute for custom logic  2) EventBridge: event bus for choreographing asynchronous events  3) Step functions: orchestrates stateful, synchronous workflows involving many services  4) SQS: event buffer for decoupling serverless workflows  5) SNS. 6) API Gateway: receive events triggered by external producers  7) DynamoDB: scalable, serverless, NoSQL DB service. 8) S3 (simple storage service) | | | |
| Orchestration: orchestrating highly coupled workflow, can involve things like decision trees, retries  Choreography: asynchronous coordination of many event producers and consumers | | | |
| SNS | Enables a Publish/Subscribe design pattern. – Topics = channel for publishing a notification  - Subscription = configuring an endpoint to receive messages published on the topic  - Endpoint protocols include HTTP(S), Email, SMS, SQS, Amazon Device Messaging (push notif), Lambda  - Fan-out architecture | | | |
| SQS | Reliable, highly scalable, hosted message queuing service. – Integrates w KMS for encrypted messaging  - Transient storage – default 4 days, max 14 days. – Optionally supports FIFO queue ordering  - Max message size of 256 KB but using special JAVA SQS SDK, can have messages as large as 2GB | | | |
| Amazon MQ: managed implementation of Apache ActiveMQ (message broker)  - Fully managed and HA within a region. – ActiveMQ API and support for GMS, NMS, MQTT, WebSocket  - Designed as a replacement for on-prem message brokers. – Use SQS if creating from scratch | | | |
| Lambda, Serverless Application Model, EventBridge | | Lambda: run code on-demand w/o need for infra. – Supports Node.js, Python, Java, Go, C# - Code is stateless and executed on an event basis (SNS, SQS, S3, DynamoDB Streams, …0  - No limits to scaling a function since AWS dynamically allocates capacity in relation to events | | |
| AWS Serverless Application Model (SAM): open source framework for building serverless apps on AWS  - Use YAML as config language. – Includes AWS CLI-like functionality to create, deploy, update serverless apps using AWS services like Lambda, DynamoDB, API Gateway  - Enables local testing and debugging of apps using Lambda-like emulator via Docker  - Extension of CloudFormation so can use everything CloudFormation can provide by resources & fns  - sam local <filename?>. - sam deploy | | |
| EventBridge: Designed to link variety of AWS and 3rd party apps to rules logic for launching other event-based actions | | |
| Scaling Containers | | Control and Complexity: EKS -> ECS -> Fargate -> App Runner  - Both EKS and ECS can run on Fargate, EC2, Outposts, Local Zones, AWS Wavelength (5G optimized edge compute solution) | | |
| EKS: supports Kubernetes ecosystem  - Requires customization to power SageMaker, Polly, Batch, …  - Adding Load Balancers to container instances requires generalized abstractions | | ECS: AWS solution for running containers at scale  - Natively integrates w many AWS services  - Seamlessly integrates w ALB or NLB |
| Fargate: purpose-built compute layer for containerized workloads  - Handles scaling, security, server management for ECS or EKS containerized workload compute layer  - Integrates w CloudWatch and Container Insights for monitoring apps  - Pay for what you use, and scale quickly to meet demand.  - Great for request/response, batch, event handling, ML, … workloads  - Scale compute layer per task, as opposed to hosting multiple tasks per instance (on EC2) | | |
| App Runner: minimize operational overhead for HTTP apps  - Designed exclusively for Synchronous HTTP apps  - Supplies compute and networking for container images, Python, Java, Node.js projects  - Supports public and private endpoints. – Scales to 0 | | |
| AWS Batch: run thousands of containerized Batch Jobs  - Plans, schedules, executes compute workloads. - Runs jobs using ECS, EKS or Fargate  - Dynamically provisions CPU or memory-optimized compute resources based on needs  - Reduce costs by optionally running jobs on spot instances | | |
| Step Functions & Batch | Step Functions: managed workflow and orchestration platform. - Scalable and HA  - Define app as state machine. – Create tasks, sequential steps, parallel steps, branching paths or timers  - Amazon State Language declarative JSON. – Apps can interact and update stream via Step Functions API  - Visual interface describes flow and realtime status. – Detailed logs of ea step execution | | | |
| AWS Batch: management tool for creating, managing, executing batch-oriented tasks using EC2  1) Create Compute Env: managed or unmanaged, Spot or On-Demand, vCPUs  2) Create Job Queue w priority and assigned to Compute Env  3) Create Job Definition: Script or JSON, env variables, mount points, IAM role, container image, …  4) Schedule Job | | | |
| |  |  | | --- | --- | | Step Functions | Out-of-the-box coordination of AWS service components | | Simple Workflow Service | Need to support external processes or specialized execution logic | | SQS | Messaging Queue; Store and forward patterns | | AWS Batch | Scheduled or reoccurring tasks that do not require heavy logic | | | | |
| EMR | - Hadoop HDFS: Persistent Datastore. – Hadoop MapReduce: Distributed Processing  - ZooKeeper: Resource Coordination. – Oozie: Workflow framework. – Pig: Scripting language  - Hive: SQL interface for Hadoop. – Mahout: ML component. – HBase: columnar DB for Hadoop data  - Flume: log collection. – Sqoop: Data transfer. – Ambari: management and monitoring | | | |
| EMR: managed Hadoop framework for processing huge amounts of data  - Supports Spark, HBase, Presto, Flink. – Commonly used for log/financial analysis, ETL  - Step = programmatic task for performing some process on data  - Cluster = collection of EC2 provisioned by EMR to run your Steps  - Master Node, Core Nodes (HDFS storage), Task Nodes (run steps) | | | |
| Monitoring & Visualizing Data at Scale | | QuickSight: BI solution. – serverless pay-per-use service, low cost  - Use SPICE (in-memory cache for super-fast data analytics) and AutoGraph (best-fit graph) technologies  - Accessible from web browsers or mobile apps. – Creates hybrid datasets from multiple sources  - Can leverage AWS AI/ML for natural language queries and automated insights | | |
| OpenSearch: open-source search and analytics suite forked from ElasticSearch  - Provides AWS-managed OpenSearch clusters to analyze logs/metrics in real time  - Create OpenSearch dashboards or use built-in Kibana  - More advanced querying and lower cost at scale compared to CloudWatch  - Can be used as data source for QuickSight | | |
| Questions | 1) AWS Batch need run on EC2 w custom AMI and can handle interruptions.  - Modify AWS Batch job to run in managed EC2 compute env that uses the AMI. Select Spot as provisioning model  2) CloudWatch basic vs detailed monitoring  - Basic free and enabled by default, metric every 5 min  - Detailed paid and not enabled by default, metric every 1 min: RDS, EC2, Auto Scaling, ELB, Route 53 | | | |

|  |  |
| --- | --- |
| Business Continuity | |
| Concepts | Business Continuity (BC): minimize business activity disruption when something unexpected happens  Disaster Recovery (DR): act of responding to an event that threatens business continuity  High Availability (HA): designing in redundancies to reduce chance of impacting service levels  Fault Tolerance: designing in the ability to absorb problems w/o impacting service levels  Recovery Time Objective (RTO): time taken after disruption to restore business processes to service level  Recovery Point Objective (RPO): acceptable amount of data loss measured in time |
| |  |  | | --- | --- | | Category | Example | | Hardware Failure | Network switch power supply fails and brings down LAN | | Deployment Failure | Deploying a patch that breaks a key business process | | Load Induced | DDoS attacks | | Data Induced | Try to convert float 64 to int 16 causing error | | Credential Expiration | SSL/TLS certificate expires | | Dependency | S3 subsystem failure cause other AWS service failures | | Infrastructure | Accidental cut of fiber optic data line | | Identifier Exhaustion | Don’t have sufficient capacity in AZ you requested | |
| AWS Continuum of HA | In decreasing failover & recovery time: 1) Backup & Restore. 2) Pilot Light. 3) Warm Standby. 4) Multi Site |
| 1) Backup & Restore: minimal effort to configure (can use snowball or create 2nd copy)  - Least flexibility. – Analogous to off-site backup storage. – RPO/RTO: hours |
| 2) Pilot Light: cost effective way to maintain a “hot site” concept. - Have replicas of web servers but stopped. When main down, start up replicas and point route 53 to replica  - Suitable for a variety of landscapes and apps.  - Usually requires manual intervention for failover. – Spinning up cloud envs will take mins or hours  - Must keep AMIs up to date w on-prem counterparts. – RPO/RTO: 10s of minutes |
| 3) Warm Standby: all services up and ready to accept a failover faster within mins or secs  - Can be used as a “shadow env” for testing or production staging  - Resources need to be scaled to accept production load. – Still requires some env adjustments but can be scripted. - RPO/RTO: minutes |
| 4) Multi Site: ready all the time to take full production load (effectively mirrored data center)  - Fails over in seconds or less. – No or little intervention required to fail over. - Most expensive DR option. – Wasteful as resources just waiting for primary to fail. – RPO/RTO: near real-time |
| Storage HA options | EBS Volumes: annual failure rate < 0.2% compared to commodity hard drive at 4%  - Availability target of 99.999%. – Replicated automatically within a single AZ. - Vulnerable to AZ failure.  – Easy to snapshot, which is stored on S3 and multi-AZ durable  - Can copy snapshots to other regions as well. – Supports RAID configs |
| RAID0/striping: no redundancy. Write data across multiple hard drives  - Get good reads and good writes as distribute load across multiple I/O devices  RAID1/mirroring: copy identical data to another drive. – Redundancy: 1 drive can fail  - Capacity = 50% as have duplicate copies. - Read and Write worse than RAID0  RAID5: use parity bit to allow recovery of lost info. Requires min 3 drives. – Redundancy: 1 drive can fail  - Capacity = (n-1)/n. – Read same as RAID0, Write worse than RAID1  RAID6: has 2 methods of parity. – Requires min 4 drives. – Redundancy: 2 drives can fail  - Capacity = (n-2)/n. – Read same as RAID0, Write worse than RAID5 |
| EBS Provisioned IOPS SSD (Io1): RAID0 throughput and usable space twice of RAID1 |
| S3 Storage. Standard = 99.99% availability. – Standard IA = 99.9% availability. – 1-zone IA = 99.5 availability. Eleven 9s of durability (99.999999999%)  - Standard & Standard IA have multi-AZ durability. – 1-zone only single AZ durability  - Serves as backing service for EBS snapshots and many other AWS services |
| EFS: file locking, strong consistency, concurrently accessible. – Mount targets are HA  - Each file obj and metadata stored across multiple AZs. – Can be accessed from all AZs concurrently |
| Compute HA options | Up-to-date AMIs are critical for rapid fail-over  - AMIs can be copied to other regions for safety or DR staging  - Horizontally scalable architectures preferred as risk spread across multiple smaller machines  - Reserved Instances only way to guarantee resources will be available when needed  - Route 53 Health Checks provide “self-healing” redirection of traffic |
| DB HA options | In decr operational efficiency but incr control: DynamoDB -> Aurora -> RDS -> EC2 |
| DynamoDB HA: distributes data and incoming traffic across partitions by default  - Partitions are replicated synchronously across 3 AZs in a region  - Global Tables allow for multi-region availability and fault tolerance |
| RDS HA: for multi-AZ RDS, can get synchronous replication from main RDS to 2 backup RDS  - Read replicas different from backup RDS  - In Region A, can have multi-AZ RDS cluster. In Region B, can continuous replicate for read replicas. If regional failure occur, read replica can be promoted to main RDS instance (similar to warm standby)  - If RTO is high (hours), can use Snapshot Recovery = create periodic snapshot of RDS to S3, then restore snapshot in another region if failure. (similar to backup and restore) |
| Aurora HA: main instance and multiple read replicas across AZ within a region  - Data copies are synchronously distributed across these AZs  Aurora Global Databases: allows HA and DR. – 1 primary Region and up to 5 secondary regions  - date replicates from primary to secondary regions w low latency (but not synchronous)  - leverages storage-level replication for transferring data. – secondary region can be promoted in outage |
| Redshift HA: currently only RA3 instances support multi-AZ deployment  - Best option for HA otherwise is a multi-node cluster  - Single-node clusters have to be restored from S3 snapshot on failure  - For RA3 multi-AZ: use Redshift Managed Storage (RMS) hosted in S3. Only compute layer and memory cache that’s stored in Redshift instance |
| Network HA options | By creating subnets in other AZs, you create multi-AZ presence for VPC  Best practice is to create at least 2 VPN tunnels into your Virtual Private Gateway  DX not HA by default, need establish a secondary connection via another DX or use VPN  Route 53’s Health Checks provide basic level of redirecting DNS resolutions  Elastic IPs allow flexibility to change out backing assets w/o impacting name resolution  For multi-AZ redundancy of NAT Gateways, create gateways in each AZ w routes for private subnets to use local Gateway |
| Questions | 1) EBS and DynamoDB tables need backup. Tag resource w Prod, Dev or Test. Dev and Test need back up daily. Prod back up twice daily. Need automate backup and store Prod backups in a 2nd Region  - Amazon Data Lifecycle Manager (DLM) can automate create & delete of EBS snapshots but not for DynamoDB tables. – Instead, use AWS Backup and create 2 backup plans. Configure 1st plan to include all supported resources w Prod tag. Run this every 12 hours. Store backups in a backup vault in a 2nd Region. Configure 2nd backup plan to all supported resource w Dev or Test tags. Run this daily. Store these backups in default backup vault (same Region)  2) Aurora global DB, w primary Aurora DB cluster in 1 Region, secondary Aurora DB cluster in 2nd region. Need minimize latency and ensure read-after-write consistency for DB operations. Want to regularly test DR procedures to minimize data loss and maximize recovery speed across 2 Regions  - Use secondary Aurora DB cluster endpoint to perform read and write operations. Configure write forwarding to forward write operations to primary Aurora DB cluster endpoint. Test DR procedure by using managed planned failover.  3) Create HA: Internet gateway and ALB are HA by default and managed by AWS. NAT gateway only HA within the AZ it is deployed. So for 2 AZ, each need have its own NAT gateway  4) Need backup S3, EBS, EFS, RDS, on-prem data connected by Storage Gateway Volume Gateway in another region:  - Create AWS Backup backup vault in another Region. Create a cross-Region backup plan in AWS Backup by using the vault that includes data that is in S3 bucket, EBS, EFS, RDS and Storage Gateway volumes  To use AWS Elastic Disaster Recovery Service: a) Install the replication agent on the source servers to be replicated. b) Connect source servers to endpoints in the VPC to be used as failover. c) Connect the replication servers to the endpoints  d) Validate the route tables |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Deployment and Operations Management | | | | |
| Types of Deployments | | | | 1) Rolling Deployment: Create new Launch Config w updated AMI. Start terminating old EC2 instances, which will spawn new EC2 w new configs  2) A/B Testing: Use Route 53 to send x% of traffic to 1 version, (100-x)% to version 2  3) Canary Deployment: Deploy V2 for some traffic, if no errors, deploy v2 for rest of traffic  4) Blue/Green Deployment: Deploy V2 whole environment, Route 53 switch over to new landscape |
| Blue Green Methods: - Update DNS w Route 53 to point to new ELB or instance  - Swap Auto-scaling group already primed w new version instances behind the ELB  - Change Auto-Scaling Group Launch Config to use new AMI version and terminate old instances  - Swap env URL of Elastic Beanstalk. – Clone stack in AWS OpsWorks and update DNS |
| Blue Green Contraindication: - Data store schema too tightly couples to code changes  - Upgrade requires special upgrade routines to be run during deployment  - Off-the-shelf products might not be blue-green friendly |
| Continuous Integration & Deployment | | | | CI: merge code changes back to main branch frequently, enabled by automated testing as you go  Continuous Delivery: automating release process to the point where you can deploy at click of button  Continuous Deployment: each code change triggers a series of automated release stages, ultimately deploying to prod w/o human intervention |
| CodeCommit: AWS-managed Git repo  CodePipeline: orchestrates CI/CD events triggered by new code commit  CodeBuild: compiles code, runs tests, produces deployable packages  CodeDeploy: deploys packages to EC2, Elastic Beanstalk, ECS, Lambda, …  New Code CodeCommit CodePipeline (CodeBuild Test CodeDeploy) Instances w CodeDeploy Agent  - CodeBuild artifacts S3 built artifacts CodeDeploy |
| Multi-Acct pipelines: CodeBuild Build artifact (CloudFormation template) Staging acct if no error Prod acct. - CodePipeline assumes a role in staging/prod acct to provision the resources |
| Cloud9: pre-configured, cloud-native development IDE  CodeGuru: automates code reviews, and provide intelligent code recommendations  CodeStar: easily deploy pre-configured CI/CD ecosystems  X-Ray: trace distributed systems and visualized distributed pipelines  CodeArtifact: securely fetch, store, publish packages across organizations |
| Elastic Beanstalk | | | Orchestration service to make it push-button easy to deploy scalable web landscapes  Wide range of supported platforms – from Docker, PHP, Java, Node.js  Multiple Environments within Application (DEV, QA, PRD, …)  Great for ease of deployment, not great if need lots of control and flexibility  Management Layer (app) -> Instances, Web Server, Engines, Monitoring, Scaling (Environments) -> App code (App versions) | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Deployment | What | Deployment Time | Downtime? | Rollback process | | All At Once | New version deployed to existing instances all at once | Least | Y | Manual | | Rolling | New version deployed 1 by 1 to existing instances | Less | N | Manual | | Rolling w additional Batch | Launch new version before taking old version out | Average | N | Manual | | Immutable | Launch full set of new version in separate auto-scaling group, cut over if health check passed | Most | N | Terminate new Instances | | Traffic Splitting | % of traffic routed to new instances (canary) | Most | N | Reroute DNS & terminate new | | Blue/Green | CNAME DNS entry changed when new version up, old version in place until new verified | Most | N | Swap URL | | |
| Cloud-Formation | | | Infrastructure as Code. – Use JSON or YAML to model and provision entire landscapes  - Repeatable, auto deployments and rollbacks. – Nest common components for reusability  - Supports over 300 resource types (components of AWS services)  - Supports creating custom resources via SNS or Lambda | |
| Templates: JSON or YAML text file that contains the instructions for building out the AWS env  Stacks: the entire env described by the template and created, updated, and deleted as a single unit  Change Sets: summary of proposed changes to stack to allow you to see how those changes might impact existing resources before implementing them | |
| Stack Policies: protect specific resources within your stack from being unintentionally deleted or updated  - Add Stack Policy via console or CLI when creating a stack  - Adding Stack Policy to an existing stack can only be done via CLI  - Once applied, Stack Policy cannot be removed (but can be modified via CLI)  - By default, all changes are denied. Need allow updates for all resources then deny specific resource | |
| Deploy infra across accts | | | | IaC: add version control to Cloud Infra. – CloudFormation templates define infra in JSON or YAML  - CloudFormation can be abstracted by frameworks like AWS CDK, SAM or Terraform  - StackSets allows deploy and update a set of resources in multiple accts or regions |
| Use StackSets to define rules in a single/admin acct and deploy to multiple accts and regions  - CloudFormation in admin acct must have permission to assume an execution role in member accts  - StackSets can be defined in admin/management acct. Control Tower baseline (= Blueprints + Guardrails) applied to all OU/accts in organization using StackSets |
| Service Catalog: another way to share approved templates across organization  - Create and govern a curated list of AWS products  - Allows users to provision resources w/o full access to AWS services  - Portfolios can contain many products. – Share portfolios w specific accts or across org  StackSets you push to specific acct. Service Catalog let individual acct choose |
| ECS | |  |  | | --- | --- | | ECS | EKS | | Managed, HA, highly scalable container platform | | | AWS specific platform supporting Docker containers | Compatible w upstream K8s, so easy to lift and shift from other K8s | | Simpler to learn and use | More feature-rich and complex w steep learning curve | | Leverages AWS services like Route 53, ALB, CloudWatch | Hosted K8s platform that handles many things internally | | Tasks = instances of containers that run on underlying compute but more or less isolated | Pods = containers collocated w one another and can have shared access to each other | | Limited extensibility | Extensible via wide variety of 3rd party add ons | | | | |
| Launch type for ECS: EC2 Launch type OR Fargate Launch type | | | |
| API Gateway | | Managed, HA service to host REST APIs. - Backed w custom code via Lambda, as proxy for another AWS Service or any other HTTP API on AWS or elsewhere  - Regionally based, private or edge optimized (deployed via CloudFront)  - Support API Keys and Usage Plans for user identification, throttling or quote management  - Using CloudFront behind the scenes and custom domains and SNI are supported  - Can be published as products and monetized on AWS Marketplace. - Can cache responses | | |
| Management Tools | | | | 1) AWS Config: assess, audit, evaluate configs of AWS resources. – Config Management as part of ITIL  - Creates baseline of various config settings and files, then can track variations against that baseline  - AWS Config Rules to check resources for certain desired conditions. If violations found, resources flagged as noncompliant (backup enabled, CloudTrail enabled, volumes encrypted) |
| 2) AWS OpsWorks: managed instance of Chef and Puppet (popular automation platforms)  - Provide config management to deploy code, automate tasks, configure instances, perform upgrades…  - 3 offerings: OpsWorks for Chef Automate, OpsWorks for Puppet Enterprise, OpsWorks Stacks  - OpsWorks for Chef Automate and Puppet Enterprise are fully managed implementation of each respective platform  - OpsWorks Stacks created by AWS, uses embedded Chef solo client installed on EC2 instances to run Chef recipes  - OpsWorks Stacks support EC2 instances and on-prem servers as well w an agent |
| In OpsWorks Stacks: - stacks = collections or resources needed to support a service/app  - Layers = diff components of the app delivery hierarch (EC2, RDS, ELBs)  - Stacks can be cloned, but only within same region  - OpsWorks is a global service, but to create stack, must specify region and stack can only control resources in that region |
| AWS System Manager | | | | Centralized console and toolset for wide variety of system management tasks  - Designed for managing large fleet of systems – tens or hundreds  - SSM Agent enables System Manager features and support all OSs supported by OS as well as back to Windows Server 2003 and Raspbian (Raspberry Pi deployment of Debian)  - SSM Agent installed by default on recent AWS-provided base AMIs for Linux and Windows  - Manages AWS-based and on-prem based systems via agent |
| |  |  |  | | --- | --- | --- | | Service | Description | Example | | Inventory | Collect OS, app, instances metadata | Which instances have Apache HTTP Server 2.2.x or earlier? | | State Manager | Create states that represent a certain config applied to instances | Keep track of which instances updated to current version of Apache HTTP Server | | Logging | CloudWatch Log agent | Stream logs of web servers to CloudWatch | | Parameter Store | Shared secure storage for config data, connection strings, pws, … | Store and retrieve RDS credentials to append to config file upon boot | | Insight Dashboard | Acct-level view of CloudTrail, Config, Trust Advisor | Single viewport for any exceptions on config compliance | | Resource Groups | Group resource through tagging for organization | Create dashboard for all assets belonging to Prod | | Maintenance Windows | Define schedules for instances to patch, update apps, run scripts | Define hours of 00:00 to 02:00 as maintenance windows for Patch Manager | | Automation | Automating routine maintenance tasks and scripts | Stop DEV and QA instances every Friday and restart Monday morning | | Run Command | Run commands and scrips w/o logging in via SSH or RDP | Run shell script on 54 diff instances at the same time | | Patch Manager | Automates process of patching instances for updates | Keep a fleet at same patch level by applying new patches during next Maintenance Window | |
| Patch Manager uses baseline to determine which patches are auto-approved for being applied  - Default for Windows Server 2008 to 2016: Auto approve patches of “CriticalUpdate”, “SecurityUpdate”, and MSRC severity “Critical” or “Important” 7 days after release |
| System Manager (SSM) Documents   |  |  |  | | --- | --- | --- | | Type | Used w | Purpose | | Command Doc | Run CMD  State Manager | Run CMD uses command docs to execute commands. State Manager uses command docs to apply a config. These actions can be run on ≥ 1 targets at any point during lifecycle of an instance | | Policy Doc | State Manager | Enforces a policy on targets. If policy doc removed, policy action no longer happens | | Automation Doc | Automation | Use automation doc when performing common maintenance and deployment tasks like creating/updating AMI | |
| Enterprise Apps | | | 1) Amazon WorkSpaces: desktop as a service. Full Windows or Linux desktop that you can remote into  2) Amazon AppStream: encapsulates specific apps & allows you to access them (usually through browser) | |
| 3) AWS Connect: Fully managed cloud-based contact center solution w configurable call handling, inbound and outbound telephony, interactive voice response, chatbot tech and analytics  - Can integrate w other enterprise apps like CRM systems  4) Amazon Chime: Online meeting and video conferencing service  - Supports usual conferencing features like desktop sharing, group chat, session recording | |
| 5) Amazon WorkDocs: Online document storage and collaborative platform  - Supports version management, sharing docs and collaborative edits  6) Amazon WorkMail: fully managed email and calendar as a service  - Compatible w Microsoft Exchange (Outlook), IMAP, Android, iOS mail clients | |
| 7) Amazon WorkLink: provide secure access to internal web apps for mobile devices  - When mobile user request app, it’s rendered on a secure machine then image sent to mobile client  8) Alexa for Business: deploy Alexa functionality and skills internally in your enterprise  - Management functionality more appropriate for enterprise organization than buying and provisioning individual Alexa devices | |
| ML Landscape | | | | AI Services: for app developers, no ML experience required. Comprehend, Lex, Polly, …  ML Services: for ML Developers, Data Scientists. SageMaker, Ground Truth, Notebooks,  ML Frameworks & Infra: for ML Researchers, Academics. DL AMIs, AWS Greengrass  - Interfaces (Gluon, Keras). – Frameworks (mxnet, TensorFlow) |
| For AI Services: – Highly scalable and available. – Pay per use. - Leverage using API or AWS SDK  1) Comprehend: NLP service to find insight and r/s within text. (Sentiment analysis)  2) Forecast: time-series forecasting. 3) Lex: chatbot. 4) Personalize: recommendation engine  5) Polly: text to speech. 6) Rekognition: image classification, object detection  7) Textract: OCR. 8 ) Transcribe: speech to text. 9) Translate: translate text |
| IoT Landscape | | | | For Device Connection (connecting to and responding to messages w IoT devices):  - AWS IoT Core. – AWS IoT 1-Click. – AWS IoT Events. – AWS IoT Greengrass. – AWS IoT Things Graph  For Device Management (securing & managing IoT devices):  - AWS IoT Device Management. – AWS IoT Device Defender  For analytics & visibility: - AWS IoT Analytics. – AWS IoT SiteWise |
| AWS IoT Core: route IoT messages by the trillions. – Ingest messages from IoT devices  - Publish messages to diff AWS services like S3, Lambda, IoT Analytics  - Supports MQTT (Message Queuing Telemetry Transport) protocol |
| AWS IoT Events: trigger alerts when events occur. – Monitor sensor data from IoT devices  - Define conditional logic to determine when alarms should be triggered. – Detect and trigger events |
| AWS IoT 1-Click: simple integration for compatible devices.  - Directly trigger Lambda functions w 1-Click compatible IoT devices  - Compatible devices are pre-provisioned w certificates for secure access  - Manage and group your 1-Click devices |
| AWS IoT Things Graph: develop models to integrate devices and web services  - Design low-code logical workflows for IoT devices  - Drag and drop interface allows for easy integration and visibility  - Save device types and groups of services as reusable models |
| AWS IoT Greengrass: build IoT Device software. - Install Greengrass client software on IoT devices  - Deploy Lambda functions, Docker images or ML models w local inference  - Exchange data w IoT Core or other AWS services |
| AWS IoT Device Management: device registry that allows for bulk registration, organizing, devices into groups, and securely send over-the-air (OTA) firmware updates to devices |
| AWS IoT Device Defender: managed security service that continually audits IoT configs.  - Uses ML anomaly detection to publish alerts in response to device behavior |
| AWS IoT Analytics: develop reports from time-series data  - Aggregate and enrich messages from IoT devices at scale  - Data stored in time-series data for near-live analysis. - Build reports using SQL or ML analysis insights |
| AWS IoT SiteWise: monitor IoT devices on the edge  - Install SiteWise Edge software in data center. – Collect data from IoT devices locally  - Model and monitor devices w/o need for internet access. – Integrate easily w local/cloud-based apps |
| Questions | | | | 1) Provide user access to resources in diff accts: - Use cross-acct IAM roles and attach permissions  2) AWS Organization w all features. Want enforce standardized set of tag names and values and correct noncompliant tags:  - Create tag policies in Organizations for required tags. Attach tag policies to organization entities as required. Use AWS Resource Groups to find noncompliant tags. Correct tags by using AWS service that created the resources. Use SCP to enforce use of tags across Organization  3) App allow access on TCP port 22 from private IP of bastion host. SG of bastion host allow access to TCP port 22 from 0.0.0.0/0. Want to eliminate brute force SSH login to bastion host, retain log of commands run, retain ability to forward ports  - Configure app instances to communicate w AWS Systems Manager. Grant access to system admin to use Session manager to establish a session w app instances. Terminate bastion host |

|  |  |  |  |
| --- | --- | --- | --- |
| Cost Management | | | |
| Concepts | Capital Expenses (CapEx): money spent on long-term assets  Operational Expenses (OpEx): money spent for on-going costs. Usually considered variable expenses  Total Cost of Ownership (TCO): comprehensive look at entire cost model of a given decision, including both hard and soft costs  Return on Investment (ROI): amt an entity expect to receive back within a certain time given an investment | | |
| Cost Optimiza-tion Strategies | 1) Appropriate Provisioning: - only provision resources you need. – consolidate where possible for greater density and lower complexity. – CloudWatch can help monitor utilization  2) Right-Sizing: - use lowest-cost resource that still meets technical specs.  – Architecting for most consistent use of resources is best versus spikes and valleys  - Loosely coupled architectures using SNS, SQS, Lambda, DynamoDB can smooth demand and create more predictability and consistency  3) Purchase Options: - Reserved Instances for permanent apps or needs  - Spot instances for temp horizontal scaling.  – EC2 Fleet lets you defined target mix of On-Demand, Reserved, Spot instances  4) Geographic Selection: - pricing vary from region to region.  – Potential savings by locating resources in a remote region if local access not required  - Route 53 and CloudFront can be used to reduce potential latency of remote region  5) Managed Services: RDS, RedShift, Fargate, EMR  6) Optimized Data Transfer: only data out cost money.  – DX can be more cost effective depending on volume & speed | | |
| Tagging & Resource Groups | Tags: arbitrary name/value pairs you assign to any AWS assets as metadata  - Tagging strategies can be used for Cost Allocation, Security, Automation, access controls to certain resources in IAM. – Most resources can have up to 50 tags  - Enforce standardized tagging can be done via AWS Config Rules or custom scripts. | | |
| Resource Groups: groupings of AWS assets by tags (E.g. DEV, QA, PRD, by project, by department)  - Create custom consoles to consolidate metrics, alarms, config details around given tags | | |
| Managing Costs across Accts | Cost and Usage Reports: get insights into organization’s spending  - Generate CSV files to track costs by acct, service, tags.  – W consolidated billing enabled, track spending across Organization from management acct  - Adjust granularity to hourly, daily, monthly. – Analyze & visualize reports w Athena, Redshift, QuickSight | | |
| Centralized Budget Alerts: in management acct, access AWS Budgets, can create budget and filter by acct | | |
| Tagging Strategy | | |
| Spot & Reserved Instances | Reserved Instances: purchase (or agree to purchase) usage of EC2 instances in advance for significant discount over On-Demand pricing. – Provides capacity reservation when used in specific AZ  - AWS Billing auto applies discounted rates when launch an instance that matches your purchased RI  - 3 RI types: Standard, Convertible, Scheduled. – If don’t need RI’s, can sell on Reserved Instance Market  – Can share across multiple accts within Consolidated Billing | | |
| |  |  |  | | --- | --- | --- | |  | Standard | Convertible | | Terms | 1 year, 3 year | | | Average discount off On-Demand | 40%-60% | 31%-54% | | Change AZ, Instance Size, Networking Type | Yes via API or console | | | Change instance family, OS, Tenancy, Payment Options | No | Yes | | Benefit from Price Reductions | No | Yes | | Sellable on Reserved Instances Marketplace | Yes (only deposit in US bank acct) | Coming Soon | | | |
| RI Attributes: - Instance Type = designates CPU, memory, networking capability  - Platform: Linux, SUSE Linux, RHEL, Microsoft Windows, Microsoft SQL Server  - Tenancy: Default (shared) tenancy or Dedicated tenancy  - AZ (optional): If AZ selected, RI reserved and discount applies to that AZ (Zonal RI).  If no AZ specified, no reservation created but discount applied to any instance in the family in any AZ in the region (Regional RI). Can change Zonal to Regional RI via console or ModifyReservedInstance API | | |
| Spot Instances: Excess EC2 capacity AWS tries to sell on a market exchange basis  - Customer creates a Spot Request and specifies AMI, desired instance types and other key info  - Customer defines highest price willing to pay for instance. If capacity constrained and others willing to pay more, your instance might get terminated or stopped  - For “One Time Request” , after instance provision, if killed, ephemeral data lost  - For “Request and Maintain”, instance can be configured to Terminate, Stop or Hibernate until price point can be met again. – For “Duration-based”, instance provision for finite duration | | |
| Dedicated Instance:  - virtualized instances on hardware just for you  - May share hardware w other non-dedicated instances in same acct  - Available as On-Demand, Reserved Instances, Spot Instances  - Cost additional $2 per hour per region | | Dedicated Host:  - Physical servers dedicated to just your use  - Control over which instances deployed on that host  - Available as On-Demand / Dedicated Host Reservation  - Useful if have server-bound software licenses that use metrics like per-core, per-socket or per-VM  - Ea dedicated host can only run 1 EC2 instance size & type |
| Cost Management Tools | | AWS Budgets: set pre-defined limits and notifications if nearing/exceed budget  - Can based on Cost, Usage, RI Utilization, RI Coverage | |
| Consolidated Billing: enable a single Payer acct that’s locked down to only those who need access  - Economies of scale by bringing tgt resource consumption across accts | |
| Trusted Advisor: runs series of checks on resources and proposes suggested improvements  - Can help recommend cost optimization adjustments like RI or scaling adjustments  - Core checks available to all customers.  – Full Trusted Advisor benefits requires Business or Enterprise support plan | |
| Questions |  | | |