AWS APN Technical ivan notes - 2022

1. Intro

- What is cloud computing?
  - on demand delivery of computer power, database, storage, applications, and other IT resources via the internet with pay as you go pricing
- Benefits
  - Agility, elasticity, cost savings, deploy globally in minutes (BROMM)
- Regions
  - completely isolated from each other, certain resources tied to regions
  - AZs - isolated from other AZs, high-speed low latency connection between AZs within a region

2. Core Tech

- Compute
  - EC2 - resizable compute capacity, building blocks - apply AMIs to customize, configure auto scaling and load balancing, only pay for what is used
    - Benefits of EC2: elasticity, control, flexibility, integrated, reliable, secure, cost-effective, easy
    - General purpose, compute optimized, memory optimized, accelerated computing, storage optimized
  - ASG - increase or decrease number of instances
    - auto adjusts resource capacity, define where to deploy resources, specify VPC and subnets
  - ELB - distribute incoming traffic
    - auto distribute traffic across multiple EC2 instances, increase availability and fault tolerance, configure health checks, offload encryption and decryption
    - ALP - app layer
NLB - network layer
Gateway load balancer - 3p virtual appliances
  - ECS - run apps on a managed cluster
  - EKS - run K8s apps on AWS and on-prem
  - Lambda - run code in response to events
- Storage
  - EBS - persistent block-level storage
    - network-attached block storage for use with EC2, persist independently from instance, used like physical drive, automatically replicated, attached to any instance in the same AZ, one EBS volume to one EC2 instance, one instance to many EBS volumes, EBS volumes can retain data after EC3 instance termination, allow point-in-time snapshots to S3 GiB increments
  - S3 - durable, scalable object storage
    - infinite scalability, greater analysis, faster data retrieval, 11 9's durability and 99.99% availability
    - standard > standard IA > one zone IA > glacier
  - Glacier - data archiving and backup
  - Storage Gateway - integrate cloud storage with on-site workloads
  - EFS - file storage for EC2 instances
  - FSx - file storage for widely-used systems
- Databases
  - RDS - cost-efficient and resizable capacity
  - DynamoDB - fast and predictable performance
  - ElastiCache - fast, managed information retrieval
  - AWS Managed: easy to set up, manage, maintain, reduce management tasks, push-button HA, auto backup/recovery, scale up or down based upon pattern
  - On EC2 benefits: more control / flexibility, OS access, need features of specific application
- Networking
  - VPC - build a virtual network in the cloud
    - networking layer for AWS resources, a virtual network dedicated to a customer's AWS account
    - Subnet - a range of IP addresses in a VPC
    - Flow Logs capture network flow information
    - host-based firewalls - OS firewalls
  - Security Groups - control access to instances
  - NACL - control access to subnets
  - Route 53 - DNS, route end users to internet applications
- Security
  - inherit benefits from AWS data center and network architecture, similar to on prem without maintaining facilities and hardware, can be easily automated, inherit all best practices of AWS
  - Security, Identity, and Compliance Services
    - infra protection: network firewall, shield, WAF, firewall manager
    - IAM: IAM, SSO, Orgs, RAM, directory service, cognito
    - Detection: guardduty, inspector, cloudtrail, security hub, config, IoT device defender
    - Data Protection: macie, cloudhsm, secrets mgr, KMS, ACM
    - IR: detective, cloudendure disaster recovery
    - Compliance: artifact, audit mgr
  - AWS shared Responsibility Model
    - AWS responsible for security of the cloud, customer for security IN the cloud
  - IAM - securely manage access to AWS services and resources
    - fine-grained access control to AWS resources, MFA, ability to analyze access, integration with corporate directories
  - Cloud Compliance
    - Sharing Info: industry certifications, security and control practices, compliance reports directly under NDA
    - Assurance Programs: certifications and attestations, laws, regs, privacy, alignments and frameworks
- AWS Management Interfaces
  - Management Console, CLI, SDKs - built on common, REST-like API that servers as foundation of AWS

3. Solution Design

- Well Architected Framework
  - Framework to ensure infra is secure, high-performing, resilient, efficient, sustainable.
  - 6 Pillars
    - Security
    - Cost optimization
    - Operational excellence
    - Reliability
    - Performance efficiency
    - Sustainability
- Cloud Adoption Framework (AWS CAF)
  - provides smooth cloud migration, split into two perspectives: business capabilities & technical capabilities - perspectives create inputs to use to create CAF action plan
  - Business Capabilities perspectives
    - Business: ensure IT aligns with business, creates strong business case for cloud adoption
    - People: support change mgmt strategy, evaluate org structures and roles, evaluate new skill and process reqs, identify gaps, prioritize training
    - Governance: focus on skills and processes, ensure business values are max'd and risks are minimized, update the staff skills and processes, measure cloud investments to evaluate business outcomes
  - Technical Capabilities perspectives
    - Platform: implement new solutions in the cloud, migrate on-prem workloads to cloud, understand and communicate structure of IT systems and their relationships, describe the architecture of the target state environment in detail
    - Security: meet security objectives for visibility, auditability, control, and agility, structure the selection and implementation of security controls
    - Operations: enable run, use, operate, and recover IT workloads, define how business is conducted, align with and support the business operations, define current operating procedures
- Migration Strategies: 7Rs
  - Rehost: lift and shift - recreating on prem on AWS, automating with tools such as AWS app migration service, easier to optimize and re-architect apps after migration
  - Relocate: hypervisor-level lift and shift - migration specific to VMware cloud on AWS, e.g. migrate hypervisor host Oracle DB to VMware cloud on AWS
  - Replatform: lift, tinker, and shift – retaining the core architecture, making targeted AWS cloud optimizations, e.g. migrating DBs to RDS, migrating apps to EB
  - Refactor: modernize – re-imagining how the app is architected and developed, using cloud-native features
  - Retire: shutting off non-useful apps, reducing spend, management, and security
  - Retain/Revisit: keeping certain apps on-prem
  - Repurchase: moving workflows to SaaS
- Architecture Best Practices
  - Design for failure and nothing fails
    - Avoid SPoF, multiple instances, multiple AZs, separate single server into multiple tiered app, for RDS, use multi AZ feature
  - security in every layer
    - Encrypt data at rest & transit, least priv, both sec groups & nacls, consider advanced security features & services
  - Leverage different storage options
    - Move static web assets to S3, use CloudFront to serve globally, store session state in DynamoDB, use ElastiCache between hosts and DBs
  - Implement elasticity
    - Implement auto scaling policies, architect resiliency to reboot and relaunch, leverage managed services like S3 and DynamoDB
  - Think parallel
    - Scale horizontally, decouple compute from session/state, use ELB, right-size your infra
  - Loose coupling sets you free
    - Instead of single, ordered workflow, use multiple queues, use SQS and SNS, leverage existing services
  - Don't fear constraints
    - Rethink traditional constraints, need more RAM?, better IOPS for DBs?, response to failure?
- Addressing Customer Challenges
  - Customer's current architecture, customer challenges, ideas for services and features, any use case studies to leverage
- Resources
  - AWS Customer Success for use cases
  - AWS Quick starts - arch accelerators

4. Presenting Solutions to Customer

- Presenting the Solution - whiteboarding (may be more than one)
  - Best Practices
    - Prepare - 3hrs for every 1hr of presenting
    - Anticipate - conversations
    - Differentiate - every customer interaction is different
    - Stay on Message - do not get thrown off course
    - DO NOT: use words like definitely, never, guaranteed - do not over commit, use acronyms or technical jargon, focus on technology, focus on the short/mid-term, read the slides
  - Handling Objections
    - Connect - identify with customer's POV
    - Condense - identify the core concern
    - Continue - Take steps toward resolution
    - Best Practices: data-driven approach, use case studies, dive deep, have backbone, keep the momentum going
- Discovery - understanding the customer challenges
  - Research customer's business, determine market segment, identify industry trends, identify customer's competitors, research recent news, research customer relationship with AWS
  - Encourage detailed conversation - open ended questions, targeted questions
Five Whys - dive deeper, uncover real desired outcomes
Whiteboarding - keep track of conversation, illustrate workflows and ideation
  - Discovery meeting: open-ended question, digging deeper, why?, why?, digging deeper, request technical detail, suggest follow
- Delivering a POC - evaluating the solution in the customer's environment
  - Building a POC
    - Practical example of solution, evaluation mechanism, evaluation tool
    - Determine what success looks like, include any mods, consult as necessary, collect reqs: networking and security, app code, databases
    - Skills: building on AWS, migrating data, validating and testing
    - Resources: cost of running POC

5. Looking Ahead

- Going to Production
  - Best practices: Involve AWS account team (SA or TAM), customer-specific regulatory reqs
  - Do well architected review with tool: architectural guidance continuous review, improved architecture
- Modernization
  - Modernize to drive growth: retire expensive legacy solutions, reduce TCO, improve cost optimization, gain agility through automation, free up resources to drive innovation
  - Containers: package code, configs, and dependencies into a single object, share an OS, run as resource-isolated processes
    - Use cases: microservices, batch processing, ML, hybrid apps, app migration to the cloud, PaaS
  - Serverless: no provisioning, maintaining, and administering servers, AWS handles fault tolerance and availability, focus on product innovation
  - Data lakes and analytics: data in different silos can be difficult to access and analyze, store data in a "data lake", easy to read data and obtain insight
    - Interactive analysis: Athena
Big Data Processing: EMR
Data Warehousing: Redshift
Real-time analytics: Kinesis
Operational Analytics: OpenSearch
Dashboards and viz: QuickSight
Cataloging and Securing: Lake Formation
- Migration Considerations
  - Assessment: identify readiness, identify potential business outcome
  - Readiness and Planning: analyze environment, determine migration strategies, create a well-arch landing zone (POC)
  - Migration and Modernization: design, migrate, and validate each app, automatic or manual, migrate data, modernize
  - Minimum viable product (MVP): Just enough features to satisfy requirements, avoid building solution where you only discover if there is success at the end, start with something basic and gather feedback as you get more complex -- this is different than POC