coggle
made for free at coggle.it

AWS Cloud Practicioner

- Technology
  - Cloud Managment
    - Service and Workload Configuration
      - **Lunch Wizard** - tool for guided deployment of some pre-build AWS applications like EKS
      - **AWS Batch** - manage and schedule your processes
      - **AWS Compute Optimizer** - AI which scan CloudWatch metrics and resource configuration and gives recommendations
      - Service Configuration
        - **AWS Config** - control service configuration from one place (check if it pass company policy) it's more like monitoring configuration tool
        - **AWS License Manager** - control account software licences from one place
      - **AWS System Manager** - central place to manage fleets/applications
        - **Secret Manager** - similar to Parameter store but values not shared publicly
        - **Parameter Store** - keep parameters for your app in one place
        - **AppConfig** - manage configuration for you app for many environments. Versioning/API/CRUD/live changes
        - **AWS OpsWork** - alternative to System Manager via Puppet Chef
      - **RAM or AWS Resource Manager** - manage access to your resources for multiple accounts
      - **Service Catalog**
    - **IaaC** - Infrastructure as a Code - It helps us use this same configuration(without custom problems for each account)
      - **Cloud Development Kit** - service build on top CloudFormation it helps you write infrastructure in chosen language
      - **CloudFormation** - AWS IaaS Solution
      - Terraform
    - **Proton** - Increase your impact with self-service infrastructure templates for serverless solutions Lambda, containers
    - **Tags** - tags everywhere
    - **AWS Organisations** - Central management for multiple AWS accounts
      - Global Billling
      - Global Services
      - Global Backups on accounts
      - Global Policies
      - Global Configurations
      - **Control Tower** - it helps to create multi-account environment with best practices using AWS Organisations behind the scenes (support tool)
  - App Integration
    - **Step Function** - you are able defines some steps which can help create complex tesks/ code, is used to orchiestrate workflows
    - **Event Bridge**
    - **SNS** - Simple Notification Service
    - **SQS** / **Amazon MQ** - Simple Queue Service
    - **SES** - Simple Email Service
    - **CloudMap** - helped for microservice architecture, you are able to assign names to resources and other services are able to track resources of each service which can be helpful for microservice architecture
  - Monitoring
    - **VPC Flow Logs**
    - **Enabling Monitoring Services**
    - **CloudWatch** - default service for monitoring, collecting logs and analyze Data, you can enable CloudWatch Agent for detailed logs, also you can set alarms if some metrics are too higher
    - **X-Ray**- it's tool which add metadata across requests in network and you are able spy single request how is handle and from where comes from
  - Developing
    - **Amplifie** - This is set of apps which allows you to build fullstack app like Cognito + API Tools etc.
    - **Cognito** - OAuth server for end-users
    - **Cloud Development Kit** - up-build service on CloudFormation, allows write IaS in chosen language
    - **Cloud9** - AWS IDE
    - **CodeCommit** - AWS git repository
    - **CodeArtifacts** - code packages registry likne(Verdacio,npm,jfrog,artifactory)
    - **CodeBuild** - Test & Build App Ci tool where output is via S3
    - **CodeDeploy** - Deploy Code to EC2,ECS or Lambda
    - **CodePipeline** - Define all deploy steps and automate them using Code-family tools
    - **CodeStar** - simplified CodePipeline
    - API TOOLS
      - **Api Gateway** - REST API on AWS
      - **AppSync**- GraphQL on AWS
  - Simplifying Deployment
    - **Elastic Beanstalk** - easy-use deployment package np web server on node.js with mysql
    - **Elastic Lightsail** - less configuration, easy-use pre-configured package of host environment
    - **Copilot** - CLI for containerized app
    - **AppRunner** - easy deploy of containerized app
  - Cloud Migration and Hybrid Solutions
    - Key Migration Services
      - Migration Hub - central hub to track all migration process
      - Aplication Migration Service - automated server migration (agget softwareanalyzes + replicates system)
      - Database Migration Service - automated database migration(homo and heterogeneous + schema conversion)
      - DataSync - synch. on-premises data with cloud ones via local NFS
      - Snow Familly
      - Transfer Familly - maps FTP endpoints to S3 or EFS storage
    - Hybrid Cloud Computing Services
      - Outposts - AWS infrastructure added to your local
      - Snow Familly
        - Snowball - physical device to get on-premises data offline
      - ECS / EKS Anywhere - tooling to run local containerised /k8s infra
      - DataSync - sync your local data with cloud
      - Interface for enabling on-permises workload to use S3
      - System Manager - manage your large-scaling server fleets
    - Connections
      - Client VPN
      - Site-to-Site VPN
      - Direct Connect - your direct connection with AWS netwrok
  - Networking
    - **S3 Transfer Accelerator** - it helps to boost file transfer via AWS edge network
    - **Global Accelerator** - service which boost network requests, it Load balancing traffic between regions!
    - **VPC**
      - Request Management
        - **Internet gateways** allow to-way access to internet (only for public subnets)
        - **NAT gateways** enable outgoing(for fetch some packages/dependencies) internet requests only on subnet level
        - Endpoints(PrivateLink) connect AWS services to VPC's f.ex. with S3
        - Network ACL's allow or deny requests on subnet level
          - One NACL can be reused with other subnets
          - Firewall for entire subnet
          - checking ingress / egress requests
          - Stateless: requests and responses are decoupled managed
      - Network Management
        - Elastic IPs can be used for fixed IP addresses but with limit to quatas
        - Every VPC has CIDR block(range if IP's), each subnet is part of this VPC range
        - EC2 instances has auto assigned public and private IP's
        - VPC peering(pair connection) or transit gateways(many connection between VPC's) can connect to VPC to other VPC's (transit tools between few VPCs)
      - Your own network in cloud to group for grouping Instances
      - Can have public / private subnets
      - VPC have to have min. 2 subnets and 1 route table(which control subnets visibility)
      - Every VPC has CIDR block(range if IP's), each subnet is part of this VPC range
      - Scaling
        - **AWS Auto Scaling**
          - It's free
          - It can help you quickly scale your resources (add/remove instances) according to your metrics from CloudWatch
        - **ELB** - Elastic Load Balancer - this is service which helps you to distribute traffic in network equally to your instances of service which is managed by Auto Scaling Service, also it health-check your instances, it's good for HTTP trafic
        - **NLB** - Network Load Balancer - It is service which operate on non HTTP logic where you can distribute traffic on subnet level depending on workload, it doesn't have some many configuration as ELB cause is designed to application which compute data
    - **Route53** - DNS tool
    - **CloudFront** - this is CDN for static files to cache it near to user on edge locations
  - Computing
    - **EC2** - Elastic Compute Cloude
      - It is virtual machine
      - Connecting via SSH or EC2 Instance Connect
      - You can set-up firewall which call **Security Group**
        - For single EC2
        - Stateful - Responses are always allowed if request passed
        - Re-usable - can be attached to other instances
        - Multiple instances can have different security groups
      - You can choose your own AMI(Amazon Machine Image) which contain operation system and other pre-configured software
      - You rent "slices" / resources of real machines(instances)
      - Each Instances is fully isolated from other instance
    - Lambdas - serverless code execution
    - Containerise
      - ECR - Elastic Container Registry
      - ECS - Elastic Container Service
      - EKS - Elastic Kubernetes Service
      - Fargate - serverless build on ECS/EKS system where you are not worried about cluster resources
  - Data Storage
    - EC2 Instance Store
      - alternative to EBS but not default
      - hard drive which is a part of Machine / rack in Data Center
    - **EFS** - Elastic File System
      - default choice
      - this is (network) file system
      - have pre-formatted and configured file system
      - we can create custom structure and work on any files
      - multiaccess and auto scalable
    - FSx Lustre
      - File system optimased to high-performance data access
      - alternative to EFS
    - **EBS** - Elastic Block Storage
      - this is managed hard-drive hub
      - scope only to EC2 instances
      - unformatted hard drive so there is need format and structure before usage
      - we have to create custom structure and we are ready to store any files
      - Core features
        - diffrent types SSD / HDD
        - Elastic Volume - autoscaling of volume
        - Snapshots - keeping versions of store
        - MultiAttachment - can be attached to many EC2
    - **S3** - Simple Storage Service
      - Storage Classes
        - Reduced redundancy - noncritical data, frequently access data
        - **S3 Glacier** - there is 3 options for glacier data mainly it is al about archive data
        - Standard - frequent access
        - Inteligent Tiering - Data with changing or unknown access pattern
        - Standard IA - infrequent access data once at month
      - Independent object storage
      - Access with or without other services (CLI,SDK,API,HTTP)
      - Unlimited scaling build-in
      - Can be use as Data Lake for Analytics
      - Advance features
        - Static Http Server
        - Versioning
        - Lifecycle Management - can be transition between storage classes
        - Inventory & Analytics
        - Compliance & Object Lock - prevent object deletion / changes chmod
        - Replication
        - Data Encryption
    - Databases
      - RedShift- for analytics
      - **RDS** - Relational Database Service
      - **ElasticCache** - In memory Database like Redis
      - **DynamoDB** - No relational Database
      - **Aurora DB** - Relational standalone DB supports Mysql and PostgreSQL
      - Other Services
        - MemoryDb - in memory Database here you can choose Redis Solution
        - DocumentDb- document database (nested data structure)
        - Keyspaces - wide column database
        - Neptun - graph Database
        - TimeStream - Time series database
        - Quantum Ledger Database - Immutable log of data changes (crypto)
  - Data Analytics & Science
    - Transforming Data Tools
      - **EMR** - Elastic Map Reduce
        Similar to glue service but doesn't provide aws solution to transform data instead it helps to create our own bid data compute cluster which transform our data in language which we want to use
      - **Glue** - serverless service which can extract, map, transform, load data across stores which keep unstructured data to db which contain strucutured data
    - **Amazon Kinesis** - Streaming Data Tool
      - Kinesis Firehose
      - Kinesis Data Streams
      - Kinesis Data Analytics
    - **Amazon QuickSight**
      BI Business Intelligence Service which help to create analysis, charts reports, dashboard for product department
    - Data Lakes
      - **Amazon Athena**
        This is a service which help create queries in Mysql into S3 bucket and works with Data Lakes
    - Data Warehouse
      - **Redshift** - relational database optimised to analytics purposes
        - queries
    - Visualisation and Searching
      - **OpenSearch**
        Searching and analysing data
      - **Grafana**
        interactive visualisation of data
      - **Cloud Search**
        works like search engine for end users on big data
- Billing and Pricing
  - Tools
    - AWS billing
      - Cost Managment
        - Cost Explorer - Visualize and Explore Your AWS Costs and Usage
        - Reports
      - Budgets - create your budgets and alerts
      - Billings
  - Pricing Models
    - **Reserved Instances**
      - must be bought separately
      - pay in advance
      - discounts
      - the best for long-term, commitment on period time
    - **Saving Plans**
      - must be bought separately
      - pay in advance
      - discounts
      - any compute service f.ex lambdas
      - the best for long-term, commitment on resource time usage
    - **Spot Instances**
      - must be selected
      - can be reclaimed
      - spared with others
      - discounts
      - price depends from configuration
      - the best for workloads which can be interrupted
    - **On Demand Instances**
      - default option
      - pay for usage (pay as you go)
      - price depends from configuration
      - no discounts
- Cloud Concepts
  - Infrastructure
    - Edge Locations
    - Regions
      - **Local Zones** - small data centres in metropolitan to be close the user as it is possible
      - **WaveLenght** - similar for Local Zones but for 5G network, embedded AWS services for 5G
      - **Outposts** - this is a server near you company infrastructure which can serve AWS services close us it is possible of your network, AWS Infrastructure in your on-presmises
    - Availablies Zones
  - Concepts
    - **Designing for failure** yields a self-healing infrastructure that acts with the maturity that is expected of recent workloads. Preventing emergency calls guarantees a base level of satisfaction for the service-owning team. This also removes a level of stress that can otherwise grow into accelerated attrition.
    - **Right sizing** is the process of matching instance types and sizes to your workload performance and capacity requirements at the lowest possible cost. It's also the process of looking at deployed instances and identifying opportunities to eliminate or downsize without compromising capacity or other requirements, which results in lower costs.
  - Benefits of Cloud Computing
    - **Flexible**
      AWS enables you to select the operating system, programming language, web application platform, database, and other services you need. With AWS, you receive a virtual environment that lets you load the software and services your application requires. This eases the migration process for existing applications while preserving options for building new solutions.
    - **Secure**.
      AWS utilizes an end-to-end approach to secure and harden our infrastructure, including physical, operational, and software measures. For more information, see the AWS Security Center.
    - **Reliable**
      With AWS, you take advantage of a scalable, reliable, and secure global computing infrastructure, the virtual backbone of Amazon.com's multi-billion dollar online business that has been honed for over a decade.
    - **Easy to use**
      AWS is designed to allow application providers, ISVs, and vendors to quickly and securely host your applications – whether an existing application or a new SaaS-based application. You can use the AWS Management Console or well-documented web services APIs to access AWS's application hosting platform.
      - **Increase speed and agility** – In a cloud computing environment, new IT resources are only a click away, which means that you reduce the time to make those resources available to your developers from weeks to just minutes. This results in a dramatic increase in agility for the organization, since the cost and time it takes to experiment and develop is significantly lower.
    - **Cost-Effective**
      You pay only for the compute power, storage, and other resources you use, with no long-term contracts or up-front commitments. For more information on comparing the costs of other hosting alternatives with AWS, see the AWS Economics Center.
      - **Stop spending money running and maintaining data centers** – Focus on projects that differentiate your business, not the infrastructure. Cloud computing lets you focus on your own customers, rather than on the heavy lifting of racking, stacking, and powering servers.
      - **Trade fixed expense for variable expense** – Instead of having to invest heavily in data centers and servers before you know how you're going to use them, you can pay only when you consume computing resources, and pay only for how much you consum
      - **Benefit from massive economies of scale** – By using cloud computing, you can achieve a lower variable cost than you can get on your own. Because usage from hundreds of thousands of customers is aggregated in the cloud, providers such as AWS can achieve higher economies of scale, which translates into lower pay as-you-go prices.
    - **Scalable and high-performance**
      Using AWS tools, Auto Scaling, and Elastic Load Balancing, your application can scale up or down based on demand. Backed by Amazon's massive infrastructure, you have access to compute and storage resources when you need them.
      - **High Availability** Most providers of real-time communications align with service levels that provide availability from 99.9% to 99.999%. Depending on the degree of high availability (HA) that you want, you must take increasingly sophisticated measures along the full lifecycle of the application.
      - **Stop guessing capacity** – Eliminate guessing on your infrastructure capacity needs. When you make a capacity decision prior to deploying an application, you often end up either sitting on expensive idle resources or dealing with limited capacity. With cloud computing, these problems go away. You can access as much or as little capacity as you need, and scale up and down as required with only a few minutes' notice.
      - **Elasticity**
        The ability to acquire resources as you need them and release resources when you no longer need them. In the cloud, you want to do this automatically.
  - Well Architected Framework
    - Tools
      - **The Well Architected Tool** - tool which allow you to go through all pillars and answer on questions. It will helps you improve your infrastructure.
      - **Automatic Cloud Resources Analytics** - automated tools which can go through your infrastructure and recommend some amendments
    - Fillars
      - **Operational Excellence Pillar** focuses on running and monitoring systems, and continually improving processes and procedures. Key topics include automating changes, responding to events, and defining standards to manage daily operations.
      - **Security Pillar** focuses on protecting information and systems. Key topics include confidentiality and integrity of data, managing user permissions, and establishing controls to detect security events.
      - **Reliability Pillar** The reliability pillar focuses on workloads performing their intended functions and how to recover quickly from failure to meet demands. Key topics include distributed system design, recovery planning, and adapting to changing requirements.
      - **Performance Efficiency Pillar** The performance efficiency pillar focuses on structured and streamlined allocation of IT and computing resources. Key topics include selecting resource types and sizes optimized for workload requirements, monitoring performance, and maintaining efficiency as business needs evolve.
      - **Cost Optimization Pillar** The cost optimization pillar focuses on avoiding unnecessary costs. Key topics include understanding spending over time and controlling fund allocation, selecting resources of the right type and quantity, and scaling to meet business needs without overspending.
      - **Sustainability Pillar** The sustainability pillar focuses on minimizing the environmental impacts of running cloud workloads. Key topics include a shared responsibility model for sustainability, understanding impact, and maximizing utilization to minimize required resources and reduce downstream impacts.
- Security and Complaince
  - Security Hub - center place for security stuff
  - **The Shared Responsibility Model**
  - Network protection
    - Firewalls
      - **NACL** - ACL for subnets
      - **WAF** - prevent SQL Injection, XSS etc
      - **AWS Shield** - prevent DDoS
      - **Security Groups** - ACL for EC2
    - **Firewall Manager** - central firewall managing place
  - Data Protection
    - **KMS** - Key Management Service, CloudHSM - cryptographic keys to encrypt data
    - Secret Manager
    - Amazon **Macie** - discover vulnerabilities on AWS accounts, it scans S3 bucket using AI to find sensitive, unsecured data and shows them
    - **ACM** - Amazon Certification Manager
  - Aplication Protection
    - **Inspector** - it's AI scanning of workloads(EC2, DB) to check if there are any vulnerabilities and exposure them to give you oportunity to take an action
    - **Detective** - it's tool which looking for suspicion traffic which is made by accounts or resources, you can analyse what bad actor did and how to handle it.
  - Account Protection
    - AWS Organisations
    - **AWS Config, Audit Manager** - you can manage here compliance rules for legal case
    - **GuardDuty** - AI which monitor suspicious IAM account activity, and create alerts if detect something
    - **AWS Artifact** - it's tool where you can download report of compliance to prove that you are good with company policies which you set f.ex. in AWS Config
    - **RAM** - Resource Access Manager
    - SSO - Single Sign On
    - **CloudTrail** - it can help you track actions taken by AWS IAM account
    - **IAM** - Identity and Access Manager
      - Users and User Groups
      - Policies nad Permissions
      - Roles
      - MFA