System-design-2

This document attempts to detail the implementation of the cloud architecture as described in the scenario.

General Config:
A VPC will be created with 2 sets of private and public subnets in different availability zones. This will allow for multi-AZ failover for deployed AWS services and other applications.

As much as possible, services (Such as RedShift, EKS) are deployed to private subnets to prevent public access. They can only be accessed via connection to a bastion host deployed in the public subnet and IP restricted.

All application logs emitted are captured by CloudWatch.

Storage:

- S3 can be used to store the images uploaded by the API or Kafka.
- Lifecycle policy can be configured in S3 to delete objects which are older than 7 days in the buckets.
- Security:
    - Data is transmitted into S3 via TLS/SSL which secures the connection and traffic, providing encryption-in-transit.
    - Server-Side-Encryption can be enabled to encrypt data within the S3 buckets using either AWS KMS or self-provided keys, providing encryption-at-rest.
    - API and Kafka should have their own IAM roles for transmitting data into S3.
    - API and Kafka should also upload to their own S3 buckets to segregate data and access. Bucket polices can also be implemented to limit access to only their respective IAM roles. Limiting potential impact of a breach to only 1 location.

Code Management:

- Software engineers can create repositories in CodeCommit and commit their codes into them.
- CodeCommit functions similarly to GitHub, allowing the engineers to version control and manage their code.
- Security:
    - Repositories can only be accessed by users with a valid IAM account which has a SSH key link to it or, CodeCommit HTTPS credentials.

Code Deployment:

- CI/CD and deployment can be performed using CodeBuild and CodePipeline.
- A buildspec.yml file which details the actions to perform (Such as docker build and pytest) after committing to a repository is create in the root of the repository and committed.
- CodeBuild will pick up the file and perform the corresponding actions whenever a commit and push are performed. CodePipeline is used to monitor and orchestrate the interaction between CodeCommit and CodeBuild.
- Disadvantage:
    - Multibranch pipelines are not supported hence a separate CodeCommit and CodeBuild will have to be created for each branch. The alternative is to look to commercial options such as GitLab which can be more expensive.
- CodeBuild can be used to push code to desired locations (I.E. .py files to S3 for usage by EMR) or build code into Docker images and push them into ECR.

- Docker scan or CodeGuru can used to detect vulnerabilities in the image.
- ECS or EKS can then be used to run the Docker containers
    - EKS:
        - AWS managed Kubernetes service, suitable for hosting large numbers of microservices as each node can run many pods (~750).
        - Disadvantages:
            - Pod scaling is manually configured.
            - More complex to manage that ECS.
    - ECS:
        - Elastic Container service by AWS, straight-forward to use, integrates with other AWS services and scaling is managed by AWS.
        - Disadvantages:
            - Vendor locked
            - Might not be suitable for large scale deployments
    - Cost:
        - Additional cost for EKS as each cluster node is also charged.

Analytics/BI:

- RedShift is the recommended OLAP/columnar database for AWS.
- RedShift Serverless is selected where AWS manages and scales the database up and down in compute resources depending on the load the database is currently under.
- For BI users and analysts, they can use Tableau, Quik or other tools to connect after establishing the SSH tunnel and port-forwarding.
- Cost:
    - Being able to dynamically scale the database will allow for reduced cost as compared with the traditional RedShift where instances are selected and ran 24/7.
- Security:
    - Security group to control which applications/services can access RedShift Serverless.
    - Deployed to private subnets to prevent public access.
    - Can only be accessed within the AWS environment or, via a SSH tunnel with port-forwarding (ssh -v -i "<ssh pem file>" <user>@<host> -L < port>:<remote host address>:<port> -N)

Assumptions:

- Data required for BI users and analysts are present in S3 or other databases and inserted into RedShift Serverless.