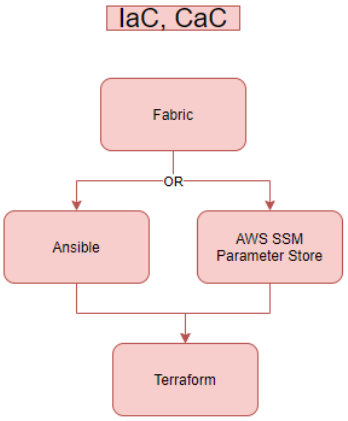
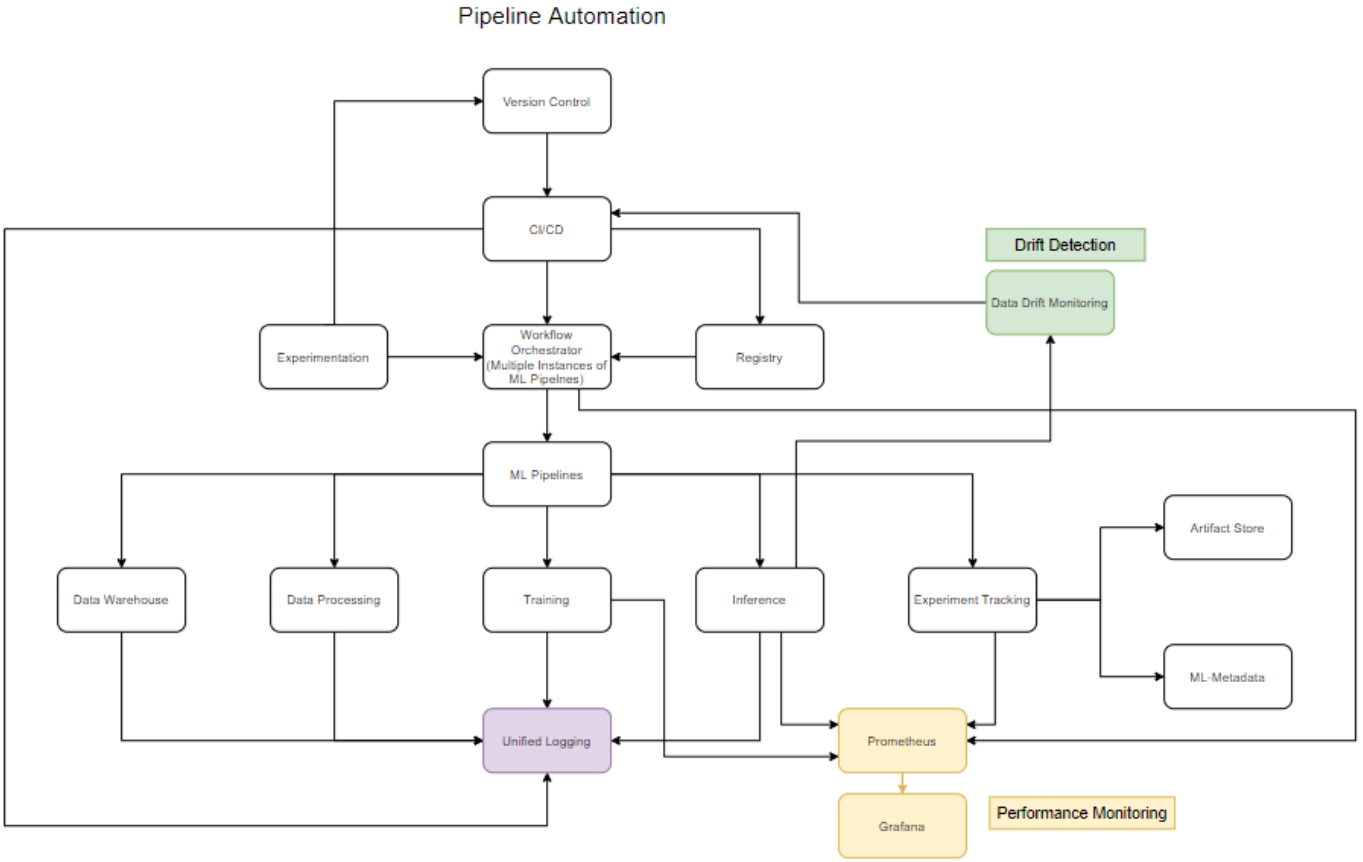
Updated Architecture





Proposed Services

1. **Version-Control**:

Primary Choices:

- GitHub
- DVC

Secondary Choices:

- AWS Code Commit
  - Least Preferred
  - Might not support for cross-cloud, cross-platform (On-Premise & Cloud)

2. **CI/CD**:

- Jenkins (Standalone + Autoscaling group)
- Jenkins-X (Jenkins on K8s)
  - Pros:
    - Can use Spot Instances
    - Can leverage different machine types combined with SPOT-Capacities
    - Good Integration with K8
    - Cross-Platforms
    - Can easily migrate to other clouds
  - Cons:
    - Not Serverless
    - Can make it serverless using Event-Grid from AWS
- Jenkins with Argo CD
- Jenkins with Argo CD and Seldon Core

- AWS Code-Pipeline & AWS Code Stars
  - Pros:
    - Serverless
    - Good Integration with AWS Sagemaker and AWS resources
  - Cons:
    - Need to check whether it has integration with K8s for leveraging SPOT-Capacities and different machine types

3. **Registry for Build**:

- Container Registry:
  - AWS ECR

- Build Artifacts Registry:
  - AWS S3
    - Supports Life-Cycle configuration policies
    - Cheap with cold-storage and archive options

  DVC

  Mlflow doesn't support S3

4. **Experimentation**:

- JupyterHub (https://access-emr.sandbox.sbd-caspian.com:9443/hub/login)
- Kubeflow jupyter environment supporting namespaces
- JupyterHub hosted on Kubernetes
- Sagemaker Studio/Notebooks

5. **ML Workflow Orchestrator**:

Primary Choices:

- Kubeflow
- Airflow
  - Pros:
    - Almost everything is possible
    - Supports cross-clouds
    - Has support for K8s and Celery Executors
  - Cons:
    - Heavy weight
    - Not Serverless
    - AWS Serverless option is a bit costly (Need to explore this)
- Metaflow
  - Pros:
    - Light Weight
    - Has good integration with Step Functions & AWS Batch
    - Serverless
  - Cons:

- Have integration only with AWS Batch

6. **ML Pipeline**:

    - Use the Kubeflow orchestrator from Step-5 as the ML Pipeline
    - Sagemaker Pipelines

7. **Data Warehouse**:

    - EMR
    - Snowflake (Later)
    - AWS Redshift

    - AWS S3 Data Lake (with Glue ETL jobs)
      AWS Lake Formation (Glue Crawler/Glue Catalog)

8. **Data Processing**:

    Primary Choices:

    - EMR
    - Snowflake

    Secondary Choices:

    - Glue ETL
    - Serverless
    - AWS Redshift
    - AWS Sagemaker Processing Jobs
        - Pros:
            - Uses Spark in the backend
            - Support for SPOT-Capacities
        - Cons:
            - Need to explore this
            - A bit costly
            - Need to check quota limits

9. **Training**:

    - Kubernetes
    - Sagemaker Computes
        - Cons:

        - Sagemaker instances are 1.5X the normal cost
    - AWS Batch (queues, compute environment) (many to many)
      Metaflow (serverless)

10. **Inference**:
    - Real-Time Predictions:
        - Sagemaker Endpoints:
            - Pros:
                - Supports Fractional GPUs
                - Elastic Computes
                - Support Chaining of Multiple Models
        - Seldon Core:
            - Need to analyse
        - KFServing from Kubeflow:
            - Need to analyse
    - Batch Predictions:
        - Sagemaker Batch Processing
        - Seldon Core

11. **Experiment Tracking**:
    - MLFlow
        - Supports Cross-Platforms
    - Sagemaker Experiments
        - No option for Cross-Platforms
        - Plots and UI are not as extensive as MLFlow

12. **ML Metadata**:
    - Postgres DB: AWS Aurora
        - Aurora is preferred because of the Serverless-Option
        - Can be used as a backend for MLFlow
        - Serverless
        - Use DB-URI for logging (private link, bastion (ui)/autoscale)

13. **Artifact Store for ML**:

- S3:
  - Supports Life-Cycle configuration policies
  - Cheap with cold-storage and archive options
  - Supports Versioning

  DVC ? To check ?

14. **Drift Detection**:
    - Kubeflow alibi detect
    - Sagemaker Model Monitoring:

15. **Unified Logging**:
    - Struct Logs (python) - json format
      - Kinesis Firehose (Serverless)
        - S3 (partitioned) Glue + Athena ES (kibana UI)
    - FluentD
      - Need to check on this
    - Jenkins + Log integration, Centralized UI
    - UI Webserver(S3 Websites) python RestAPi/App Server (API-Gateway)
    - To Figure out ?