**Document: Automating Deployment with AWS CI/CD framework**

**Introduction**

This document outlines the architecture and implementation of a CI/CD pipeline using AWS customised frameword. The solution automates the deployment process for multiple repositories and branches (develop, qa, master) using a customized buildspec.yml. This automation eliminates manual intervention and ensures consistent, repeatable deployments across different environments.

**Pipeline Architecture**

1. **Repositories and Branch Pipelines**:
   * Separate pipelines are created for each repository and branch.
   * Branch-specific configurations ensure isolated environments for development (develop), testing (qa), and production (master).
2. **CodePipeline**:
   * Orchestrates the CI/CD process, triggering builds and deployments.
   * Integrates with CodeBuild for the build and deployment phases.
3. **CodeBuild**:
   * Executes a dynamic and reusable buildspec.yml file.
   * Manages Docker image creation, storage in Amazon ECR, and deployment to EC2.
4. **Amazon ECR**:
   * Centralized Docker container registry for storing images built in CodeBuild.
5. **SNS Notifications**:
   * Sends success or failure notifications for each phase of the pipeline.

**Build Specification (buildspec.yml)**

The buildspec.yml file is designed to handle tasks for all environments dynamically. Below is a detailed explanation of each phase:

**Version**

version: 0.2

**Phases**

1. **Install**:
   * Downloads a private key from S3.
   * Configures permissions for secure SSH connections.

**Key Commands**:

aws s3 cp s3://cloudbuild-privatekey/key.pem /tmp/key.pem

chmod 400 /tmp/key.pem

1. **Pre-Build**:
   * Authenticates with Amazon ECR.
   * Deletes existing Docker images in the target repository.

**Key Commands**:

aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin <account\_id>.dkr.ecr.<region>.amazonaws.com

aws ecr batch-delete-image --repository-name "$REPO\_ENV/$REPO\_NAME" --image-ids imageDigest="$DIGEST"

1. **Build**:
   * Builds the Docker image from the project source.
   * Tags the image with a branch-specific identifier.

**Key Commands**:

docker build -t "$REPOSITORY\_URI:$IMAGE\_TAG" .

1. **Post-Build**:
   * Pushes the Docker image to Amazon ECR.
   * Deploys the image to an EC2 instance via SSH and Docker Compose.

**Key Commands**:

docker push "$REPOSITORY\_URI:$IMAGE\_TAG"

ssh -i /tmp/key.pem ubuntu@<EC2-Instance-IP> "docker-compose up -d --no-deps --force-recreate $SERVICE"

**Artifacts**

Specifies the files to be retained after the build for debugging or reporting.

artifacts:

files:

- '\*\*/\*'

**Dynamic Variables**

* **Environment Variables**:
  + $BRANCH: Indicates the active branch (e.g., develop, qa, master).
  + $REPO\_ENV: Environment-specific repository path.
  + $SNS\_TOPIC: SNS topic for notifications.
  + $SERVICE: Name of the Docker service to deploy.

**Failure Notifications**

Each phase of the build includes error-handling logic to send failure notifications via SNS. Examples:

aws sns publish --topic-arn arn:aws:sns:<region>:<account\_id>:$SNS\_TOPIC --message "Phase failed: Reason" --subject "CodeBuild Failure Notification for $BRANCH"

**Deployment Workflow**

1. **Trigger**: CodePipeline triggers CodeBuild for changes in repositories/branches.
2. **Build and Push**: CodeBuild builds the Docker image and pushes it to Amazon ECR.
3. **Deploy**:
   * CodeBuild SSHs into the target EC2 instance.
   * Pulls the latest Docker image and updates the service using Docker Compose.
4. **Notify**: Sends a notification upon success or failure.

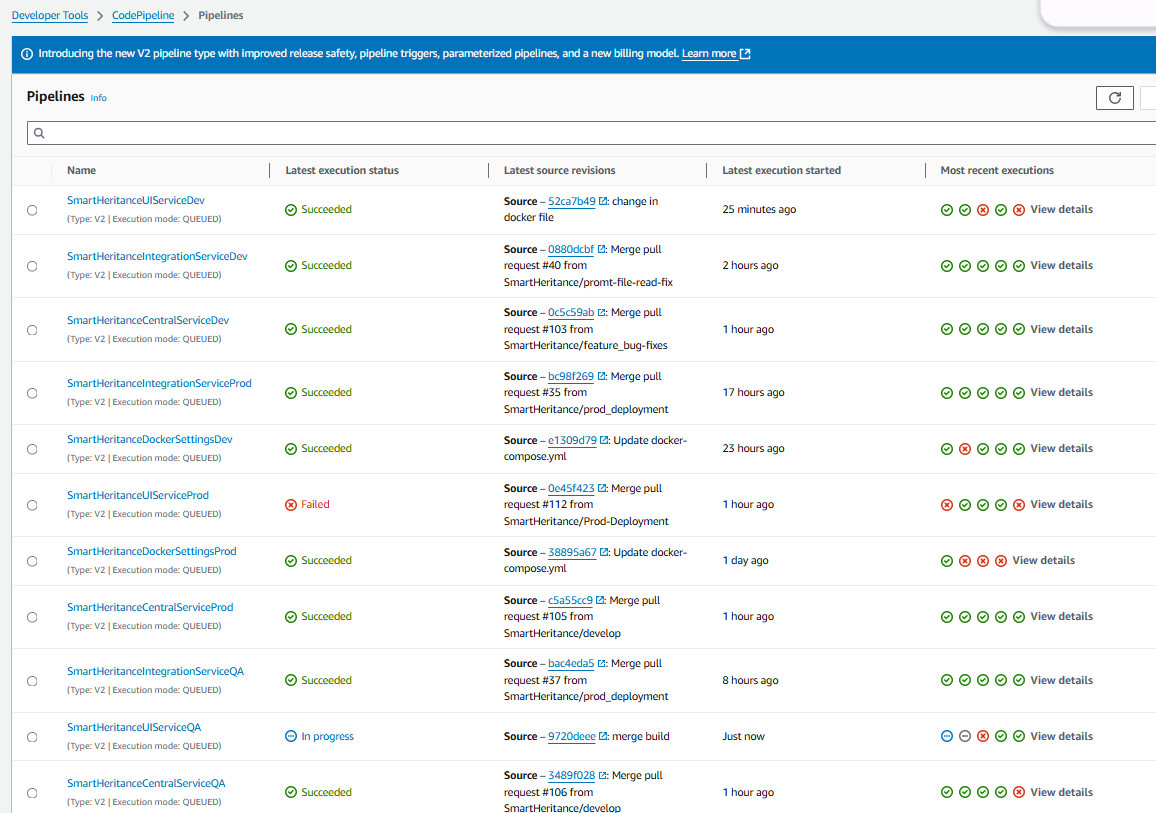
**Advantages**

* **Dynamic Buildspec**:
  + A single buildspec.yml supports multiple environments and branches.
* **Error Handling**:
  + Integrated notifications for monitoring pipeline failures.
* **Reusability**:
  + The same pipeline structure is reusable across multiple repositories.
* **Automation**:
  + End-to-end deployment with no manual intervention.

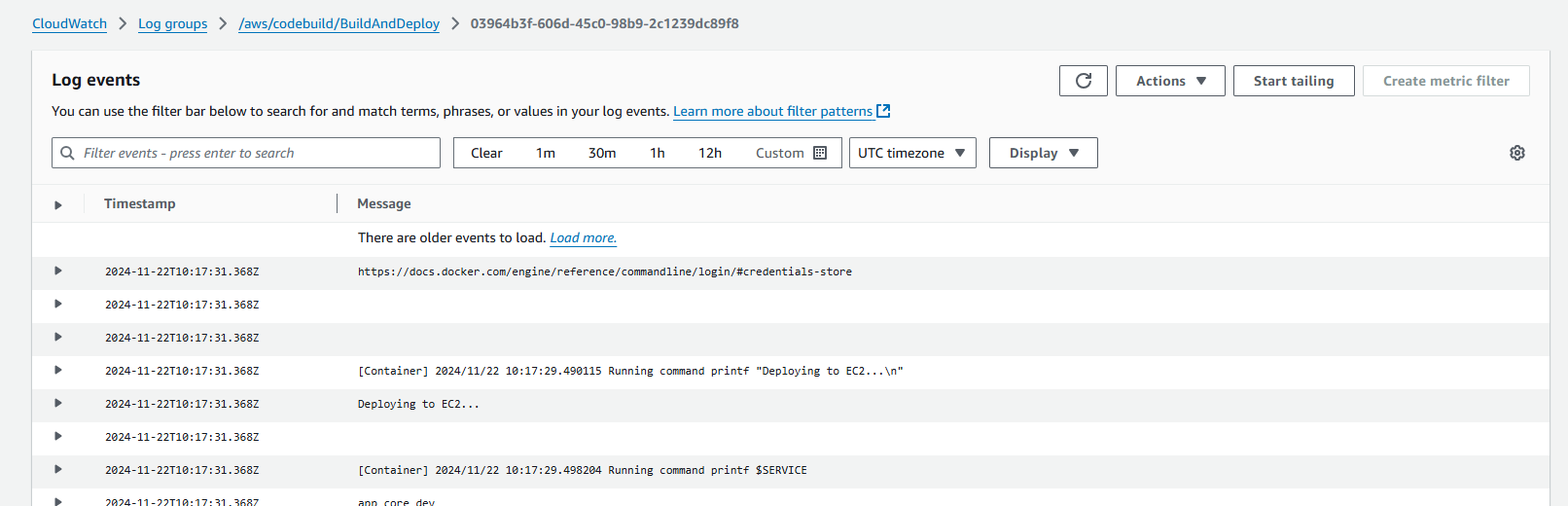
**Screenshots**

Add screenshots for:

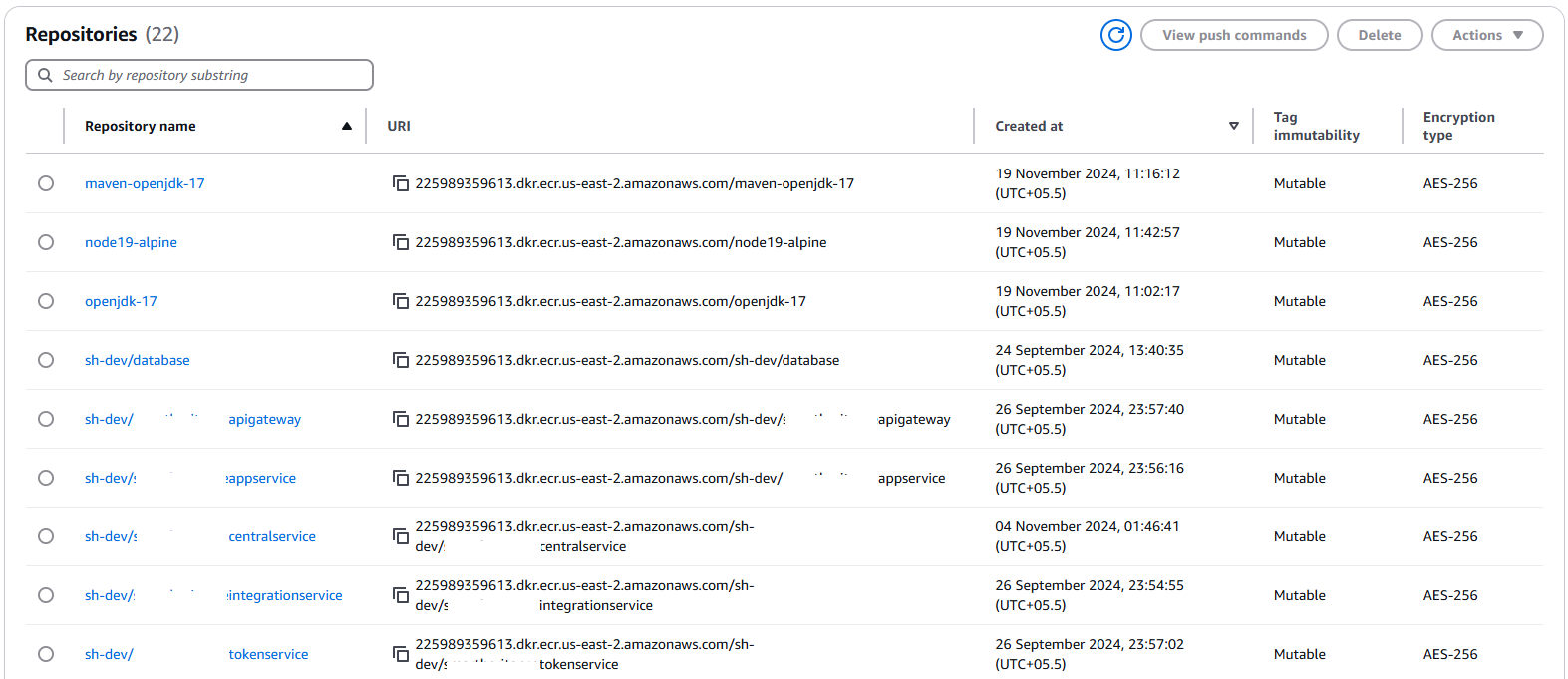
1. AWS CodePipeline overview.



1. CodeBuild phases and logs.



1. Amazon ECR repositories and Docker images.



1. SNS notifications for success and failure scenarios:

