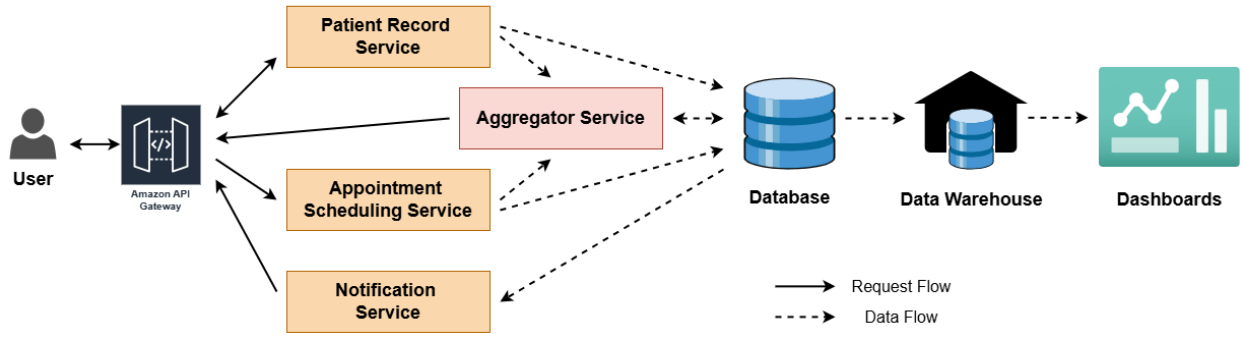
HealthSync

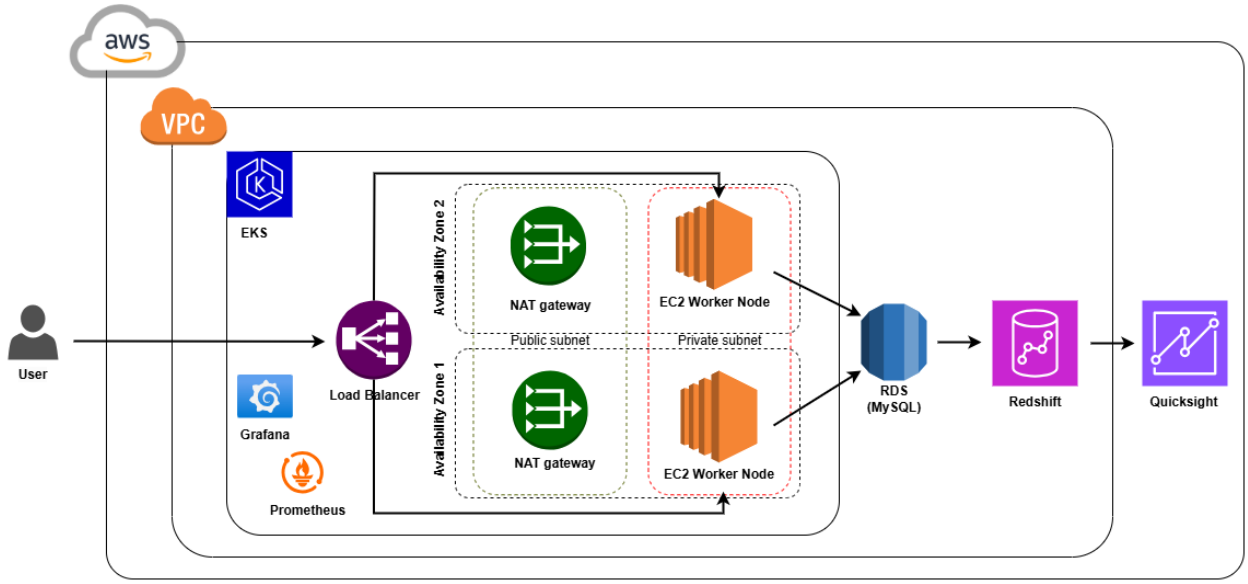
Runbook name	**Manage patient health records and streamline medical appointment tracking solution**
Runbook description	Runbook – Patient record management and Appoinment handling
Owner	@Ranali Wijendra
Version	V.0.0.1
Version date	29 Dec 2024
On this page	Architecture | Support contacts | Runs | Process

Architectures

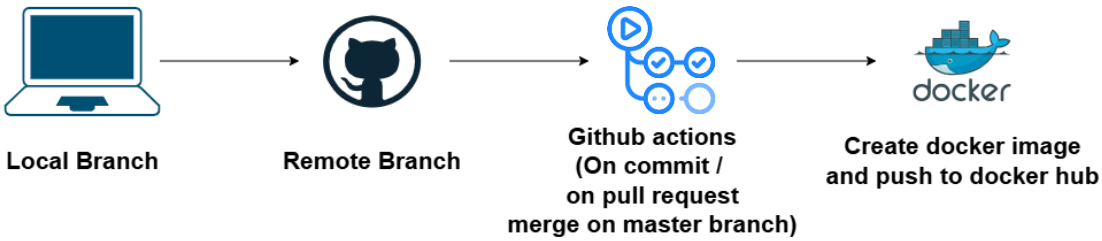
Service Communication Architecture / Solution Architecture



Deployment Architecture



Continuous Integration



Continuous Deployment

Support Contacts

Expertise Level	Team	Team lead	Contact Info
Developer	@Ranali Wijendra		ranali.wijendra@live.com
Product Owner	@Ranali Wijendra		ranali.wijendra@live.com

Runs

Name	State	Start Time	Completed Time	Duration
Database backup	SUCCESS / ERROR			
Code review				
Service health monitoring				

Process

	Step Instructions	Execution location	Run environments	Run conditions	Run instructions	Documentation
1	Configure AWS CLI	Local Machine	Local Machine CLI		aws configure	https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-configure.html

2	Setup ektcl	Local Machine	Local Machine CLI	Run as administrator	choco install –y eksctl	
3	Install Kubernetes CLI	Local Machine	Local Machine CLI	Run as administrator	choco install -y kubernetes-cli	
4	Create EKS cluster	AWS	AWS Configured CLI in local machine		eksctl create cluster --name HealthSyncCluster --region us-east-1 --nodes 2 --node-type t3.small --version 1.30	
5	Create RDS for services	AWS EKS Cluster VPC	AWS	MySQL RDS on same VPC as the EKS Cluster	After creating the RDS, create the healthsync database and create tables required by each service	
6	Update connection details for patient record service	GitHub	Local Machine		Clone the repository, change the connection string, commit and merge to main branch using a pull request	https://github.com/ranaliH/Patient-Record-Service
7	Update connection details for notification service	GitHub	Local Machine		Clone the repository, change the connection string, commit and merge to main branch using a pull request	https://github.com/ranaliH/Notification-Service
8	Update connection details for appointment scheduling service	GitHub	Local Machine		Clone the repository, change the connection string, commit and merge to main branch	https://github.com/ranaliH/Appointment-Scheduling-Service

					using a pull request	
9	Update connection details for aggregation service	GitHub	Local Machine		Clone the repository, change the connection string, commit and merge to main branch using a pull request	https://github.com/ranaliH/Aggregator-Service
10	Set Environmental variables for secrets	AWS	AWS Configured CLI in local machine		kubectl create secret generic healthsync-db-credentials --from-literal=DB_HOST=xxx --from-literal=DB_USER=xxx --from-literal=DB_PASSWORD=xxx --from-literal=DB_NAME=xxx	
11	Set Environmental variables for KUBECONFIG	GitHub	Local Machine		Set the DB_HOST, DB_NAME, DB_USER and DB_PASSWORD in the secrets	
12	Deploy service of patient record service	AWS	AWS Configured CLI in local machine		kubectl apply -f patient-record-service-blue/deployment.yaml kubectl apply -f patient-record-service-green/deployment.yaml	https://github.com/ranaliH/RunBook/tree/main/deployment/patient_record_service

13	Deploy service of notification service	AWS	AWS Configured CLI in local machine		kubectl apply -f notification-service-blue/deployment.yaml kubectl apply -f notification-service-green/deployment.yaml	https://github.com/ranaliH/RunBook/tree/main/deployment/notification_service
14	Deploy service of appointment scheduling service	AWS	AWS Configured CLI in local machine		kubectl apply -f appointment-scheduling-service-blue/deployment.yaml kubectl apply -f appointment-scheduling-service-green/deployment.yaml	https://github.com/ranaliH/RunBook/tree/main/deployment/appointment_scheduling_service
15	Deploy service of aggregator service	AWS	AWS Configured CLI in local machine		kubectl apply -f aggregator-service-blue/deployment.yaml kubectl apply -f aggregator-service-green/deployment.yaml	https://github.com/ranaliH/RunBook/tree/main/deployment/aggregator_service
16	To monitor get the details from Prometheus	AWS	AWS Configured CLI in local machine		kubectl port-forward -n monitoring svc/prometheus-kube-prometheus-prometheus 9090 http://localhost:9090	

17	Monitor Health of EKS cluster	AWS	AWS Configured CLI in local machine		kubectl port-forward -n monitoring svc/prometheus-grafana 3000:80 http://localhost:3000	
18	Access analytics data using Quicksite	AWS	AWS Quicksight		https://us-east-1.quicksight.aws.amazon.com/sn/dashboards/bfcd1a25-3131-46b2-a9cb-156c19601924	