**04. Pre-requisites**

--- Reference - <https://github.com/stacksimplify/aws-eks-kubernetes-masterclass/tree/master/13-Microservices-Distributed-Tracing-using-AWS-XRay-on-EKS>

**Introduction**

--- Introduction to AWS X-Ray & k8s DaemonSets

* Understand about AWS X-Ray Services
* Understand Kubernetes DaemonSets
* Understand the AWS X-Ray and Microservices network design on EKS Cluster
* Understand about Service Map, Traces and Segments in AWS X-Ray

**Usecase Description**

--- User Management getNotificationAppInfo will call Notification service notification-xray which will evetually send traces to AWS X-Ray service

--- We are going to depict one Microservice calling other Microservice

**List of Docker Images used in this section**

**Application Name** **Docker Image Name**

**User Management Microservice** stacksimplify/kube-usermanagement-microservice:3.0.0-AWS-XRay-MySQLDB

**Notifications Microservice V1** stacksimplify/kube-notifications-microservice:3.0.0-AWS-XRay

**Pre-requisite: AWS RDS Database, ALB Ingress Controller & External DNS**

--- AWS RDS Database

* We have created AWS RDS Database as part of section 06-EKS-Storage-with-RDS-Database
* We even created a externalName service: 01-MySQL-externalName-Service.yml in our Kubernetes manifests to point to that RDS Database

**ALB Ingress Controller & External DNS**

* We are going to deploy a application which will also have a ALB Ingress Service and also will register its DNS name in Route53 using External DNS
* Which means we should have both related pods running in our EKS cluster.
* We have installed ALB Ingress Controller as part of section 08-01-ALB-Ingress-Install
* We have installed External DNS as part of section 08-06-01-Deploy-ExternalDNS-on-EKS

**# Verify alb-ingress-controller pod running in namespace kube-system**

--- **kubectl get pods -n kube-system**



**# Verify external-dns pod running in default namespace**

--- **kubectl get pods**

