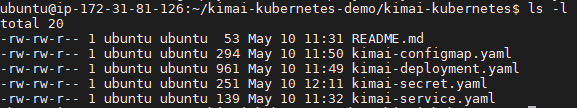
**KIMAI App Deployment on Kubernetes**

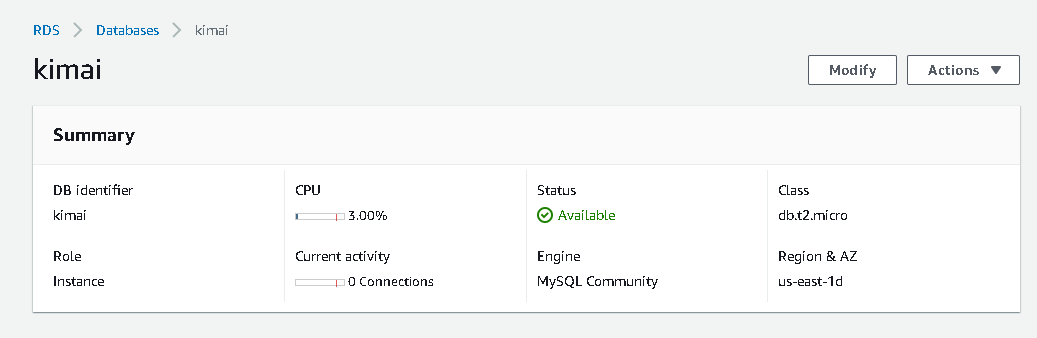
* Kubernetes Cluster setup info:
  + kubeadm 1.21
  + 1 Control Plane Node (Master)
  + 1 Worker Node
  + OS: Ubuntu 18.04
  + AWS EC2 instance type: t3.medium
  + Kimai App Containers will run on Kubernetes and Database is running on AWS RDS.



* Below are Kubernetes manifest files to deploy “KIMAI App” on kubernetes cluster.



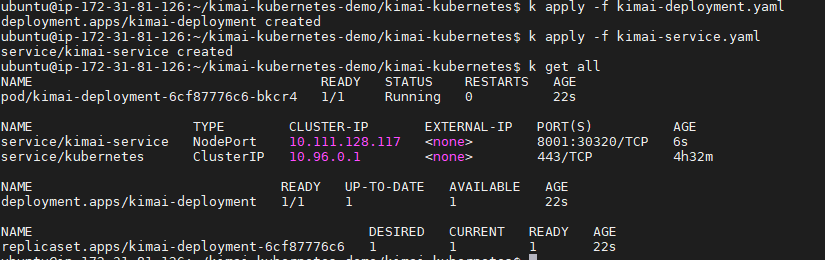
* **Database**: AWS RDS Mysql Server and create database having name “kimai”.



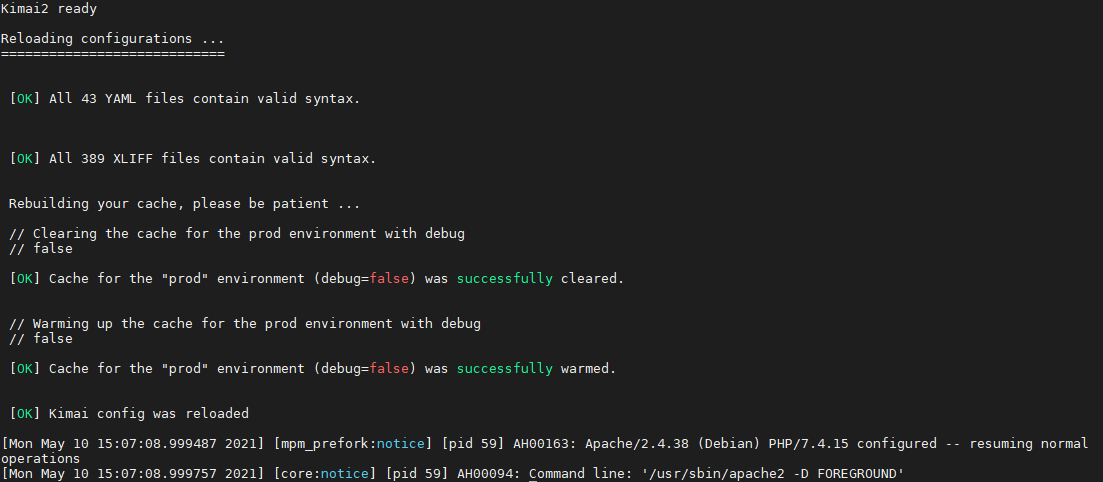
* Kimai App Containers requires some input parameters as environment variables i.e. ADMINMAIL, ADMINPASS, DATABASE\_URL and TRUSTED\_HOSTS. For the same, Please create configmap and secret object to store these environment variables values. Which can be used in deployments.yaml file for the KIMAI APP Deployment.



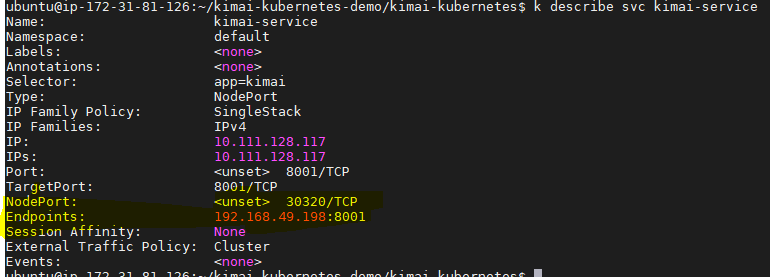
* To deploy the application on kubernetes cluster fire below command which will create deployment and expose these deployment using service.



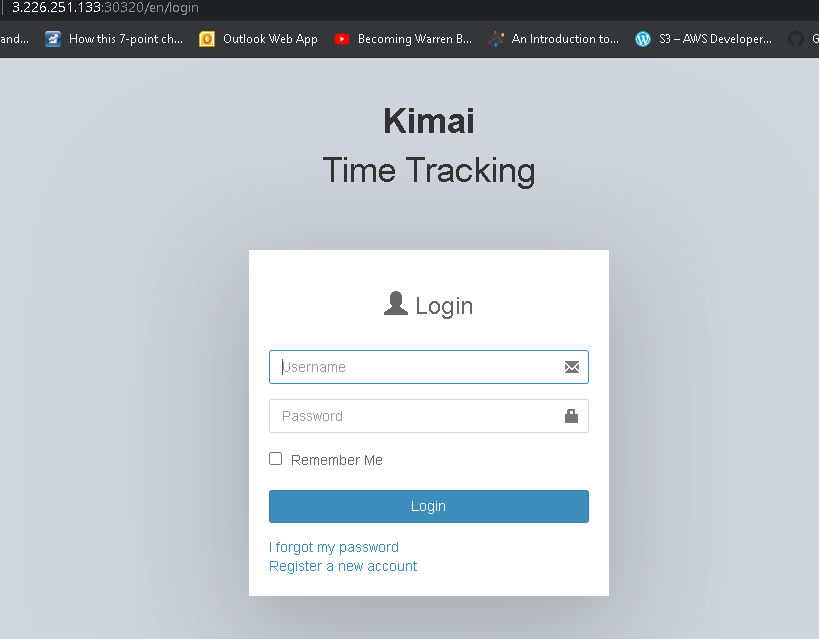
* Below command is used to see logs inside kimai-app pods:
  + Kubectl logs deployment/kimai-deployment



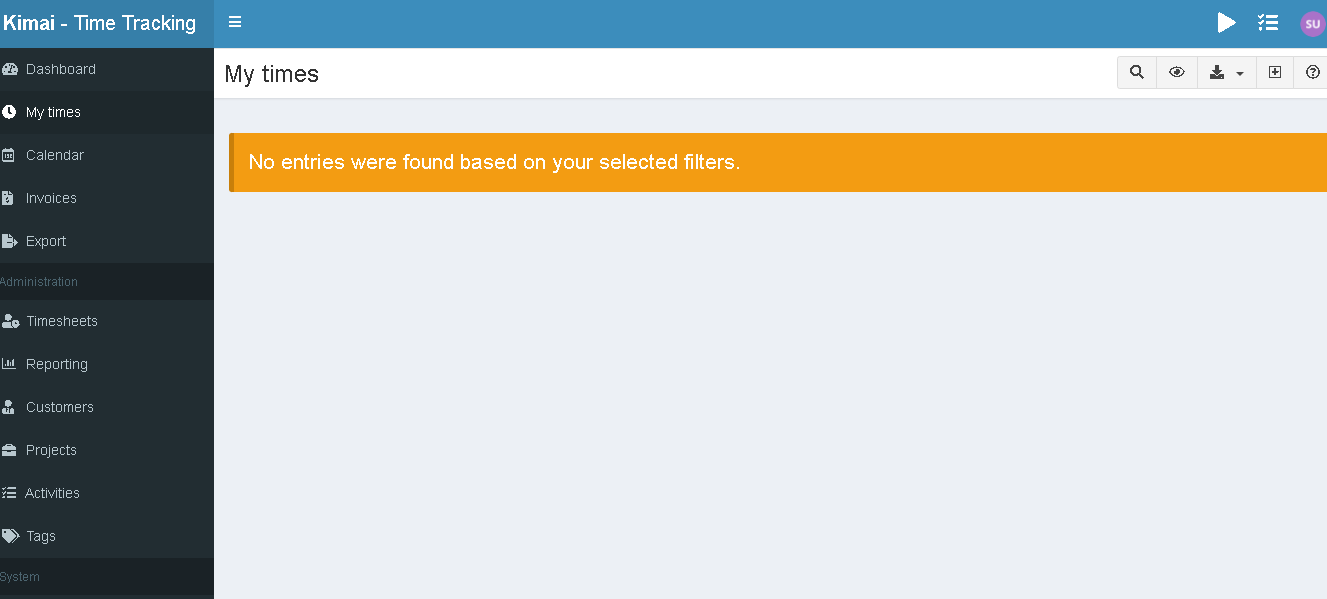
* Kimai-app pods are expose using nodeport service so below command will give us a nodeport.



* To access the application visit ***NodeIP:NodePort using browser*** and check whether your application is running or not.



* Login to application using admin username and password which was defined in kimai-secrets.yaml while deployment:



* Areas of optimization:
  + We can use AWS ECS service to host this application which is cost-effective solution since AWS Provides ECS Master free of cost.
  + We can use AWS EKS for the kubernetes cluster deployment.
  + We can use Terraform to provision the infrastructure.
  + If we are using AWS EKS, we can use AWS ALB as ingress-controller and application pods will be exposed using ingress and ingress-controller (AWS ALB).
  + Since “KIMAI-APP” is PHP application, if it requires continuous read/write operation on file system, we can use AWS EFS and mount Application file system on AWS EFS using Persistent-Volumes and Persistent-Volume-claims.
  + We can also use helm-charts for the deployment