Simple Spring Boot API - Pollinate Timestamp

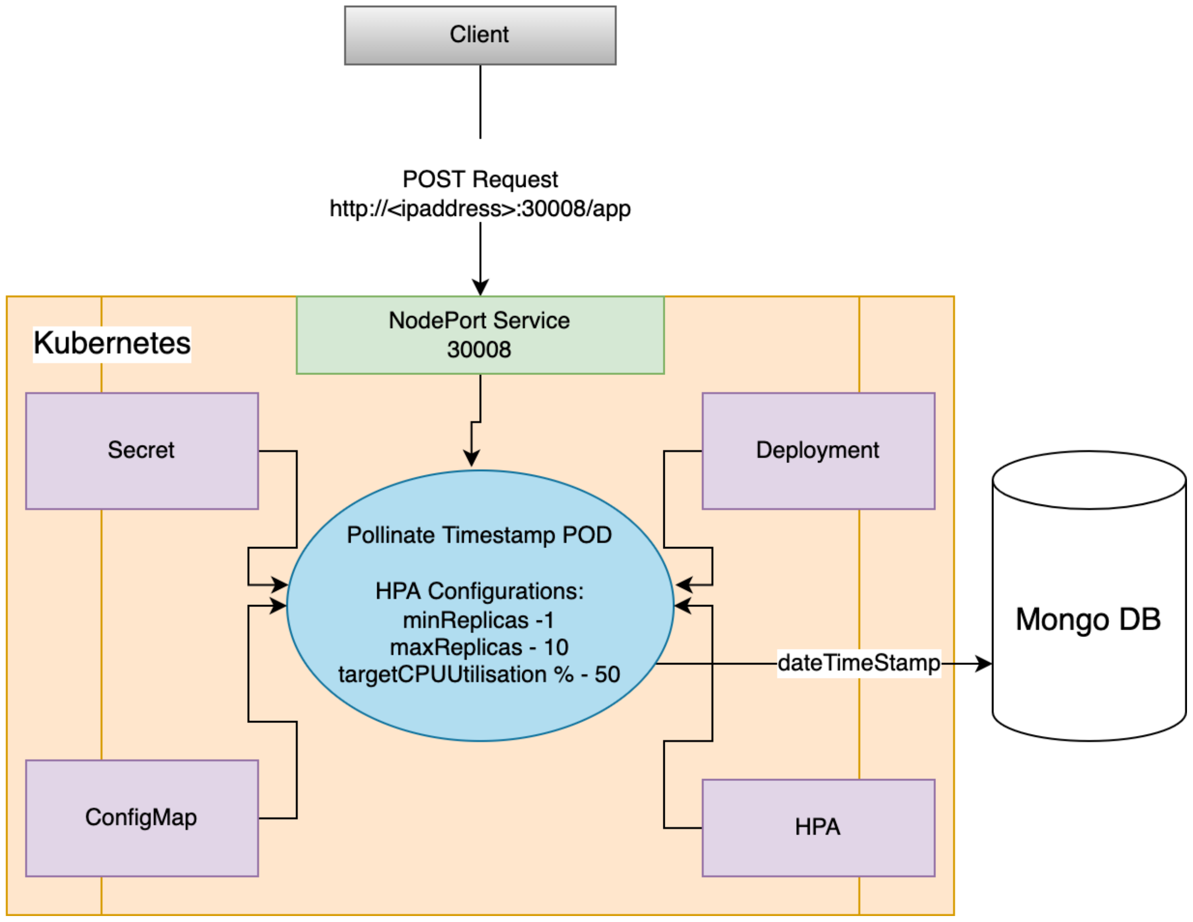
Objective

1. Create a proof of concept of the following:
   Create an application with simple API that will be used to submit a curl command (curl -X POST http:// /app) that will insert the date-time stamp into your database.
2. The proof of concept should consist of a single web-application and a single database.
3. Create a GitHub project for this assignment.
4. Automation the provisioning your database node and web node with the API you created.
5. A readme file must be populated with your git project detailing your solution and how to run it.
6. Provide details of your high-level design that would meet the requirements outlined above.

Additional Requirements

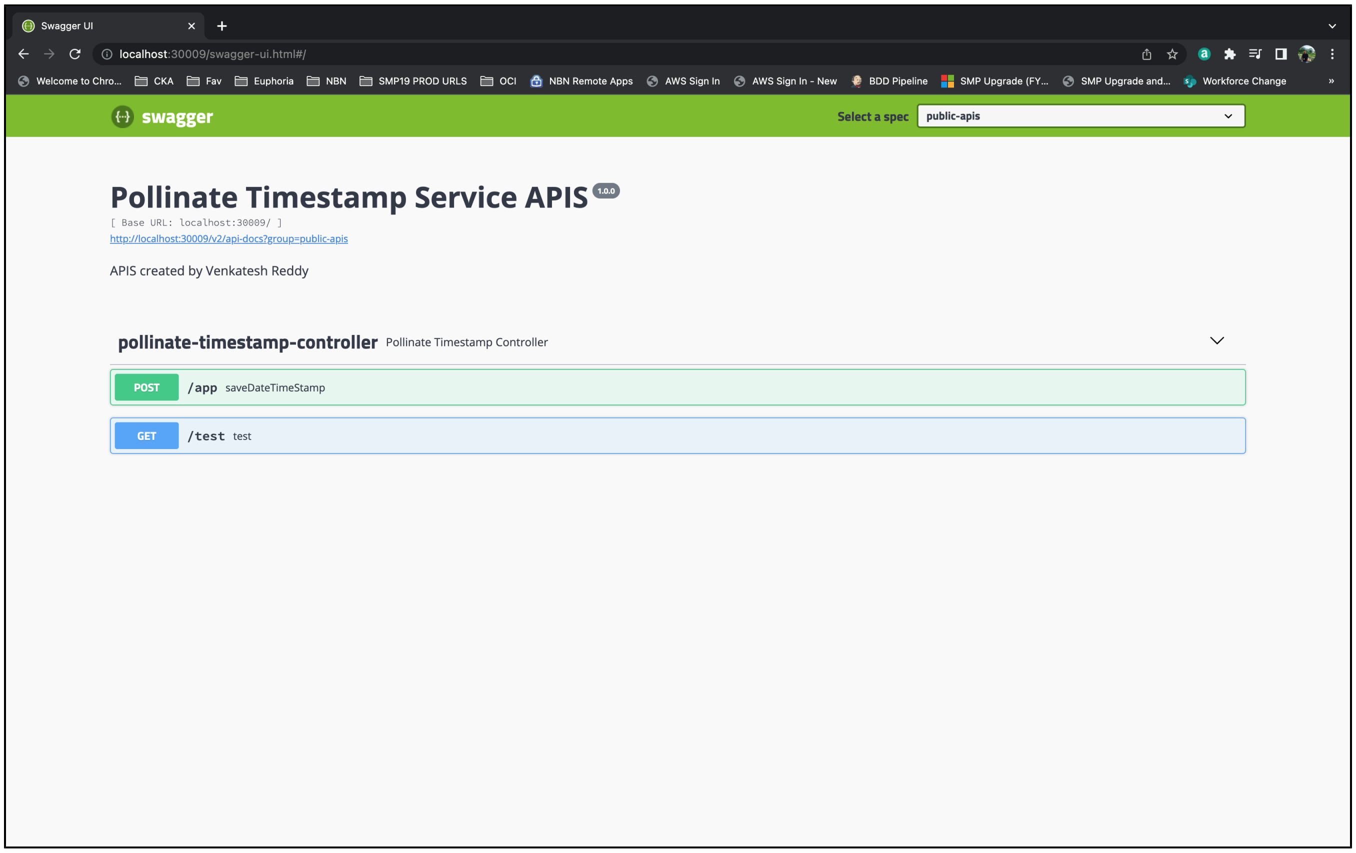
1. The service will accept a POST request that will persist the timestamp of that event into a database. For all aspects of your solution, you should consider multi-DC, zones and redundancy of components.
2. Please note that the design must be elastic based on demand.
3. For any areas, which you have not considered to automate, please state these clearly and if possible, provide details of the solution you would have implemented including your experience/depth of knowledge of that area.
4. Be ready to explain your reasons regarding your choice of solution.
5. The choice of technology is up to you. For example, if you choose to use a NoSQL database or a Relational database is your choice.
6. The provisioning tool suggested below is Ansible, however, you may use any tool of your choice for your automation.

High Level Design



1. Service
    a. Create a Node Port Service to expose the API to client on NodePort 30008.
2. Deployment
    a. Deployment to create PODs for Pollinate Timestamp Spring Boot API
3. HPA
    a. Define HPA to provide elasticity/scalability based on demand
    b. minReplicas-1, maxReplicas-10, targetCPUUtilisationPercentage-50
4. Secret & ConfigMap
    a. Store Mongo DB username, passwords in secrets and host details in configMap
5. Mongo DB
    a. None of the provided requirements vote for SQL database, as a result due to below reasons we can use NoSQL database.
    b. NoSQL databases can scale horizontally very efficiently across systems and locations, making it possible to accommodate large stores of distributed data, while supporting increased levels of traffic.
    c. A NoSQL database uses a dynamic schema that requires no predefined data structure, resulting in a a high degree of flexibility, such as being able to add documents with different fields to the same database.
    d. Provision Mongo DB as cloud service with Multi Availability Zone and Backup provision for HA.
6. Pollinate Timestamp POD
    a. Create a simple Spring Boot application in Java and provide @POST method for client to invoke the API.

API Documentation



Pollinate Timestamp Service APIS 1.0.0

[Base URL: localhost:30009/]

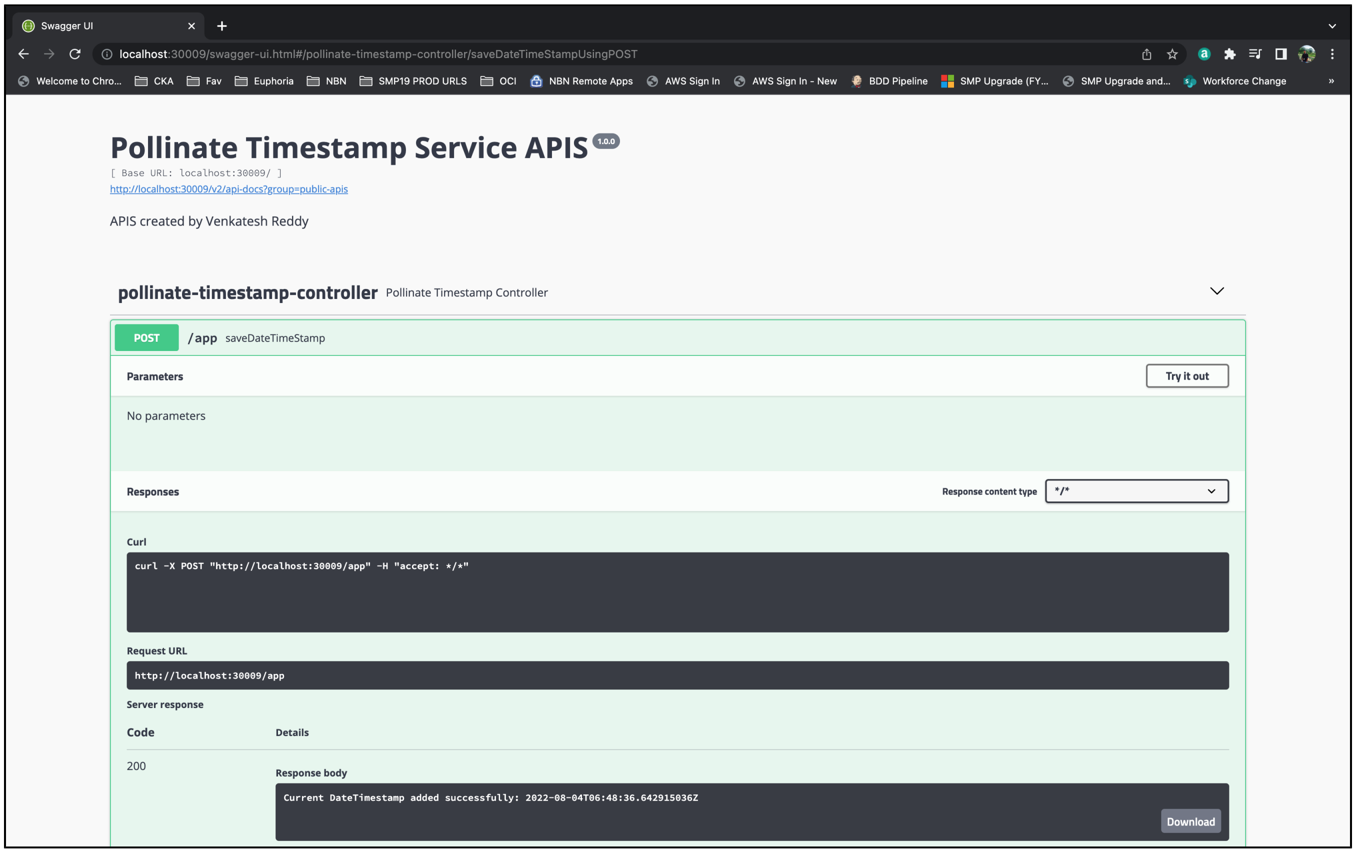
http://localhost:30009/v2/api-docs?group=public-apis

APIS created by Venkatesh Reddy

pollinate-timestamp-controller Pollinate Timestamp Controller

POST /app saveDateTimeStamp

GET /test test


Swagger UI

localhost:30009/swagger-ui.html#/pollinate-timestamp-controller/saveDateTimeStampUsingPOST

Welcome to Chro... CKA Fav Euphoria NBN SMP19 PROD URLS OCI NBN Remote Apps AWS Sign In AWS Sign In - New BDD Pipeline SMP Upgrade (FY... SMP Upgrade and... Workforce Change

Pollinate Timestamp Service APIS 1.0.0

[Base URL: localhost:30009/]

http://localhost:30009/v2/api-docs?group=public-apis

APIS created by Venkatesh Reddy

pollinate-timestamp-controller Pollinate Timestamp Controller

POST /app saveDateTimeStamp

Parameters Try it out

No parameters

Responses Response content type `*/*`

Curl

```
curl -X POST "http://localhost:30009/app" -H "accept: */*"
```

Request URL

```
http://localhost:30009/app
```

Server response

Code	Details
200	**Response body** `Current DateTimestamp added successfully: 2022-08-04T06:48:36.642915036Z`

Download

API Documentation

```
{
  "swagger": "2.0",
  "info": {
    "description": "APIS created by Venkatesh Reddy",
    "version": "1.0.0",
    "title": "Pollinate Timestamp Service APIS"
  },
  "host": "localhost:30008",
  "basePath": "/",
  "tags": [
    {
      "name": "pollinate-timestamp-controller",
      "description": "Pollinate Timestamp Controller"
    }
  ],
  "paths": {
    "/app": {
      "post": {
        "tags": [
          "pollinate-timestamp-controller"
        ],
        "summary": "saveDateTimeStamp",
        "operationId": "saveDateTimeStampUsingPOST",
        "consumes": [
          "application/json"
        ],
        "produces": [
          "*/*"
        ],
        "responses": {
          "200": {
            "description": "OK",
            "schema": {
              "type": "string"
            }
          },
          "201": {
            "description": "Created"
          },
          "401": {
            "description": "Unauthorized"
          },
          "403": {
            "description": "Forbidden"
          },
          "404": {
            "description": "Not Found"
          }
        },
        "deprecated": false
      }
    },
    "/test": {
      "get": {
        "tags": [
          "pollinate-timestamp-controller"
        ],
        "summary": "test",
        "operationId": "testUsingGET",
        "produces": [
          "*/*"
        ],
        "responses": {
          "200": {
            "description": "OK",
            "schema": {
              "type": "string"
            }
          },
          "401": {
            "description": "Unauthorized"
          },
          "403": {
            "description": "Forbidden"
          },
          "404": {
            "description": "Not Found"
          }
        },
        "deprecated": false
      }
    }
  }
}
```

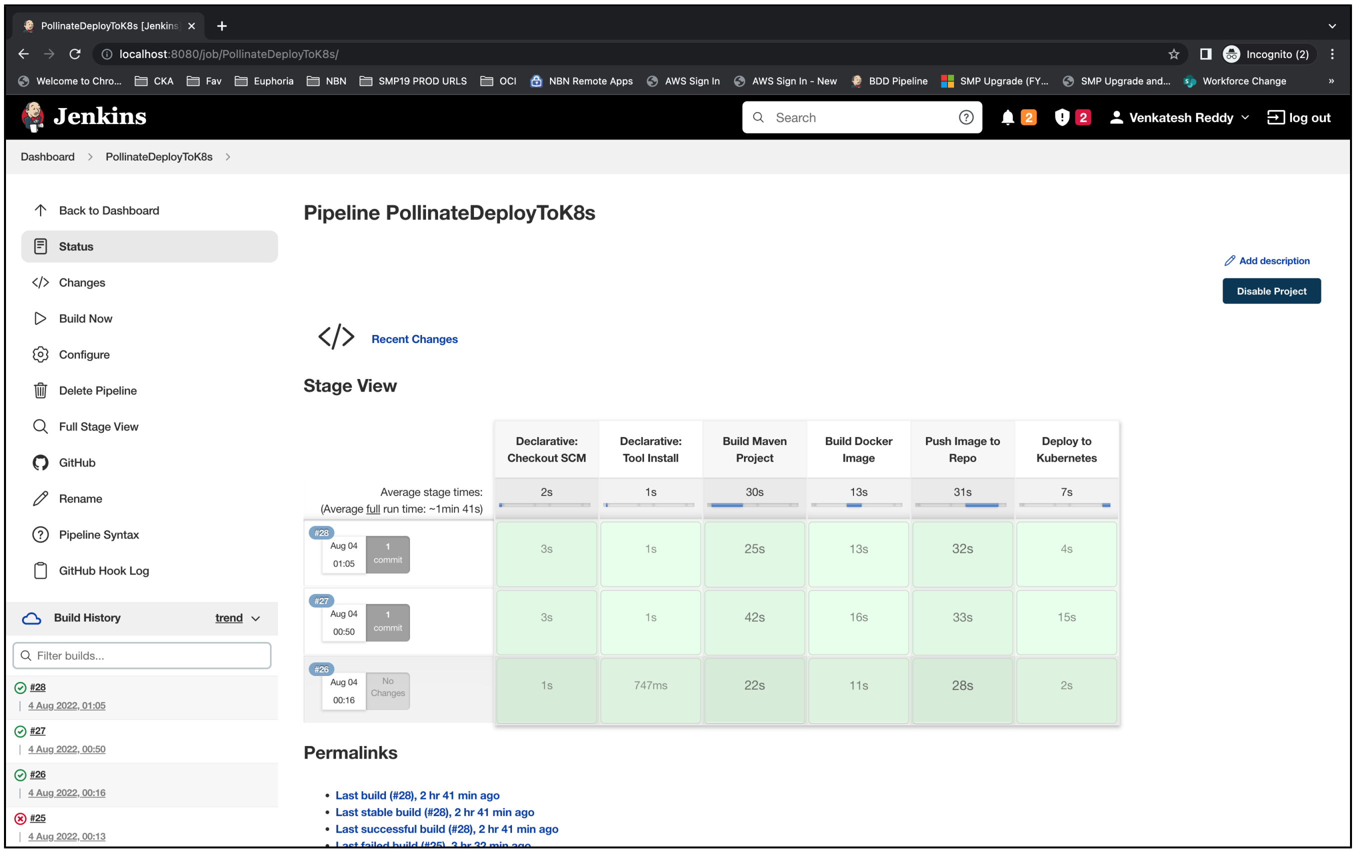
CI/CD

Automate the build, docker image, repository and deploy steps for the API.

Jenkinsfile can be found at https://github.com/itsvenkyhere/Pollinate-Service-Management-Tech-Assignment/blob/main/Section-1/Jenkinsfile

Jenkinsfile

```
pipeline{
    agent any
    tools{
        maven 'maven_3_8_6'
    }
    stages{
        stage('Build Maven Project'){
            steps{
                checkout([$class: 'GitSCM', branches: [[name: '*/main']], extensions: [], userRemoteConfigs:
[[url: 'https://github.com/itsvenkyhere/PollinateTimestampAssignment']]])
                sh 'mvn clean install -DskipTests'
            }
        }
        stage('Build Docker Image'){
            steps {
                script {
                    sh 'docker build -t itsvenkyhere/pollinate .'
                }
            }
        }
        stage('Push Image to Repo'){
            steps{
                script{
                    withCredentials([string(credentialsId: 'DockerHubPassword', variable:
'DockerHubPassword')]) {
                        sh 'docker login -u itsvenkyhere -p ${DockerHubPassword}'
                    }
                    sh 'docker push itsvenkyhere/pollinate'
                }
            }
        }
        stage('Deploy to Kubernetes'){
            steps{
                script{
                    kubernetesDeploy (configs: 'mongoDB-deployment.yaml',kubeconfigId: 'k8sconfigpwd')
                    kubernetesDeploy (configs: 'application-deployment.yaml',kubeconfigId: 'k8sconfigpwd')
                }
            }
        }
    }
}
```


PollinateDeployToK8s [Jenkins
localhost:8080/job/PollinateDeployToK8s/
Incognito (2)
Welcome to Chro... CKA Fav Euphoria NBN SMP19 PROD URLS OCI NBN Remote Apps AWS Sign In AWS Sign In - New BDD Pipeline SMP Upgrade (FY... SMP Upgrade and... Workforce Change
Jenkins
Search
Venkatesh Reddy
log out
Dashboard > PollinateDeployToK8s >
Back to Dashboard
Status
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
GitHub
Rename
Pipeline Syntax
GitHub Hook Log
Add description
Disable Project
Recent Changes

Pipeline PollinateDeployToK8s

Stage View

	Declarative: Checkout SCM	Declarative: Tool Install	Build Maven Project	Build Docker Image	Push Image to Repo	Deploy to Kubernetes
Average stage times: (Average full run time: ~1min 41s)	2s	1s	30s	13s	31s	7s
#28 Aug 04 01:05 1 commit	3s	1s	25s	13s	32s	4s
#27 Aug 04 00:50 1 commit	3s	1s	42s	16s	33s	15s
#26 Aug 04 00:16 No Changes	1s	747ms	22s	11s	28s	2s

Build History trend

Filter builds...

- #28 4 Aug 2022, 01:05
- #27 4 Aug 2022, 00:50
- #26 4 Aug 2022, 00:16
- #25 4 Aug 2022, 00:13

Permalinks

- Last build (#28), 2 hr 41 min ago
- Last stable build (#28), 2 hr 41 min ago
- Last successful build (#28), 2 hr 41 min ago
- Last failed build (#25), 3 hr 32 min ago

GitHub Code Repository

You can find the code of API at https://github.com/itsvenkyhere/Pollinate-Service-Management-Tech-Assignment/tree/main/Section-1

Tools Required

- IntelliJ
- JDK Latest Version
- Springboot framework
- Github
- Docker Hub (Repository)
- Docker Desktop (Docker & Kubernetes Cluster)
- Jenkins