Infrastructure Engineer Assignment

Objective:

Develop an API or CLI script to automate operations on a bare Kubernetes cluster. The script should connect to the k8 cluster, install necessary tooling including KEDA (Kubernetes Event-Driven Autoscaling), create deployments with event-driven scaling, and provide health status for a given deployment ID. This script should be modular in nature meaning functions should be generalized and work on different deployment configurations.

Specifications:

1. **Connect to the Kubernetes Cluster:**
   - Connect a `kubectl` client to the provided cluster.
   - Install necessary tools within the cluster, including:
     - `Helm` for package management.
     - `KEDA` for event-driven autoscaling.
   - Verify the installation of tools and provide a summary of the cluster setup.
2. **Install KEDA:**
   - Use `Helm` to install KEDA on the Kubernetes cluster.
   - Verify the installation and ensure that the KEDA operator is running.
3. **Create Deployment:**
   - Utilize the provided details to create a deployment:
     - Public image and tag from DockerHub.
     - CPU and RAM request and limit specifications.
     - Ports to expose.
     - Autoscaling targets for CPU and RAM.
     - Event source configuration for any metrics of KEDA (e.g., Kafka, RabbitMQ).
   - Create namespaced resources for:
     - **Deployment:** Define the necessary resources and environment variables.
     - **Service:** Expose the deployment to the outside world.
     - **Horizontal Pod Autoscaler (HPA):** Use KEDA to define autoscaling based on any event metrics. You are free to choose any metric.
   - Return deployment details, including endpoints and scaling configuration, to the user upon successful creation.
4. **Health Status Retrieval:**
   - Given a deployment ID, retrieve its health status by:
     - Checking the deployment and pod status.
     - Returning relevant metrics such as CPU and memory usage.
     - Reporting any issues or failures in the deployment.
5. **Documentation:**
   - Provide clear documentation on how to use the script, including prerequisites and usage examples.

6. **Additional Features [Optional] :**
    - **Integrate a Continuous Deployment Pipeline:** Set up a CD pipeline to automate the deployment of the API/CLI script changes.

Guidelines:

- Ensure the script is modular, scalable, and well-documented.
- Implement error handling and provide clear error messages for user guidance.
- Follow best practices for Kubernetes configuration and resource management.
- Consider security measures and implement appropriate access controls.
- Validate user inputs to prevent potential issues during deployment creation.
- Test thoroughly to ensure script reliability and functionality across various scenarios.
- Adhere to coding standards and conventions for clarity and maintainability.
- Provide comprehensive instructions for script usage, including prerequisites and dependencies.

Deliverables :

- Submit the API/CLI script along with any configuration files and documentation.
- Provide a brief overview of the design choices made during implementation.
- Include any relevant screenshots or logs that demonstrate the functionality of the script.