

# DevOps technical assessment – “Mini API Redis Cache”

## General Objective

You will need to deploy a simple service in Kubernetes that exposes an HTTP API to interact with Redis(save and list keys), using good DevOps practices, Infrastructure as Code(IaC), and basic CI/CD pipeline.

## Functional Description

The service must expose two HTTP endpoints:

- POST /store
  - Accepts a JSON payload like:

JSON

```
{
  "key": "somekey",
  "value": "somevalue"
}
```

- Stores the value in Redis using the given key .
- GET /keys
  - Returns a list of all currently stored Redis keys.

You may develop this service using any language you prefer, as long as it meets the functional requirements.

## Infrastructure

- Deploy the service on a Kubernetes cluster (can be local using Kind, Minikube, or in the cloud using EKS, AKS, or GKE).

## Infraestructura as Code (IaC)

- Provision the cluster using Terraform, Pulumi, AWS CDK, or other Infrastructure as Code (IaC) tools.
- Provide Kubernetes manifest (YAML) o Helm Charts.
- At a minimum, include:
  - Deployment for the API.
  - Deployment for the Redis.
  - ConfigMaps or Secrets if applicable
  - Dedicated Namespace.

## Pipeline CI/CD

Implement a basic CI/CD pipeline that includes:

- Docker image build.
- Push to a container registry (Docker Hub, GHCR, etc.).
- Automatic deployment to the Kubernetes cluster.

You may use any tool of your choice

## Submission Requirements

Please share a public repository (GitHub, GitLab, etc.) containing:

- API source code.
- Dockerfile.
- Kubernetes manifests or Helm Charts, along with Terraform/Pulumi/etc. files.
- CI/CD pipeline configuration.
- A README file with:
  - Deployment instructions.
  - Explanation of the CI/CD process.
  - How to test the endpoints.
  - How to view logs.