Kubernetes AI Agent - Administrator documentation

Kubernetes AI Agent - Administrator Documentation

Introduction

The Kubernetes AI Agent is a tool that helps you monitor and manage your Kubernetes cluster. It uses AI reasoning to analyze the state of your cluster and provides suggestions on how to resolve issues.

This guide provides information on how to set up the Kubernetes AI Agent to monitor your Kubernetes cluster.

Prerequisites

- A Kubernetes cluster deployed on Google Kubernetes Engine.
- A Kubeconfig file to authenticate with the Kubernetes cluster.
- An application running on the cluster that exposes Prometheus metrics.
- A Prometheus server with a public IP address to scrape the metrics.
- Activated Google Kubernetes Engine API and Google Logging API
- A Google Service Account with Kubernetes Engine Service Agent, Logs Viewer and Private Logs Viewer roles.
- (Optionally) Helm installed on your machine to deploy Prometheus or any other charts.
- (Testing) For testing purpose you can use Chaos Mesh to simulate issues on the cluster.

Connect Kubectl to the Google Kubernetes Engine

To connect `kubectl` to the Google Kubernetes Engine, run the following command:

```
gcloud container clusters get-credentials CLUSTER_NAME --zone=ZONE --project=PROJECT_ID
```

Replace `CLUSTER_NAME`, `ZONE`, and `PROJECT_ID` with your cluster details. You may need to authenticate with your Google account if it is not already done here.

Deploy Prometheus

To deploy Prometheus on the Kubernetes cluster, you can use the provided `values.yaml` file. Modify it to add your application and to suits your need. Run the following command to deploy it with Helm:

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
helm repo update
helm install prometheus prometheus-community/kube-prometheus-stack -n apm -f values.yaml
```

To access Prometheus, you can use port forwarding and access with your browser at `http://localhost:9090`.

```
kubectl port-forward -n apm prometheus-prometheus-kube-prometheus-prometheus-0 9090
```

Or you can modify the type of the service to `LoadBalancer` and access it with the public IP address.

```
kubectl patch svc prometheus-operated -n apm --type='json' -p='[{"op": "replace", "path": "/
```

To get the public IP address, run the following command:

```
kubectl get svc prometheus-operated -n apm
```

To uninstall Prometheus, run the following command:

```
helm uninstall prometheus -n apm
```

Deploy Chaos Mesh

To deploy Chaos Mesh on the Kubernetes cluster, you can use the provided Helm chart. Run the following commands to deploy it with Helm:

```
helm repo add chaos-mesh https://charts.chaos-mesh.org
helm repo update
kubectl create ns chaos-mesh
helm install chaos-mesh chaos-mesh/chaos-mesh -n=chaos-mesh --set chaosDaemon.runtime=contai
```

To access the Chaos Dashboard, you can use port forwarding and access with your browser at `http://localhost:2333`.

```
kubectl port-forward -n chaos-mesh svc/chaos-dashboard 2333:2333
```

You will see instructions to create a token to access the dashboard.

Installation

Step 1: Clone the repository

Clone the repository to your local machine.

```
git clone git@github.com:AlenBijelic99/TBAgentApp.git
```

Step 2: Set up the environment

Create a duplicate of the `.env.example` file and name it `.env`. Fill in the required environment variables.

You may change the default `SECRET_KEY` and `FIRST_SUPERUSER_PASSWORD` values.

The mandatory values are as follows:

```
DOMAIN=

ENVIRONMENT=

PROJECT_NAME=
STACK_NAME=

BACKEND_CORS_ORIGINS=[] # Add the frontend URL here
SECRET_KEY=
FIRST_SUPERUSER=
FIRST_SUPERUSER_PASSWORD=
USERS_OPEN_REGISTRATION=False # Set to True if you want to allow open registration

POSTGRES_SERVER=
POSTGRES_PORT=5432
POSTGRES_DB=
POSTGRES_USER=
POSTGRES_PASSWORD=

DOCKER_IMAGE_BACKEND=backend
DOCKER_IMAGE_FRONTEND=frontend

TIMEZONE="Europe/Zurich" # Set your timezone

NAMESPACES=
KUBE_HOST=
KUBECONFIG_PATH=
GOOGLE_APPLICATION_CREDENTIALS_FILE=

PROMETHEUS_URL=
```

```
LANGCHAIN_TRACING_V2=true   # Use LangSmith for tracing
LANGCHAIN_ENDPOINT="https://api.smith.langchain.com"
LANGCHAIN_API_KEY=  # https://smith.langchain.com/settings
OPENAI_API_KEY=     # https://platform.openai.com/account/api-keys
LLM_MODEL="gpt-4o" # Set the model you want to use : gpt-4o, gpt-3.5-turbo or llama3
OLLAMA_BASE_URL="http://host.docker.internal:11434" # Only if you use LLM_MODEL=llama3. You
```

`PROMETHEUS_URL` is the URL of the Prometheus server that scrapes the metrics from the Kubernetes cluster. It generally looks like `http://prometheus-ip:9090`.

`KUBE_HOST` is the IP address of the Kubernetes cluster. It can be found in the cluster details of the Google Cloud Platform console.

`KUBECONFIG_PATH` is the path to the kubeconfig file directory. It is used to authenticate with the Kubernetes cluster. By default, on Windows it is `C:\Users\username\.kube` and on Linux it is `~/.kube`.

`GOOGLE_APPLICATION_CREDENTIALS_FILE` is the name of the Google Service Account JSON file. It should be placed among the kubeconfig files. You can download it from the Google Cloud Platform console.

By setting up emails environment variables, you can enable email based password recovery.

Step 3: Build and start the Docker containers

To build and start the Docker containers, run the following command:

```
docker-compose up -d
```

The `docker-compose.yaml` files are provided by the Full Stack FastAPI template.

If everything is set up correctly, you should be able to access the OpenAPI documentation at `http://localhost:8080/docs` and the frontend at `http://localhost:8080`.

If you encounter any issues, check the backend logs for any configuration errors.

Step 4: Set up the namespace to monitor

Currently, the agent is monitoring the namespaces you specify in the `backend/app/monitoring_agent/main.py` file. You can add or remove namespaces as needed.

```
NAMESPACES=["testing-apps"]
```

LangSmith

By enabling LangSmith for tracing, you can use the LangSmith service to trace the agent's reasoning process. You can sign up for a free account at LangSmith.

You will be able to see all the reasoning steps and the generated AI messages as well as the number of tokens used and the price of the reasoning process.

License