

Kubernetes AI Agent

A large language model agent capable of monitoring a Kubernetes cluster. This agent can be used to analyse metrics and logs to run diagnostics if any issue is detected, identifying the root cause of the problem. The agent will then provide a solution to the problem and document each step in an incident report.

This project is the result of the Bachelor's final project by Alen Bijelic during his final year at the School of Engineering and Management, Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud (HEIG-VD). The project was supervised by Prof. Marcel Graf.

Agent workflow

Technology Stack

- [Full Stack FastAPI Template](#) for the backend, frontend and DB.
- [LangGraph](#) and [LangChain](#) for the AI agent.
- [Google Kubernetes Engine](#) for the Kubernetes cluster.
- [Prometheus](#) for monitoring.

How To Use It

You can **just fork or clone** this repository and use it as is.

It just works.

How to Use a Private Repository

If you want to have a private repository, GitHub won't allow you to simply fork it as it doesn't allow changing the visibility of forks.

But you can do the following:

- Create a new GitHub repo, for example `my-full-stack`.
- Clone this repository manually, set the name with the name of the project you want to use, for example `my-full-stack`:

```
git clone git@github.com:tiangolo/full-stack-fastapi-template.git my-full-stack
```

- Enter into the new directory:

```
cd my-full-stack
```

- Set the new origin to your new repository, copy it from the GitHub interface, for example:

```
git remote set-url origin git@github.com:octocat/my-full-stack.git
```

- Add this repo as another "remote" to allow you to get updates later:

```
git remote add upstream git@github.com:tiangolo/full-stack-fastapi-template.git
```

- Push the code to your new repository:

```
git push -u origin master
```

Update From the Original Template

After cloning the repository, and after doing changes, you might want to get the latest changes from this original template.

- Make sure you added the original repository as a remote, you can check it with:

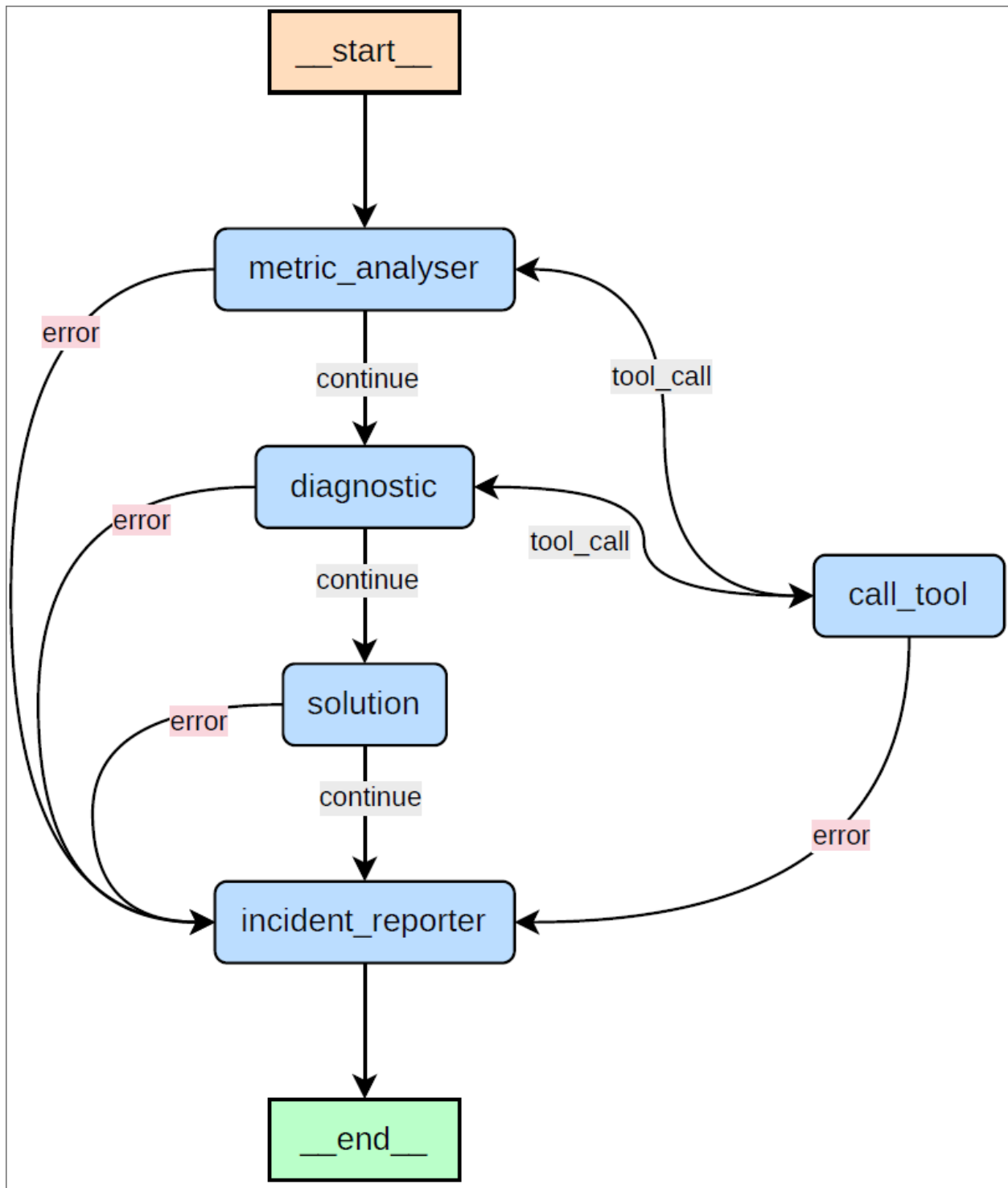


Figure 1: Graph flow

```
git remote -v
```

```
origin    git@github.com:octocat/my-full-stack.git (fetch)
origin    git@github.com:octocat/my-full-stack.git (push)
upstream  git@github.com:tiangolo/full-stack-fastapi-template.git (fetch)
upstream  git@github.com:tiangolo/full-stack-fastapi-template.git (push)
```

- Pull the latest changes without merging:

```
git pull --no-commit upstream master
```

This will download the latest changes from this template without committing them, that way you can check everything is right before committing.

- If there are conflicts, solve them in your editor.
- Once you are done, commit the changes:

```
git merge --continue
```

Configure

You can then update configs in the `.env` files to customize your configurations.

Before deploying it, make sure you change at least the values for:

- SECRET_KEY
- FIRST_SUPERUSER_PASSWORD
- POSTGRES_PASSWORD

You can (and should) pass these as environment variables from secrets.

Read the [deployment.md](#) docs for more details.

Generate Secret Keys

Some environment variables in the `.env` file have a default value of `changethis`.

You have to change them with a secret key, to generate secret keys you can run the following command:

```
python -c "import secrets; print(secrets.token_urlsafe(32))"
```

Copy the content and use that as password / secret key. And run that again to generate another secure key.

How To Use It - Alternative With Copier

This repository also supports generating a new project using [Copier](#).

It will copy all the files, ask you configuration questions, and update the `.env` files with your answers.

Install Copier

You can install Copier with:

```
pip install copier
```

Or better, if you have [pipx](#), you can run it with:

```
pipx install copier
```

Note: If you have `pipx`, installing `copier` is optional, you could run it directly.

Generate a Project With Copier

Decide a name for your new project's directory, you will use it below. For example, `my-awesome-project`.

Go to the directory that will be the parent of your project, and run the command with your project's name:

```
copier copy https://github.com/tiangolo/full-stack-fastapi-template my-awesome-project --trust
```

If you have `pipx` and you didn't install `copier`, you can run it directly:

```
pipx run copier copy https://github.com/tiangolo/full-stack-fastapi-template my-awesome-project --trust
```

Note the `--trust` option is necessary to be able to execute a [post-creation script](#) that updates your `.env` files.

Input Variables

Copier will ask you for some data, you might want to have at hand before generating the project.

But don't worry, you can just update any of that in the `.env` files afterwards.

The input variables, with their default values (some auto generated) are:

- **project_name:** (default: "FastAPI Project") The name of the project, shown to API users (in `.env`).
- **stack_name:** (default: "fastapi-project") The name of the stack used for Docker Compose labels and project name (no spaces, no periods) (in `.env`).
- **secret_key:** (default: "changethis") The secret key for the project, used for security, stored in `.env`, you can generate one with the method above.
- **first_superuser:** (default: "admin@example.com") The email of the first superuser (in `.env`).
- **first_superuser_password:** (default: "changethis") The password of the first superuser (in `.env`).
- **smtp_host:** (default: "") The SMTP server host to send emails, you can set it later in `.env`.
- **smtp_user:** (default: "") The SMTP server user to send emails, you can set it later in `.env`.
- **smtp_password:** (default: "") The SMTP server password to send emails, you can set it later in `.env`.
- **emails_from_email:** (default: "info@example.com") The email account to send emails from, you can set it later in `.env`.
- **postgres_password:** (default: "changethis") The password for the PostgreSQL database, stored in `.env`, you can generate one with the method above.
- **sentry_dsn:** (default: "") The DSN for Sentry, if you are using it, you can set it later in `.env`.

Backend Development

Backend docs: [backend/README.md](#).

Frontend Development

Frontend docs: [frontend/README.md](#).

Deployment

Deployment docs: [deployment.md](#).

Development

General development docs: [development.md](#).

This includes using Docker Compose, custom local domains, `.env` configurations, etc.

License

The Kubernetes AI Agent is licensed under the terms of the MIT license.