YAML Generator for Spheron Deployment

Project Overview:

The **YAML Generator** is a tool designed to generate **Infrastructure Composition Language (ICL)** files for deploying applications on **Spheron Network**. It enables users to create structured deployment configurations with an intuitive web interface, leveraging **Gemini AI** to assist with input validation and enhancements.

Features

- **Web-Based Interface**: Users can enter configuration details through an HTML page.
- **AI Assistance with Gemini**: Provides intelligent suggestions and validation.
- **ICL & YAML Generation**: Automatically creates valid YAML files for deployment.
- **Syntax Highlighting**: Uses Prism.js for clear YAML representation.
- **Docker & WSL Support**: Can be containerized and deployed on various platforms.

Technologies Used

- **Frontend**: HTML, CSS, JavaScript
- **Backend**: Python (Flask), Node.js (optional for frontend handling)
- **Styling**: Tailwind CSS
- **Syntax Highlighting**: Prism.js
- **Deployment Configuration**: Spheron ICL, Docker
- **AI Integration**: Gemini API

Installation & Setup

Prerequisites

Ensure you have the following installed:

- **Python 3.x**
- **Node.js & npm**
- **Docker & Docker Compose**
- **WSL (if using Windows)**

Clone the Repository

git clone https://github.com/yourusername/yaml-generator.git

cd yaml-generator

Install Dependencies

1. Backend Dependencies

pip install -r requirements.txt

2. Frontend Dependencies

npm install

Running the Application

1. Start the Backend Server

python app.py

2. Start the Frontend

npm start

The application will be available at http://localhost:5000.

Docker Deployment

1. Build the Docker Image

docker build -t yaml-generator .

2. Run the Container

docker run -p 5000:5000 yaml-generator

Spheron Deployment

1. Create a spheron-deployment.yaml file

```
version: "1.0"
services:
  yaml-generator:
    image: your-dockerhub-username/yaml-generator:latest
    expose:
      - port: 5000
        as: 5000
        to:
          - global: true
```

```
    env:
     - API_KEY=your-api-key
profiles:
  name: yaml-deployment
  mode: provider
  duration: 1h
  tier:
   - community
  compute:
   yaml-generator:
    resources:
     cpu:
      units: 1
     memory:
      size: 2Gi
     storage:
      - size: 10Gi
  placement:
   global:
    attributes:
     region: us-east
    pricing:
     yaml-generator:
      token: CST
      amount: 2
```

2. Deploy on Spheron

```
spheronctl deploy -f spheron-deployment.yaml
```

File Structure

```
yaml-generator/
```

```
├── static/
│   ├── prism.css
│   ├── prism.js
│   ├── script.js
│   ├── styles.css
│
├── templates/
│   ├── index.html
│
├── .env
├── app.py
├── Dockerfile
├── index.tsx
├── package-lock.json
├── prism-custom.css
├── README.md
├── requirements.txt
├── script.js
├── spheron-deployment.yaml
├── styles.css
├── tailwind.config.ts
```

API Endpoints

1. Generate YAML

Endpoint: /generate **Method:** POST **Description:** Generates a YAML file based on user input. **Request Body:**

```
{
  "service_name": "gpu-test",
  "image": "ghcr.io/open-webui/open-webui:ollama",
  "port": 8888
}
```

Response:

```
version: "1.0"
services:
  gpu-test:
    image: ghcr.io/open-webui/open-webui:ollama
    expose:
      - port: 8888
        as: 8888
        to:
          - global: true
```

Contributing

1. Fork the repository.
2. Create a new branch (git checkout -b feature-name).
3. Commit changes (git commit -m 'Add new feature').
4. Push to branch (git push origin feature-name).
5. Open a Pull Request.

Contact

For any issues or feature requests, create an issue on GitHub.