Swarms Installation Guide

      <img
        width="850"
        src="https://github.com/kyegomez/swarms/raw/master/images/swarmslogobanner.png"
      >

You can install `swarms` with pip in a
[**Python>=3.10**](https://www.python.org/) environment.

Prerequisites

Before you begin, ensure you have the following installed:

- Python 3.10 or higher: [Download Python](https://www.python.org/)
- pip (specific version recommended): `pip >= 21.0`
- git (for cloning the repository): [Download Git](https://git-scm.com/)

Installation Options

=== "pip (Recommended)"

Headless Installation

The headless installation of `swarms` is designed for environments where graphical user interfaces (GUI) are not needed, making it more lightweight and suitable for server-side applications.

```bash
pip install swarms
```

=== "Development Installation"

=== "Using virtualenv"

1. **Clone the repository and navigate to the root directory:**

    ```bash
    git clone https://github.com/kyegomez/swarms.git
    cd swarms
    ```

2. **Setup Python environment and activate it:**

    ```bash
    python3 -m venv venv
    ```

source venv/bin/activate

pip install --upgrade pip

```

3. **Install Swarms:**

- Headless install:

```bash
pip install -e .
```

- Desktop install:

```bash
pip install -e .[desktop]
```

=== "Using Anaconda"

1. **Create and activate an Anaconda environment:**

```bash
conda create -n swarms python=3.10
conda activate swarms
```
```

2. **Clone the repository and navigate to the root directory:**

   ```bash
   git clone https://github.com/kyegomez/swarms.git
   cd swarms
   ```

3. **Install Swarms:**

   - Headless install:

     ```bash
     pip install -e .
     ```

   - Desktop install:

     ```bash
     pip install -e .[desktop]
     ```

=== "Using Poetry"

1. **Clone the repository and navigate to the root directory:**

```bash
git clone https://github.com/kyegomez/swarms.git
cd swarms
```

2. **Setup Python environment and activate it:**

```bash
poetry env use python3.10
poetry shell
```

3. **Install Swarms:**

- Headless install:

```bash
poetry install
```

- Desktop install:

```bash
poetry install --extras "desktop"
```

=== "Using Docker"

Docker is an excellent option for creating isolated and reproducible environments, suitable for both development and production. Contact us if there are any issues with the docker setup

1. **Pull the Docker image:**

   ```bash
   docker pull swarmscorp/swarms:tagname

   ```

2. **Run the Docker container:**

   ```bash
   docker run -it --rm swarmscorp/swarms:tagname
   ```

3. **Build and run a custom Docker image:**

   ```dockerfile
   # Use Python 3.11 instead of 3.13
   FROM python:3.11-slim

   # Set environment variables
   ENV PYTHONDONTWRITEBYTECODE=1 \
   ```

```
    PYTHONUNBUFFERED=1 \
    WORKSPACE_DIR="agent_workspace" \
    OPENAI_API_KEY="your_swarm_api_key_here"

# Set the working directory
WORKDIR /usr/src/swarms

# Install system dependencies
RUN apt-get update && apt-get install -y \
    build-essential \
    gcc \
    g++ \
    gfortran \
    && rm -rf /var/lib/apt/lists/*

# Install swarms package
RUN pip3 install -U swarm-models
RUN pip3 install -U swarms

# Copy the application
COPY . .
```

=== "Using Kubernetes"

Kubernetes provides an automated way to deploy, scale, and manage containerized applications.

1. **Create a Deployment YAML file:**

   ```yaml
   apiVersion: apps/v1
   kind: Deployment
   metadata:
     name: swarms-deployment
   spec:
     replicas: 3
     selector:
       matchLabels:
         app: swarms
     template:
       metadata:
         labels:
           app: swarms
       spec:
         containers:
         - name: swarms
           image: kyegomez/swarms
           ports:
           - containerPort: 8080
   ```

2. **Apply the Deployment:**

```bash
kubectl apply -f deployment.yaml
```

3. **Expose the Deployment:**

```bash
kubectl expose deployment swarms-deployment --type=LoadBalancer --name=swarms-service
```

=== "CI/CD Pipelines"

Integrating Swarms into your CI/CD pipeline ensures automated testing and deployment.

Using GitHub Actions

```yaml
# .github/workflows/ci.yml
name: CI

on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]
```

```
jobs:
  build:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v2
    - name: Set up Python
      uses: actions/setup-python@v2
      with:
        python-version: 3.10
    - name: Install dependencies
      run: |
        python -m venv venv
        source venv/bin/activate
        pip install --upgrade pip
        pip install -e .
    - name: Run tests
      run: |
        source venv/bin/activate
        pytest
```

Using Jenkins

```groovy
pipeline {
    agent any

    stages {
        stage('Clone repository') {
            steps {
                git 'https://github.com/kyegomez/swarms.git'
            }
        }
        stage('Setup Python') {
            steps {
                sh 'python3 -m venv venv'
                sh 'source venv/bin/activate && pip install --upgrade pip'
            }
        }
        stage('Install dependencies') {
            steps {
                sh 'source venv/bin/activate && pip install -e .'
            }
        }
        stage('Run tests') {
            steps {
                sh 'source venv/bin/activate && pytest'
            }
        }
```

```
    }
}
```

Javascript

=== "NPM install (Work in Progress)"

Get started with the NPM implementation of Swarms:

```bash
npm install swarms-js
```