

Kubernetes with AI

K8s AI

- [K8sGPT](#)
- [KubectI AI](#)
- [KoPylot](#)

K8sGPT

Kubectl AI

Kubectl AI 是一个可以生成 OpenAI GPT 模型部署 manifest YAML 文件的工具。Kubectl 可以在 Github 上找到，并可以通过 Homebrew 或 Krew 安装。

Kubectl AI Github

本文档将指导您如何安装 Kubectl AI。首先，您需要安装 Git。然后，您可以通过 Homebrew 或 Krew 安装 Kubectl AI。

安装 Git

Kubectl AI 需要 Git 来管理代码。您可以通过 Homebrew 或 Krew 安装 Git。

Krew 是一个用于安装 Kubernetes 插件的工具。您可以通过 [Installing Krew](#) 安装 Krew。

本文档将指导您如何安装 Kubectl AI。首先，您需要安装 Git。然后，您可以通过 Homebrew 或 Krew 安装 Kubectl AI。

本文档将指导您如何安装 Git。Git 是一个用于管理代码的工具。您可以通过 Github 安装 Git。

```
# Red Hat 安装 git
dnf install git

# Ubuntu 安装 git (root 用户)
apt install git
```

本文档将指导您如何安装 Krew。Krew 是一个用于安装 Kubernetes 插件的工具。您可以通过 Github 安装 Krew。

```
(
  set -x; cd "$(mktemp -d)" &&
  OS="$(uname | tr '[:upper:]' '[:lower:]')" &&
  ARCH="$(uname -m | sed -e 's/x86_64/amd64/' -e 's/(arm|aarch64)/\1/' -e 's/aarch64/arm64/')" &&
  KREW="krew-${OS}_${ARCH}" &&
  curl -fsSLO "https://github.com/kubernetes-sigs/krew/releases/latest/download/${KREW}.tar.gz" &&
```

```

tar zxvf "${KREW}.tar.gz" &&
./"${KREW}" install krew
)

# 检查安装
++ mktemp -d
+ cd /tmp/tmp.sQZg2NfgQ0
++ uname
++ tr '[:upper:]' '[:lower:]'
+ OS=linux
++ uname -m
++ sed -e s/x86_64/amd64/ -e 's/\(arm\)\(64\)\{0,1\}/aarch64/' -e 's/aarch64$/arm64/'
+ ARCH=amd64
+ KREW=krew-linux_amd64
+ curl -fsSLO https://github.com/kubernetes-sigs/krew/releases/latest/download/krew-linux_amd64.tar.gz
+ tar zxvf krew-linux_amd64.tar.gz
./LICENSE
./krew-linux_amd64
+ ./krew-linux_amd64 install krew
Adding "default" plugin index from https://github.com/kubernetes-sigs/krew-index.git.
Updated the local copy of plugin index.
Installing plugin: krew
Installed plugin: krew
\
| Use this plugin:
| kubectl krew
| Documentation:
| https://krew.sigs.k8s.io/
| Caveats:
| \
| | krew is now installed! To start using kubectl plugins, you need to add
| | krew's installation directory to your PATH:
| |
| | * macOS/Linux:
| |   - Add the following to your ~/.bashrc or ~/.zshrc:
| |     export PATH="${KREW_ROOT:-$HOME/.krew}/bin:$PATH"
| |   - Restart your shell.
| |
| | * Windows: Add %USERPROFILE%\krew\bin to your PATH environment variable
| |

```

```
| | To list krew commands and to get help, run:
| | $ kubectl krew
| | For a full list of available plugins, run:
| | $ kubectl krew search
| |
| | You can find documentation at
| | https://krew.sigs.k8s.io/docs/user-guide/quickstart/.
| /
/
```

İçerik Krew'ı kurmak için Krew'ı bash Shell PATH'ına eklemek için

```
# PATH'ı kubectl krew'ın PATH'ına eklemek için
export PATH="${KREW_ROOT:-$HOME/.krew}/bin:$PATH"
```

İçerik Shell `source ~/.bashrc` komutu ile Shell'imize eklediğimiz Krew'ı aktif etmek için

```
kubectl krew
```

krew is the kubectl plugin manager.

Krew'ı kurduktan sonra Krew index'ini kubectl ai ile ekliyoruz

```
# index'ini Krew'ın PATH'ına eklemek için
# Krew'ın PATH'ına eklemek için
kubectl krew index add kubectl-ai https://github.com/sozercan/kubectl-ai
```

WARNING: You have added a new index from "https://github.com/sozercan/kubectl-ai"
The plugins in this index are not audited for security by the Krew maintainers.
Install them at your own risk.

İçerik Krew'ı kurduktan sonra Kubectl AI'ı kurmak için

```
kubectl krew install kubectl-ai/kubectl-ai
```

Updated the local copy of plugin index.

Updated the local copy of plugin index "kubectl-ai".

Installing plugin: kubectl-ai

Installed plugin: kubectl-ai

\

| Use this plugin:

| `kubectl kubectl-ai`

| Caveats:

| \

| | This plugin requires an OpenAI key.

| /

/

❏ ❏

❏ `kubectl AI` ❏ ❏ ❏, ❏ ❏ ❏ OpenAI API❏ ❏ ❏ ❏.

❏ ❏ ❏ ❏ ❏ ❏. ❏ ❏ ❏ ❏ ❏ ❏.

```
export OPENAI_API_KEY=<OpenAI ❏>
```

❏❏ OpenAI GPT ❏❏ ❏ ❏ ❏. ❏❏❏❏ `gpt-3.5-turbo-0301` ❏ ❏ ❏ ❏.

```
export OPENAI_DEPLOYMENT_NAME=<OpenAI GPT ❏ ❏>
```

❏❏❏ ❏ ❏❏ ❏❏❏:

- `code-davinci-002`
- `text-davinci-003`
- `gpt-3.5-turbo`
- `gpt-3.5-turbo-0301` (default)
- `gpt-4-0314`
- `gpt-4-32k-0314`

❏ `kubectl AI`❏ ❏ ❏ ❏ ❏❏❏ ❏❏❏ ❏ ❏❏.

❏❏❏ ❏ ❏❏ ❏❏❏❏ ❏ ❏❏

```
# 创建 nginx 部署
```

```
kubectll kubectl-ai "create an nginx deployment with 3 replicas"
```

```
? Attempting to apply the following manifest:
```

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  name: nginx-deployment
```

```
spec:
```

```
  replicas: 3
```

```
  selector:
```

```
    matchLabels:
```

```
      app: nginx
```

```
  template:
```

```
    metadata:
```

```
      labels:
```

```
        app: nginx
```

```
    spec:
```

```
      containers:
```

```
        - name: nginx
```

```
          image: nginx:1.7.9
```

```
          ports:
```

```
            - containerPort: 80
```

```
Use the arrow keys to navigate: ↓ ↑ → ←
```

```
? Would you like to apply this? [Reprompt/Apply/Don't Apply]:
```

```
+   Reprompt
```

```
  ▶ Apply
```

```
    Don't Apply
```

```
创建 nginx 部署 在 部署 YAML 中 将 副本数 从 3 修改为 2 并 使用 最新的 nginx 镜像。 输入 "Reprompt" 来 重新 提示。
```

```
# replica 2 使用 latest nginx 镜像 部署
```

```
Reprompt: Use the latest nginx image and reduce the replica count to 2
```

```
Attempting to apply the following manifest:
```

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
name: nginx-deployment
spec:
  replicas: 2# 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest# latest
        ports:
        - containerPort: 80
```

"Reprompt" 尝试添加 Service Object

```
# nodeport 添加 nodeport
Reprompt: Include a service nodeport for the nginx deployment
Attempting to apply the following manifest:
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
```


...

✓ Apply

[[[]]] get all [[[]]] [[[]]]

2 Replica

NAME	DESIRED	CURRENT	READY	AGE
------	---------	---------	-------	-----

replicaset.apps/nginx-deployment-6b7f675859 2 2 2 5s

nginx가 30080 포트에 실행되고 있습니다



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

nginx가 30080 포트에 실행되고 있습니다. nginx가 30080 포트에 실행되고 있습니다.

Kubectl AI가 실패했습니다. 오류 메시지는 다음과 같습니다:

```
[root@node1 ~]# kubectl kubectl-ai "delete nginx-service and nginx-deployment"
Error: [429:server_error] The server had an error while processing your request. Sorry about that!
```

KoPylot

An AI-powered assistant for Kubernetes

The blog: <https://medium.com/@thiagoalves/introducing-kopylot-a-kubernetes-ai-assistant-264cff0e7846>

The GitHub: <https://github.com/avsthiago/kopylot>

KoPylot features

At the current version, KoPylot has four main features. These features can be translated into subcommands for the `kopylot` CLI. The subcommands are **Audit**, **Chat**, **Ctl**, and **Diagnose**.

☐☐ Audit:

Audit resources, such as pods, deployments, and services. KoPylot will take a single resource and look for vulnerabilities based on its manifest file.

☐☐ Chat:

Ask KoPylot in plain English to generate kubectl commands. You will be able to review the command before running it ☐☐.

Diagnose:

You can use the diagnose tool to help you debug the different components of your application, such as pods, deployments, and services. The diagnose command will list for you possible fixes for the broken resource.

Ctl:

A wrapper around kubectl. All the arguments passed to the `ctl` subcommand are interpreted by kubectl.