

Reproduktion einer virtuellen Umgebung für das Deployment

Mit Pipenv können Sie eine Python-Umgebung für die Produktion reproduzieren, um sicherzustellen, dass alle Abhängigkeiten und Versionen exakt wie in der Entwicklungsumgebung installiert werden. Dies ist besonders wichtig für das Deployment, da es die Konsistenz der Umgebung garantiert und Fehler aufgrund von Versionsabweichungen vermeidet.

Hier sind die Schritte zur Reproduktion einer virtuellen Umgebung mit Pipenv.

Voraussetzungen

- Stellen Sie sicher, dass Pipenv installiert ist. Falls nicht, können Sie es mit folgendem Befehl installieren:

```
pip install pipenv
```

Schritte zur Reproduktion der Umgebung

1. Projektdateien inklusive **Pipfile** und **Pipfile.lock** vorbereiten

Um eine Umgebung exakt zu reproduzieren, benötigen Sie die beiden Dateien **Pipfile** und **Pipfile.lock**, die alle Abhängigkeiten und exakten Versionen der Pakete dokumentieren.

2. Neue virtuelle Umgebung mit Pipenv erstellen

Navigieren Sie zum Verzeichnis des Projekts und stellen Sie sicher, dass die **Pipfile** und **Pipfile.lock** dort vorhanden sind.

```
cd /pfad/zum/projekt
```

3. Reproduzieren der Umgebung

Verwenden Sie den Befehl **pipenv sync**, um eine exakte Kopie der Umgebung basierend auf der **Pipfile.lock** zu erstellen. Dieser Befehl stellt sicher, dass alle Pakete und deren Versionen identisch zur Entwicklungsumgebung sind.

```
pipenv sync
```

Der **pipenv sync**-Befehl verwendet die **Pipfile.lock**, um sicherzustellen, dass exakt dieselben Versionen der Abhängigkeiten installiert werden, die im Entwicklungsprozess verwendet wurden.

4. Optional: Nur Produktionsabhängigkeiten installieren

Falls Sie nur die Pakete installieren möchten, die für den Betrieb benötigt werden, ohne die Entwicklungsabhängigkeiten, verwenden Sie das `--ignore-pipfile`-Flag zusammen mit `pipenv install --deploy --ignore-pipfile`:

```
pipenv install --deploy --ignore-pipfile
```

- Das `--deploy`-Flag stellt sicher, dass die Installation fehlschlägt, wenn die `Pipfile.lock` nicht mit der `Pipfile` übereinstimmt oder eine Inkompatibilität vorliegt.
- Das `--ignore-pipfile`-Flag sorgt dafür, dass ausschließlich die `Pipfile.lock` verwendet wird und keine Änderungen aus der `Pipfile` übernommen werden.

5. Virtuelle Umgebung für das Deployment aktivieren

Sie können die Umgebung aktivieren, um Tests durchzuführen oder den Code direkt in der reproduzierten Umgebung zu starten:

```
pipenv shell
```

Alternativ können Sie einzelne Befehle in der Umgebung ohne `shell`-Aktivierung ausführen:

```
pipenv run python app.py
```

Zusammenfassung der Befehle

```
# Gehe zum Projektverzeichnis
cd /pfad/zum/projekt

# Erstelle eine exakte Kopie der Umgebung basierend auf Pipfile.lock
pipenv sync

# Nur Produktionsabhängigkeiten installieren
pipenv install --deploy --ignore-pipfile

# Umgebung für Tests aktivieren
pipenv shell

# Ausführen eines Befehls in der Umgebung
pipenv run python app.py
```