Apa itu Python?

Python adalah bahasa pemrograman open-source yang ditafsirkan, interaktif, berorientasi objek. Ini menggabungkan modul, pengecualian, pengetikan dinamis, tipe data dinamis tingkat sangat tinggi, dan kelas. Python menggabungkan kekuatan luar biasa dengan sintaks yang sangat jelas. Ini memiliki antarmuka ke banyak panggilan sistem dan pustaka, serta berbagai sistem jendela, dan dapat diperluas dalam C atau C++. Ini juga dapat digunakan sebagai bahasa ekstensi untuk aplikasi yang membutuhkan antarmuka yang dapat diprogram. Terakhir, Python bersifat portabel: ia berjalan di banyak varian Unix, di Mac, dan di Windows 2000 dan yang lebih baru.

wikipedia.org/wiki/Python\_(bahasa\_pemrograman)

logo

Bagaimana cara menggunakan gambar ini?

Buat Dockerfile di proyek aplikasi Python Anda

DARI python:3

WORKDIR /usr/src/app

SALIN persyaratan.txt ./

JALANKAN pip install --no-cache-dir -r requirements.txt

SALIN. .

CMD [ "python", "./your-daemon-or-script.py" ]

atau (jika Anda perlu menggunakan Python 2):

DARI python: 2

WORKDIR /usr/src/app

SALIN persyaratan.txt ./

JALANKAN pip install --no-cache-dir -r requirements.txt

SALIN. .

CMD [ "python", "./your-daemon-or-script.py" ]

Anda kemudian dapat membangun dan menjalankan gambar Docker:

$ docker build -t my-python-app .

$ docker run -it --rm --name my-running-app my-python-app

Jalankan satu skrip Python

Untuk banyak proyek file tunggal yang sederhana, Anda mungkin merasa tidak nyaman untuk menulis file Docker yang lengkap. Dalam kasus seperti itu, Anda dapat menjalankan skrip Python dengan menggunakan gambar Python Docker secara langsung:

$ docker run -it --rm --name my-running-script -v "$PWD":/usr/src/myapp -w /usr/src/myapp python:3 python your-daemon-or-script.py

atau (sekali lagi, jika Anda perlu menggunakan Python 2):

$ docker run -it --rm --name my-running-script -v "$PWD":/usr/src/myapp -w /usr/src/myapp python:2 python your-daemon-or-script.py

Beberapa versi Python dalam gambar

Dalam varian non python:slim akan ada tambahan python lama yang dapat dieksekusi di /usr/bin/python sedangkan /usr/local/bin/python yang lebih baru yang diinginkan adalah pilihan default di $PATH. Ini adalah efek samping yang disayangkan dari penggunaan gambar buildpack-deps dalam varian non-slim

Varian Gambar

Gambar python datang dalam banyak rasa, masing-masing dirancang untuk kasus penggunaan tertentu.

python:<versi>

Ini adalah gambar de facto. Jika Anda tidak yakin tentang apa kebutuhan Anda, Anda mungkin ingin menggunakan yang ini. Ini dirancang untuk digunakan baik sebagai wadah sekali pakai (pasang kode sumber Anda dan mulai wadah untuk memulai aplikasi Anda), serta dasar untuk membangun gambar lain.

Beberapa tag ini mungkin memiliki nama seperti bullseye atau buster di dalamnya. Ini adalah nama kode suite untuk rilis Debian dan menunjukkan rilis mana yang didasarkan pada gambar. Jika gambar Anda perlu menginstal paket tambahan apa pun selain yang disertakan dengan gambar, Anda mungkin ingin menentukan salah satunya secara eksplisit untuk meminimalkan kerusakan saat ada rilis baru Debian.

Tag ini didasarkan pada buildpack-deps. buildpack-deps dirancang untuk rata-rata pengguna Docker yang memiliki banyak gambar di sistem mereka. Secara desain, ia memiliki sejumlah besar paket Debian yang sangat umum. Ini mengurangi jumlah paket yang perlu diinstal oleh gambar yang berasal darinya, sehingga mengurangi ukuran keseluruhan semua gambar di sistem Anda.

python:<versi>-slim

Gambar ini tidak berisi paket umum yang terdapat dalam tag default dan hanya berisi paket minimal yang diperlukan untuk menjalankan python. Kecuali Anda bekerja di lingkungan di mana hanya gambar python yang akan digunakan dan Anda memiliki batasan ruang, kami sangat menyarankan untuk menggunakan gambar default dari repositori ini.

python:<versi>-alpine

Gambar ini didasarkan pada proyek Alpine Linux yang populer, tersedia dalam gambar resmi alpine. Alpine Linux jauh lebih kecil daripada kebanyakan gambar dasar distribusi (~5MB), dan dengan demikian menghasilkan gambar yang jauh lebih ramping secara umum.

Varian ini berguna ketika ukuran gambar akhir sekecil mungkin menjadi perhatian utama Anda. Peringatan utama yang perlu diperhatikan adalah bahwa ia menggunakan musl libc alih-alih glibc dan teman-teman, sehingga perangkat lunak akan sering mengalami masalah tergantung pada kedalaman persyaratan/asumsi libc mereka. Lihat utas komentar Berita Peretas ini untuk diskusi lebih lanjut tentang masalah yang mungkin muncul dan beberapa perbandingan pro/kontra menggunakan gambar berbasis Alpine.

Untuk meminimalkan ukuran gambar, jarang ada alat terkait tambahan (seperti git atau bash) untuk disertakan dalam gambar berbasis Alpine. Menggunakan gambar ini sebagai dasar, tambahkan hal-hal yang Anda butuhkan di Dockerfile Anda sendiri (lihat deskripsi gambar alpine untuk contoh cara menginstal paket jika Anda tidak terbiasa).

python:<versi>-windowsservercore

Gambar ini didasarkan pada Windows Server Core (microsoft/windowsservercore). Dengan demikian, ini hanya berfungsi di tempat-tempat yang menggunakan gambar itu, seperti Windows 10 Professional/Enterprise (Edisi Ulang Tahun) atau Windows Server 2016.

Untuk informasi tentang cara menjalankan Docker di Windows, silakan lihat panduan "Mulai Cepat" yang relevan yang disediakan oleh Microsoft:

Mulai Cepat Server Windows

Mulai Cepat Windows 10

Lisensi

Lihat informasi lisensi untuk Python 2 dan Python 3.

Seperti semua gambar Docker, ini kemungkinan juga berisi perangkat lunak lain yang mungkin berada di bawah lisensi lain (seperti Bash, dll dari distribusi dasar, bersama dengan dependensi langsung atau tidak langsung dari perangkat lunak utama yang terkandung).

Beberapa informasi lisensi tambahan yang dapat dideteksi secara otomatis dapat ditemukan di direktori python/ repositori repo-info.

Untuk penggunaan gambar yang dibuat sebelumnya, pengguna gambar bertanggung jawab untuk memastikan bahwa setiap penggunaan gambar ini sesuai dengan lisensi yang relevan untuk semua perangkat lunak yang ada di dalamnya.