



# DEPLOY DE APLICAÇÕES PYTHON COM DOCKER

## Como disponibilizar sua aplicação em um servidor só com Docker

# O QUE VAMOS FALAR

---

**Docker**  
porque ele é tão útil

01



**docker-compose**  
como funciona

02



**Nginx**  
como proxy reverso

03



04

**Supervisor**  
cuidando para não ficar off



05

**Gunicorn**  
o unicórnio verde



06

**Mão na massa**  
como fazer o deploy

# QUEM SOU EU

---



**SÉRGIO BERLOTTO JR**

[sergio.berlotto@gmail.com](mailto:sergio.berlotto@gmail.com)

Desenvolvedor Python há mais de 10 anos

Especialidade em: backend

Formado Técnico em Proc. de Dados

Cursando: Big Data e Inteligência Analítica (Uniassevi)

Trabalho como desenvolvedor, Scrum Master e Tech Lead na  
Jetbov (SC)

[www.jetbov.com](http://www.jetbov.com)

-> Gestão de fazendas de pecuária de corte



# VAMOS LÁ

---

Vamos ver um pouco de cada parte antes de passar para o código, para entender qual a função de cada uma delas...



# Docker

Docker é plataforma Open Source, escrita em Go, que fornece uma camada de abstração e automação para virtualização no Windows e no Linux, através de containers.

Containers docker são ambientes dentro do sistema operacional, que executam processos de forma isolada e controlada.

## Vantagens:

- Economia de recursos computacionais
- Rapidez na movimentação dos containers
- Compartilhamento de recursos
- Padronização e replicação
- Gerenciamento avançado





# docker-compose

docker-compose é uma ferramenta para a criação e execução de múltiplos containers de aplicação.

Através de um arquivo de configuração você consegue interligar vários containers, várias aplicações, compartilhar e padronizar o uso dos recursos e controlá-los de forma fácil.

A leitura e construção das configurações é bem simples de entender e é usado para definir como será o ambiente de sua aplicação, além disto, usando um único comando você criará e iniciará todos os serviços definidos.





# Nginx

---

Servidor http e de proxy reverso muito rápido. Criado para ser não-bloqueante, agiliza a resposta das requisições http para do servidor para o cliente, além de conseguir responder muitos clientes ao mesmo tempo.

É open-source, tem muita documentação disponível e muito configurável. É altamente utilizado nos servidores web mundo a fora.

É fácil de configurar como proxy reverso, como no caso que vamos ver logo mais a frente.

Podemos também configurar acesso seguro criptografado (https), balanceamento de carga, cache e compressão de dados.

Um proxy reverso é um servidor de rede geralmente instalado para ficar na frente de um servidor Web. Todas as conexões originadas externamente são roteadas por ele que repassa o tráfego de rede recebido para um conjunto de servidores (containers), tornando-o a única interface para as requisições externas.

# NGINX



# Supervisor

---

É um sistema de monitoramento de processos.

É uma ferramenta que monitora e controla os processos no linux para que não fiquem fora de operação. O Supervisor roda como um daemon no ambiente, e com pouca configuração conseguimos definir quais os processos ele irá monitorar e como isso irá ocorrer.

É bastante configurável, é criado em Python e muito leve.





# Gunicorn

---

Green Unicorn é um servidor http WSGI para sistemas Unix/Linux.

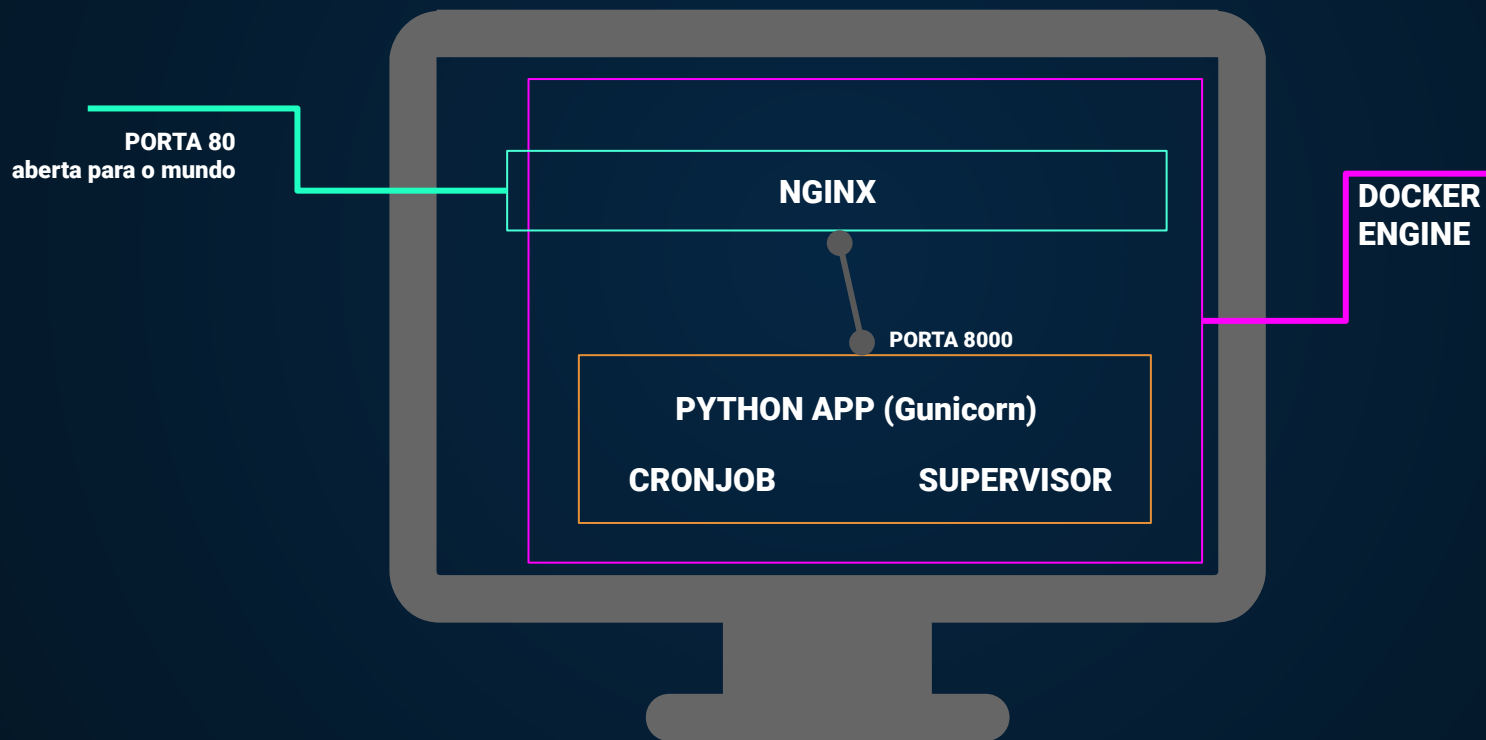
Ele trabalha como o executor da sua aplicação Python e sabe trabalhar com as várias requisições que chegam do servidor http.

Ele é basicamente o conector da sua aplicação Python com o servidor http que retorna os dados para o usuário que requisitou. É ele que executa o python para rodar sua aplicação.

Ambos (webserver e gunicorn) conversam através do protocolo conhecido como WSGI (**Web Server Gateway Interface**)




# Uma foto de tudo





Talk is cheap.  
Show me the code.

Linus Torvalds


 quote fancy



# OBRIGADO!

sergio.berlotto@gmail.com  
jetbov.com

 @berlottocdd

 @sberlotto

[https://github.com/SummitPython/elt\\_docker\\_deploy](https://github.com/SummitPython/elt_docker_deploy)



# CRÉDITOS DA APRESENTAÇÃO

Esta apresentação foi criada utilizando os recursos visuais dos sites abaixo.

- Presentation template by Slidesgo
- Icons by Flaticon
- Infographics by Freepik
- Images created by Freepik
- Author introduction slide photo created by Freepik
- Text & Image slide photo created by Freepik

Fontes sobre informações nos slides estão citadas no rodapé do próprio slide.

