



Dockumo: um gestor gráfico para containers Docker em Python e NiceGUI

Publicado em 2026-06-15 15:20:16



BOX DE FACTOS

- **Nome do projecto:** Dockumo Manager.
- **Objectivo:** gestão gráfica de containers Docker através de uma interface web em NiceGUI.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

- **Funções principais:** criar, importar imagens, listar, arrancar, parar, reiniciar, alterar e eliminar containers.
- **Inspiração conceptual:** Virtumo, mas aplicado à gestão de containers Docker.
- **Regra de ouro:** nunca expor directamente à Internet uma interface com acesso ao Docker daemon.

Dockumo: um gestor gráfico para containers Docker em Python e NiceGUI

O Dockumo nasce de uma ideia simples: se é possível gerir máquinas virtuais através de uma aplicação gráfica própria, também deve ser possível gerir

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

com aigniaade e cafe.

1. A origem da ideia

Depois do desenvolvimento conceptual do **Virtumo**, pensado para a gestão de máquinas virtuais, surgiu naturalmente a pergunta seguinte: porque não aplicar uma lógica semelhante ao universo dos containers Docker?

As máquinas virtuais continuam essenciais em muitas arquitecturas, sobretudo quando se pretende isolamento forte, sistemas operativos completos e ambientes persistentes. Mas os containers conquistaram um espaço próprio: são leves, rápidos, fáceis de recriar e muito úteis para serviços autónomos, aplicações web, bases de dados de teste, ferramentas administrativas, micro-serviços e laboratórios de desenvolvimento.

O **Dockumo Manager** nasce precisamente nesse território intermédio: não pretende substituir ferramentas como Portainer, Rancher ou Kubernetes. Pretende ser uma aplicação simples, controlada, legível e adaptável, construída em Python, para gerir containers Docker num servidor local, numa rede interna ou num ambiente de laboratório.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

serviços, parar instâncias, importar imagens, criar novos containers com portas, volumes e variáveis de ambiente são tarefas simples, mas que exigem atenção.

A linha de comandos é excelente para quem sabe exactamente o que está a fazer. O problema começa quando se pretende uma visão global, rápida e operacional do estado do sistema. Nesse ponto, uma interface gráfica torna-se útil: não para esconder a técnica, mas para a organizar.

O Dockumo foi pensado com esse objectivo: colocar num único painel as operações essenciais de gestão de containers, mantendo a simplicidade e evitando aquela tragédia moderna em que uma ferramenta criada para simplificar acaba por precisar de três cursos, dois certificados e uma assinatura empresarial.

3. Arquitectura geral

A arquitectura do Dockumo é deliberadamente simples:

- **NiceGUI** fornece a interface gráfica web.
- **Docker SDK for Python** comunica com o Docker Engine.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

utilizador.

O fluxo lógico é o seguinte:

Utilizador



Interface NiceGUI



Código Python do Dockumo



Docker SDK for Python



Docker Engine



Containers e imagens Docker

Esta arquitectura tem uma vantagem importante: todo o controlo fica concentrado numa aplicação Python, facilmente auditável, modificável e extensível. Não há aqui magia negra, apenas código a chamar uma API. O que, convenhamos, já é suficientemente assustador quando mal feito.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

- **Visualizar containers:** listagem de containers activos e parados.
- **Consultar estado:** indicação visual de containers em execução, parados ou com erro.
- **Criar containers:** criação manual a partir de uma imagem Docker.
- **Containers pré-definidos:** templates rápidos para serviços comuns.
- **Importar imagens:** operação de pull a partir do Docker Hub ou outro repositório configurado.
- **Arrancar containers:** função run/start.
- **Parar containers:** stop controlado.
- **Reiniciar containers:** restart.
- **Alterar containers:** nome, limites de memória, CPU e política de reinício.
- **Eliminar containers:** remoção controlada, com confirmação.
- **Ver logs:** consulta dos registos recentes de execução.
- **Inspect JSON:** visualização técnica da configuração completa do container.



5. Containers pré-definidos

Uma das funções mais úteis é a criação de containers a partir de modelos pré-definidos. A ideia é permitir que serviços comuns possam ser criados rapidamente, sem repetir manualmente a mesma configuração.

Entre os templates iniciais estão:

- **Nginx:** servidor web leve e rápido.
- **Redis:** serviço de cache e armazenamento chave-valor.
- **MariaDB:** base de dados relacional.
- **PostgreSQL:** base de dados relacional avançada.
- **Adminer:** ferramenta web para administração de bases de dados.
- **Portainer CE:** painel de gestão Docker.
- **Uptime Kuma:** monitorização de disponibilidade de serviços.

Estes modelos permitem acelerar a criação de serviços úteis num servidor doméstico, laboratório técnico ou pequena infra-estrutura interna. Naturalmente, qualquer template

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

O Dockumo permite importar imagens Docker através da operação de pull. Na prática, isto significa ir buscar uma imagem a um repositório, como o Docker Hub, e deixá-la disponível localmente para criação de containers.

Esta função é especialmente útil quando se pretende preparar previamente um conjunto de imagens, evitando depender da linha de comandos para cada operação. O utilizador pode importar imagens como:

```
nginx:latest
mariadb:latest
postgres:latest
redis:latest
adminer:latest
louislam/uptime-kuma:latest
```

Apesar da comodidade, esta função deve ser usada com critério. Imagens públicas devem ser escolhidas com cuidado, preferindo fontes oficiais ou projectos reconhecidos. Porque instalar uma imagem desconhecida no servidor é basicamente convidar um estranho para dentro de casa e entregar-lhe as chaves da cave.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

From each and every container, you can be altered after its creation.

Esta é uma limitação importante do próprio modelo Docker.

O Dockumo permite alterar parâmetros como:

- nome do container;
- limites de memória;
- limites de CPU;
- política de reinício.

Mas configurações como portas publicadas, volumes montados ou imagem base não são normalmente alteradas num container já criado. Para mudar esses elementos, a abordagem correcta é recriar o container com nova configuração.

Isto é importante porque evita falsas expectativas. Um gestor gráfico não deve fingir que consegue fazer aquilo que o motor subjacente não permite de forma limpa. A pior interface é aquela que sorri enquanto prepara uma catástrofe.

8. Segurança: o ponto crítico

Qualquer aplicação com acesso ao Docker daemon deve ser tratada com enorme cuidado. Quem controla o Docker

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

preferencialmente:

- em rede local;
- atrás de VPN;
- com firewall restritiva;
- atrás de proxy autenticado;
- nunca exposto directamente à Internet sem protecção forte.

O socket Docker, normalmente localizado em `/var/run/docker.sock`, dá acesso directo ao Docker Engine. Expor esse socket ou uma aplicação que o controla é uma decisão séria. Em termos práticos, acesso ao Docker pode equivaler a acesso privilegiado ao host.

Por isso, a primeira regra do Dockumo é simples: conforto operacional sim; suicídio administrativo não.

9. Instalação básica

Num sistema Linux com Docker já instalado, a instalação do Dockumo pode ser feita com um ambiente virtual Python:

```
python3 -m venv .venv
source .venv/bin/activate
```

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

Depois, a aplicação pode ser acessada no browser através da porta configurada, por exemplo:

```
http://IP_DO_SERVIDOR:8088
```

Naturalmente, num ambiente mais sério, a aplicação deve ser executada como serviço `systemd`, protegida por firewall e, se necessário, colocada atrás de um reverse proxy com autenticação.

10. Exemplo de serviço `systemd`

Um exemplo simples de unidade `systemd` poderia ser:

```
[Unit]
Description=Dockumo Manager
After=docker.service network.target
Requires=docker.service

[Service]
Type=simple
WorkingDirectory=/opt/dockumo
ExecStart=/opt/dockumo/.venv/bin/python /opt
Restart=always
```

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

```
[install]
```

```
WantedBy=multi-user.target
```

O uso de `User=root` deve ser ponderado. Em alguns sistemas, pode ser substituído por um utilizador pertencente ao grupo `docker`. Mas também aqui há que perceber a consequência: pertencer ao grupo `docker` dá grande poder sobre o sistema. O Linux avisa, mas os humanos insistem em transformar permissões em decoração.

11. Dockumo como ferramenta de aprendizagem e controlo

O Dockumo não é apenas uma ferramenta operacional. Também pode ser uma ferramenta pedagógica.

Ao mostrar visualmente containers, estados, imagens, logs e configurações, ajuda a compreender melhor o ciclo de vida de um container:

- imagem importada;
- container criado;
- container em execução;
- container parado;
- logs analisados;



comandos, pode ficar disperso por comandos sucessivos. Para quem administra sistemas, desenvolve aplicações ou mantém laboratórios próprios, esta clareza é uma vantagem concreta.

12. Possível evolução do projecto

A primeira versão do Dockumo é funcional, mas abre caminho a várias evoluções:

- autenticação por utilizador e palavra-passe;
- perfis de permissões;
- gestão de volumes Docker;
- gestão de redes Docker;
- templates personalizados guardados em ficheiro JSON ou base de dados;
- monitorização de CPU, memória e tráfego por container;
- integração com ntfy para alertas;
- backup e exportação de configurações;
- modo escuro completo no estilo dos restantes painéis internos;
- integração futura com o ecossistema Virtumo/Voltaire.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

serviços internos. Uma espécie de centro de comando doméstico-profissional, mas sem a parte ridícula dos dashboards corporativos que têm gráficos a piscar e ninguém sabe para quê.

13. Conclusão

O Dockumo é uma pequena ferramenta, mas aponta para uma ideia maior: recuperar controlo sobre a infra-estrutura.

Num tempo em que tudo parece empurrar os utilizadores para plataformas fechadas, subscrições, serviços opacos e dependências externas, construir ferramentas próprias continua a ser um acto de liberdade técnica. Não por romantismo ingénuo, mas por lucidez.

Gerir containers com uma interface própria, escrita em Python, compreensível e adaptável, é mais do que conforto. É também soberania operacional. É saber o que corre, onde corre, como corre e quando deve parar.

O Dockumo não pretende reinventar o Docker. Pretende apenas dar-lhe uma camada de comando mais clara, mais próxima do utilizador técnico e mais alinhada com uma filosofia simples: compreender antes de automatizar, controlar antes de delegar, construir antes de depender.

Blogue Fragmentos do Caos



A verdade nasce onde o pensamento é livre.

- Docker Engine SDKs — documentação Docker
- Docker — proteger o acesso ao daemon
- OWASP Docker Security Cheat Sheet
- NiceGUI — documentação oficial


Autor: Francisco Gonçalves


Co-autoria técnica e editorial: Augustus Veritas / OpenAI.

Fragmentos do Caos — tecnologia, pensamento crítico e pequenas barricadas contra a mediocridade digital.

 [GitHub Pages](#)

 [CodeBerg Pages](#)

 [Fragmentos do Caos:](#) [Blogue](#) • [Ebooks](#) • [Carrossel](#)

 Esta página foi visitada ... vezes.

[Contactos](#)