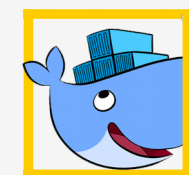


Começando com
Docker



<Diogo Alves>



- Arquiteto de Soluções e
Líder na iniciativa DevOps
da empresa Cedro



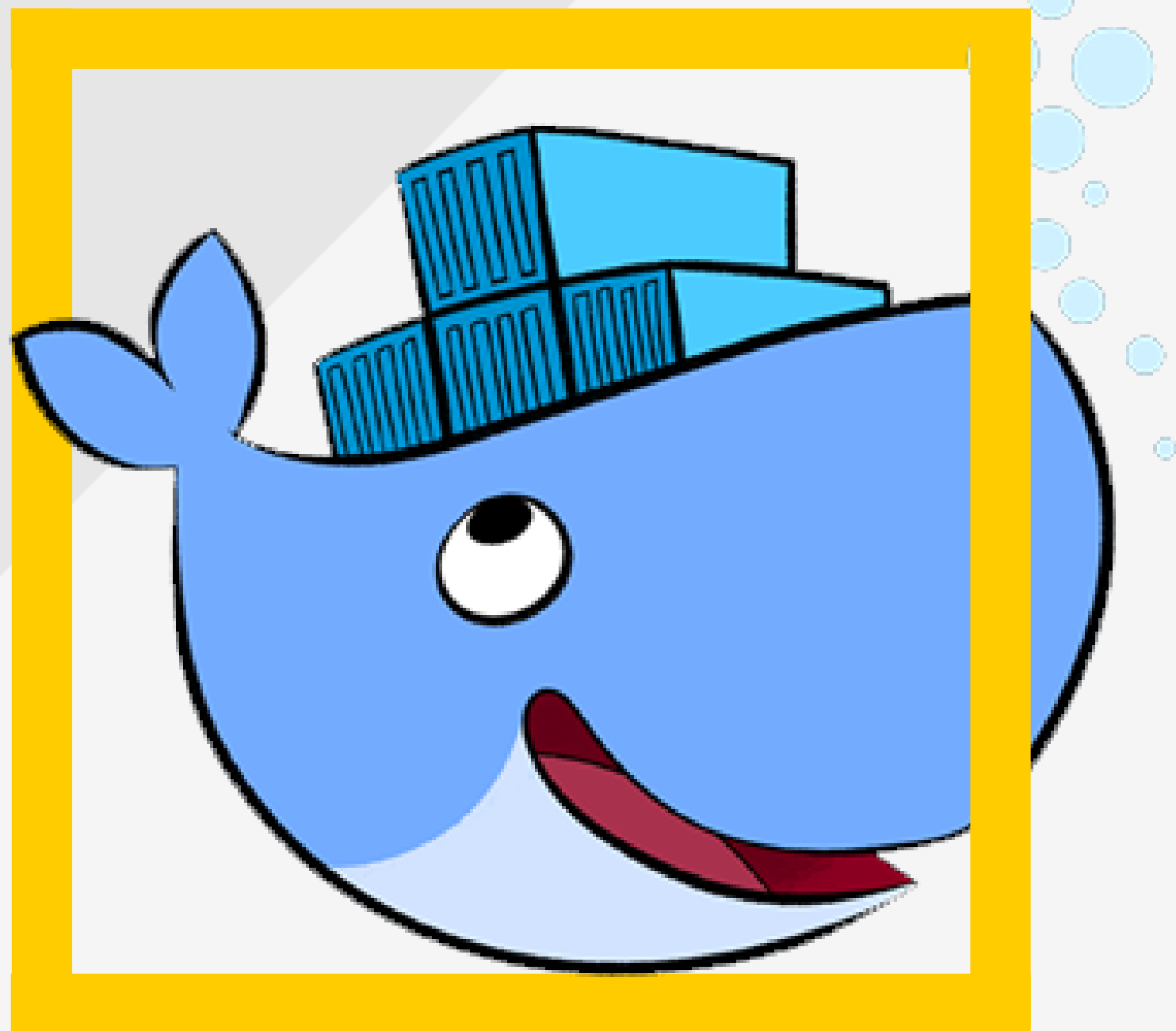
Formação:
- Sistemas de Informação



Pouco mais de 12 anos como:
- SysAdmin apaixonado por Linux
- Desenvolvedor
- Instrutor
- Entusiasta de Segurança da
Informação



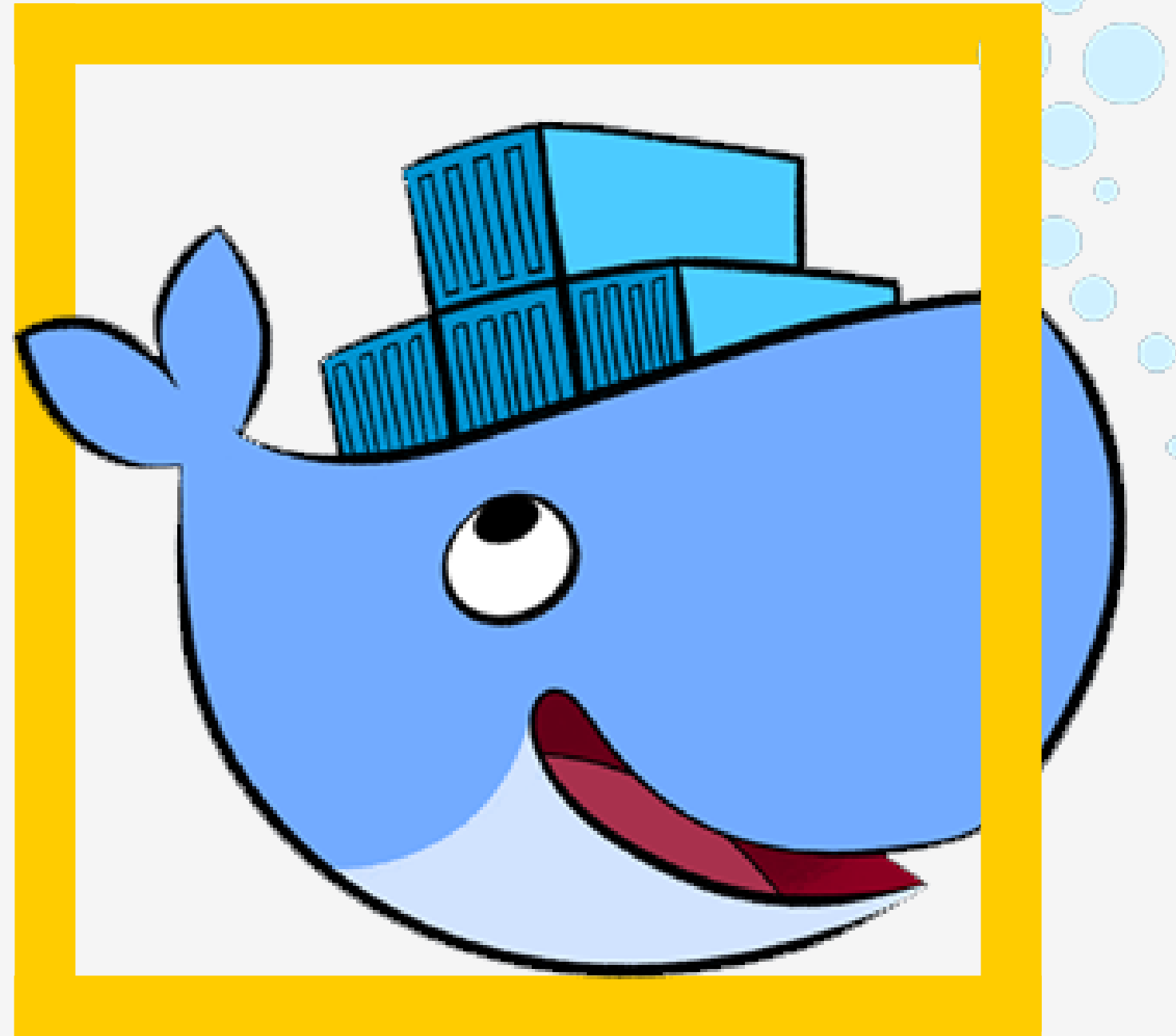
Algumas Certificações:
- CEH – Ethical Hacking
- ISO 27001
- Asterisk Administrator
- AWS



Agenda

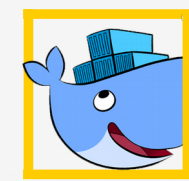
- Cluster Docker
- Configurando um cluster para Docker
- O que é services
- O que é stack
- Criando a primeira Stack Hand-ons

Começando com
Docker



Cluster
Docker

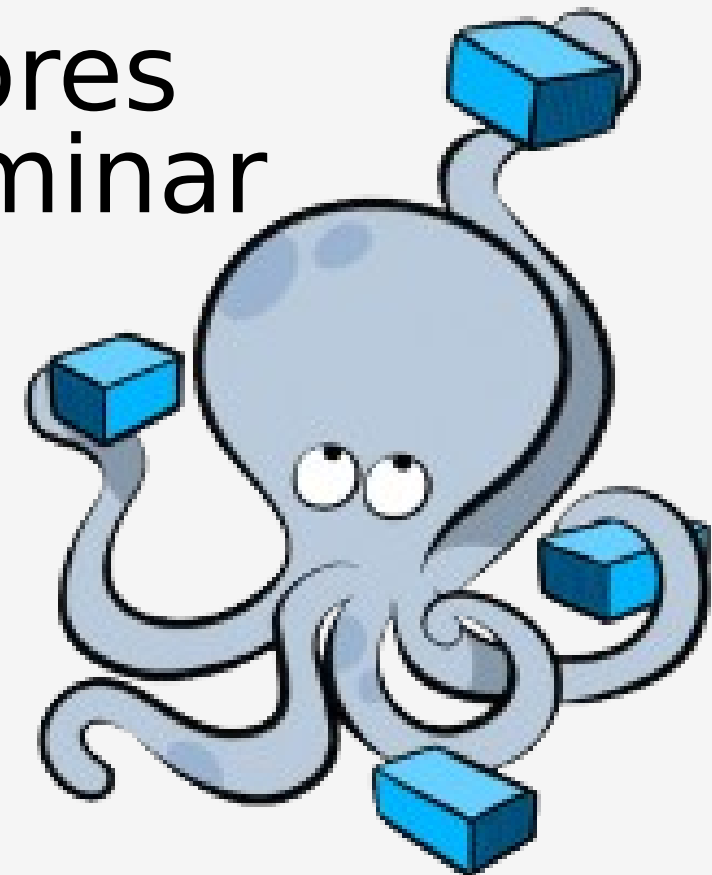
<Cluster Docker>



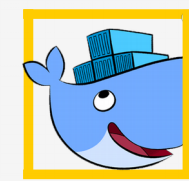
Começando com Docker

O que é um cluster?

Cluster é um termo em inglês que significa “aglomerar” ou “aglomeração” e pode ser aplicado em vários contextos. No caso da computação, o termo define uma arquitetura de sistema capaz combinar vários computadores para trabalharem em conjunto ou pode denominar o grupo em si de computadores combinados.



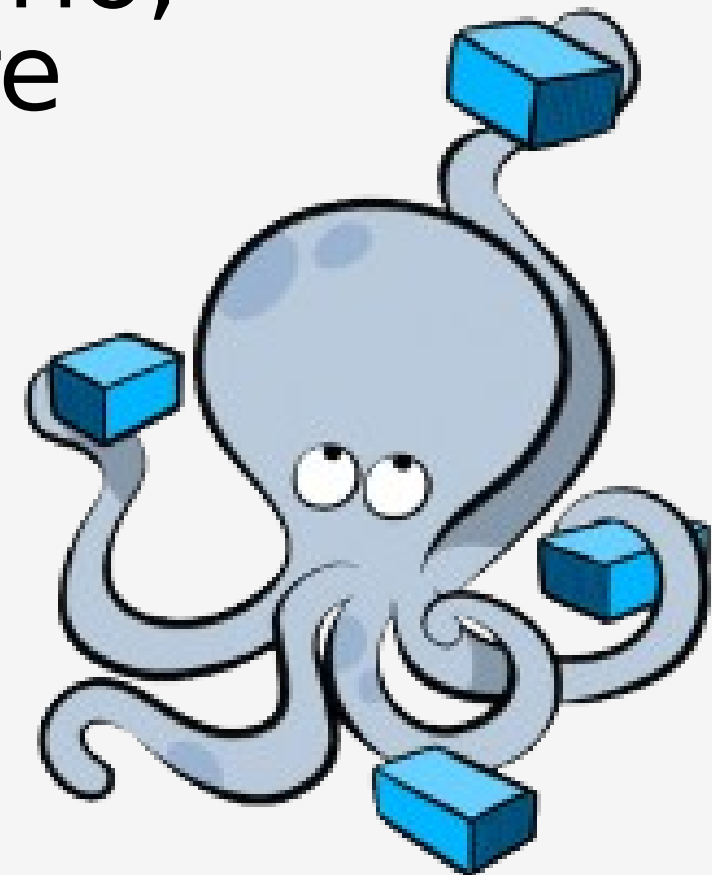
<Cluster Docker>



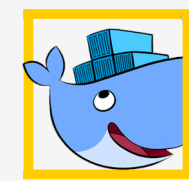
Começando com Docker

Característica de um cluster:

Cada computador é denominada “nodo” e, combinadas, formam o cluster. Em alguns casos, é possível ver referências como “supercomputadores” ou “computação em cluster” para o mesmo cenário, representando o hardware usado ou o software especialmente desenvolvido para conseguir combinar esses equipamentos.



<Cluster Docker>



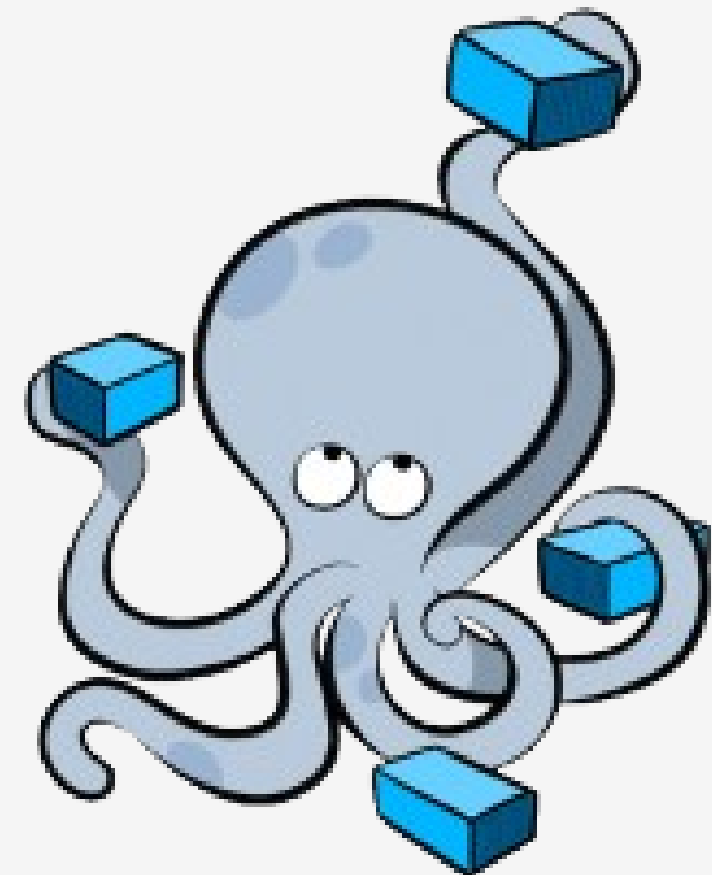
Começando com Docker

Questão:

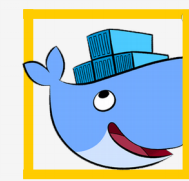
Como escalar e orquestrar containers? Essa

É a primeira pergunta que devemos nos fazer

Antes.



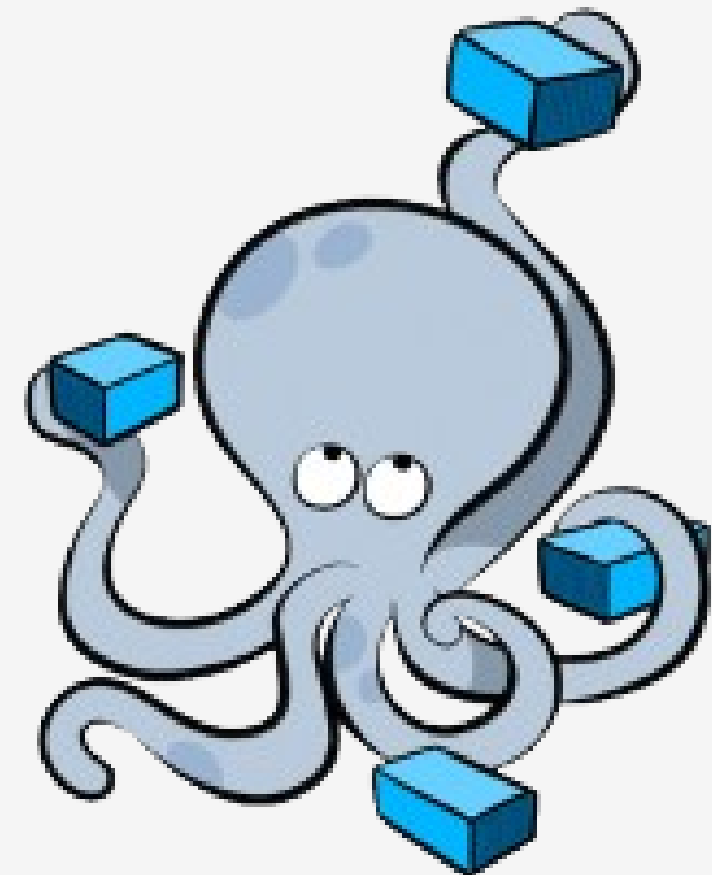
<Cluster Docker>



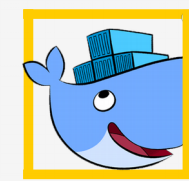
Começando com Docker

A Motivação:

Usar o docker run e orquestrar serviços usando Compose ajuda muito e torna o processo de trabalhar com containers bastante fácil. Porém, todos os cenários que foram abordados rodam local “standalone”, ou seja, na nossa máquina ou em apenas em um servidor.



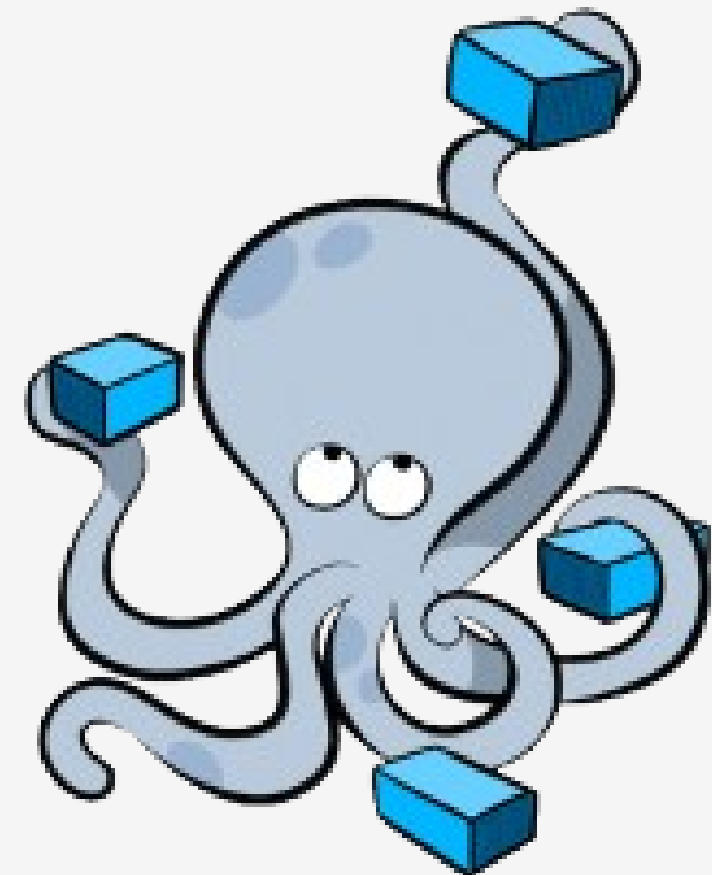
<Cluster Docker>



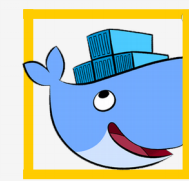
Começando com Docker

O Problema:

Quando pensamos em escalar os serviços horizontalmente entre várias máquinas tínhamos problemas (porque, lembre-se, nossos containers são isolados e um container não enxerga o processo do outro) e os hosts também precisam se falar por meio da rede. Até 2015 a Docker não tinha uma solução para esse problema.



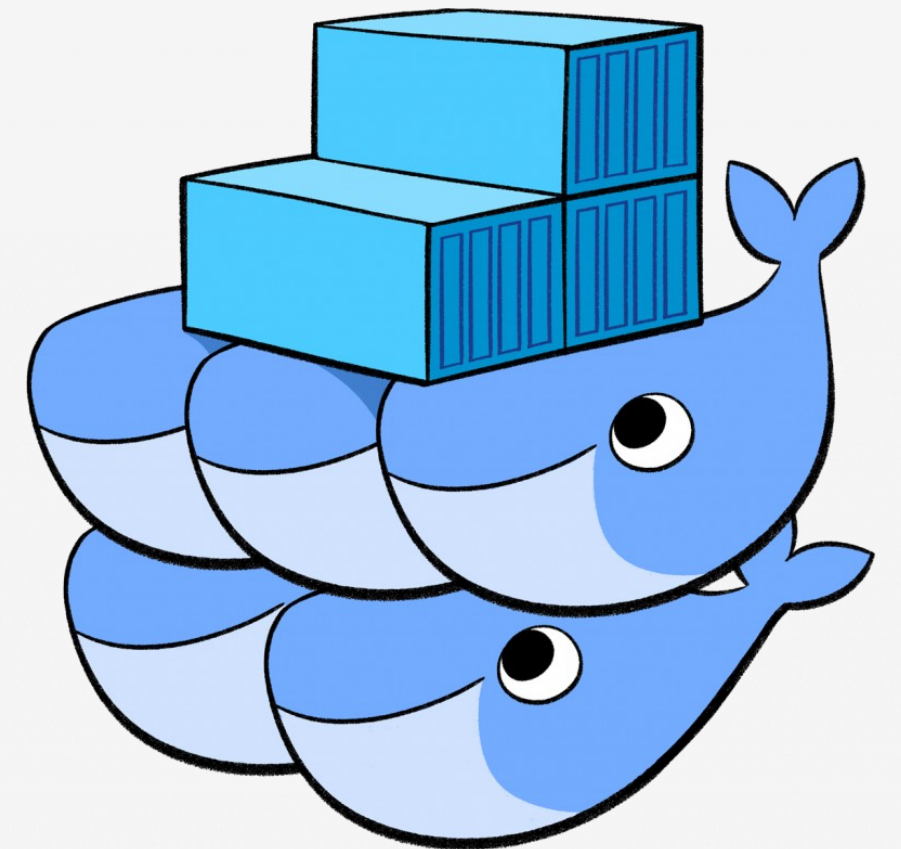
<Cluster Docker>



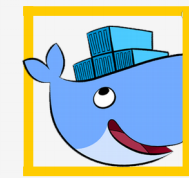
Começando com Docker

Hello, Swarm!

A ferramenta para cluster da Docker é a Swarm, que já existia como uma ferramenta à parte, mas a partir da versão 1.12 passou a ser nativa, ou seja, só atualizar o seu Docker e habilitar o swarm mode se você quiser usar.



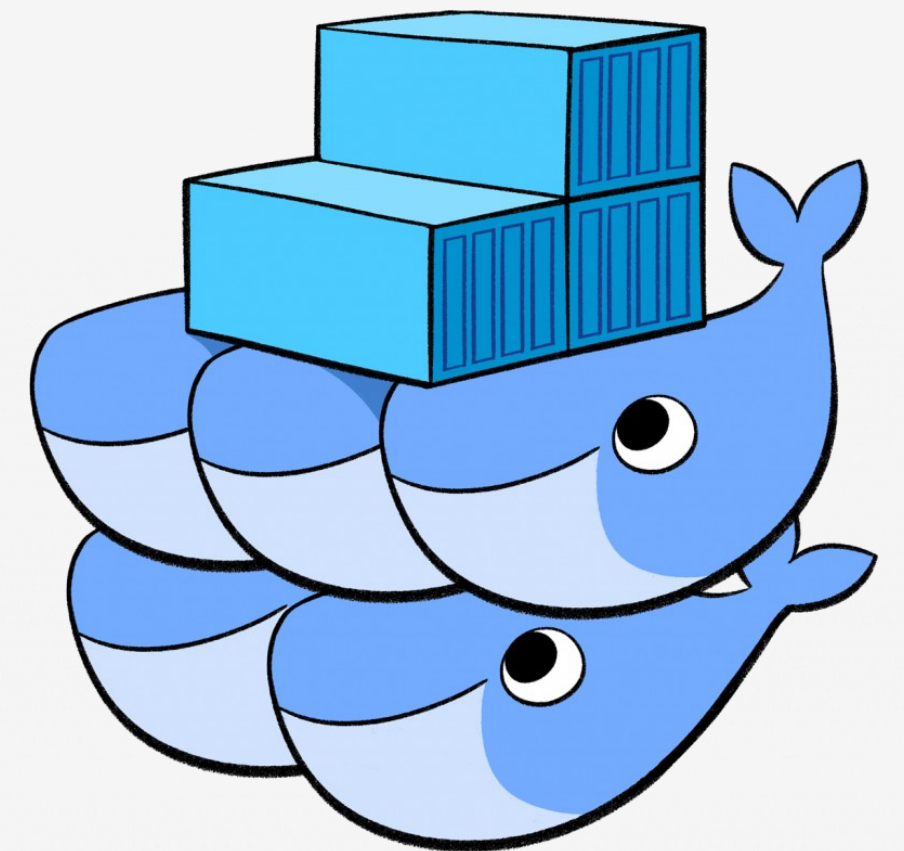
<Cluster Docker>

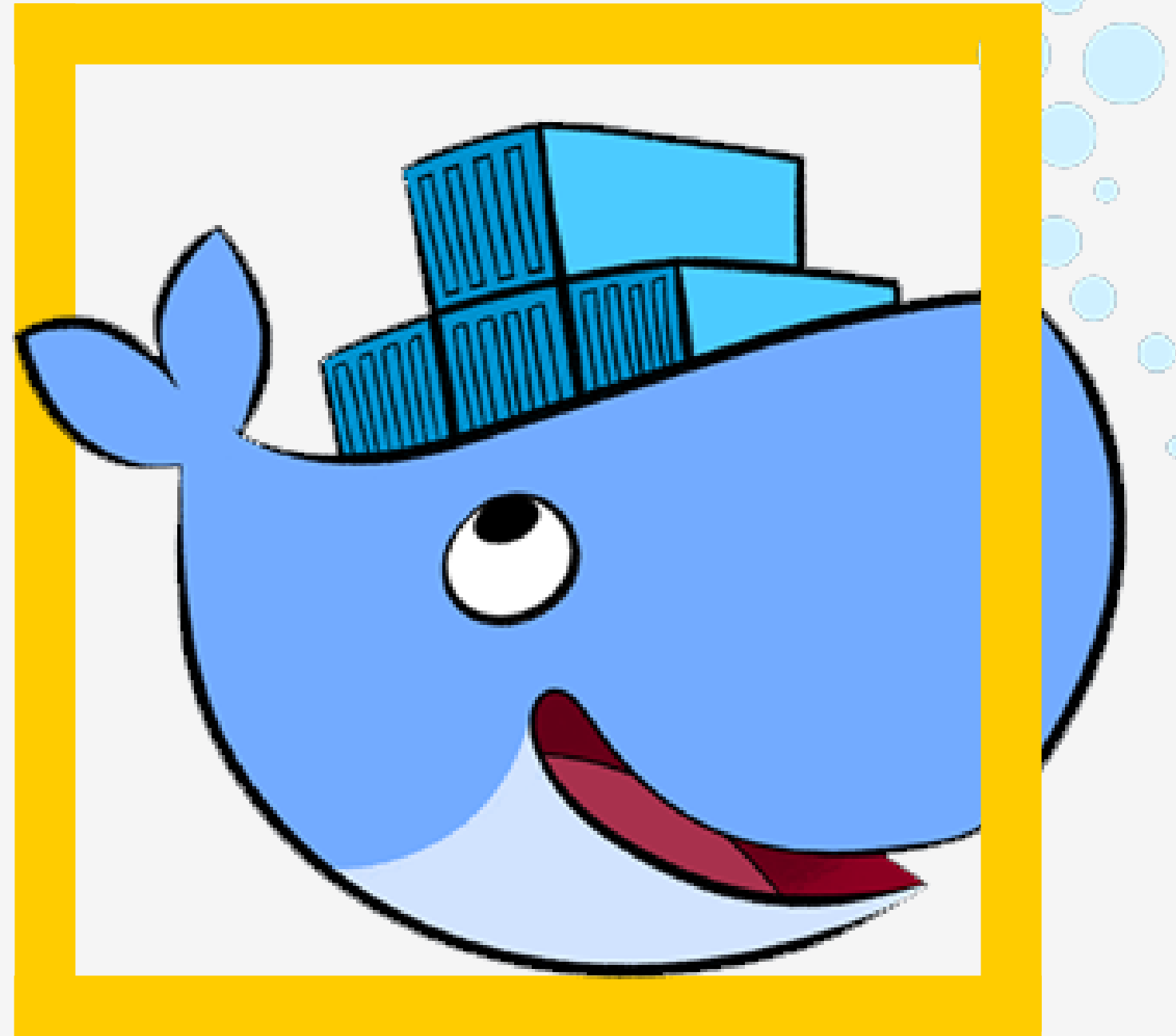


Começando com Docker

O que o Swarm resolve?

A partir do nosso master podemos adicionar hosts Docker, ou seja, máquinas rodando Docker para o nosso cluster. Isso facilita no deploy e na gerência dos serviços e tira a responsabilidade do Sysadmin sobre onde vai fazer o deploy dos Containers. O Swarm abstrai essa parte, tomando para ele a responsabilidade de escolher os locais onde alocação os containers nos nós do cluster.





Configurando um cluster para
Docker

<Configurando um cluster para Docker>



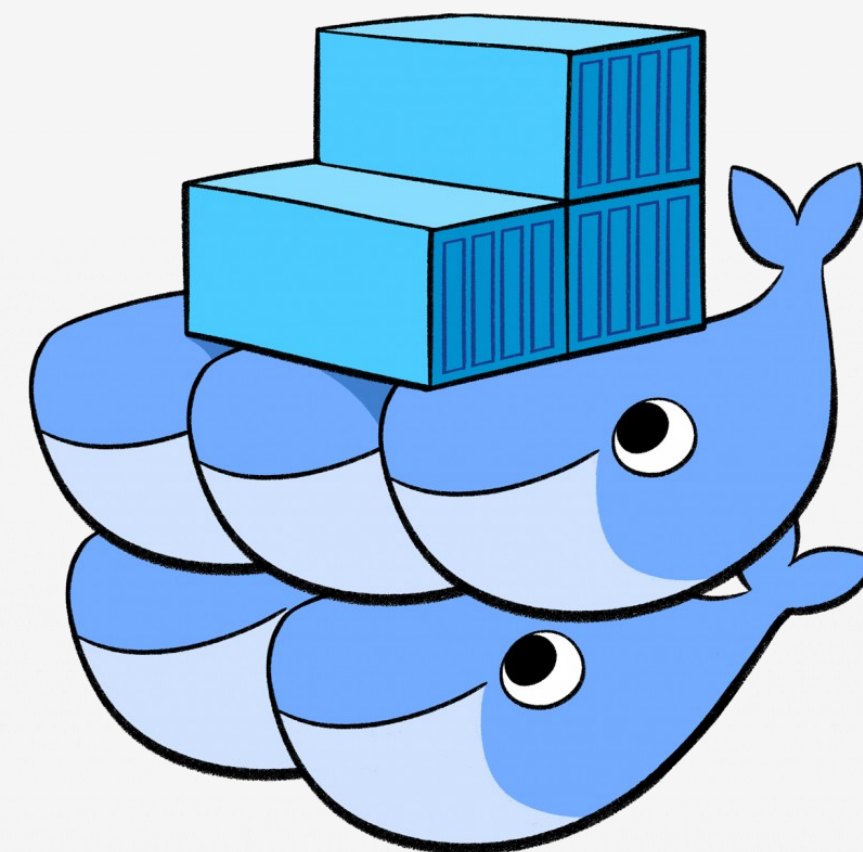
Começando com Docker

Acessar PWD

<https://labs.play-with-docker.com/>

Existe duas maneiras de se criar um Cluster Docker no PWD:

- Utilizando o template da plataforma;
- Configurando manualmente o Cluster;

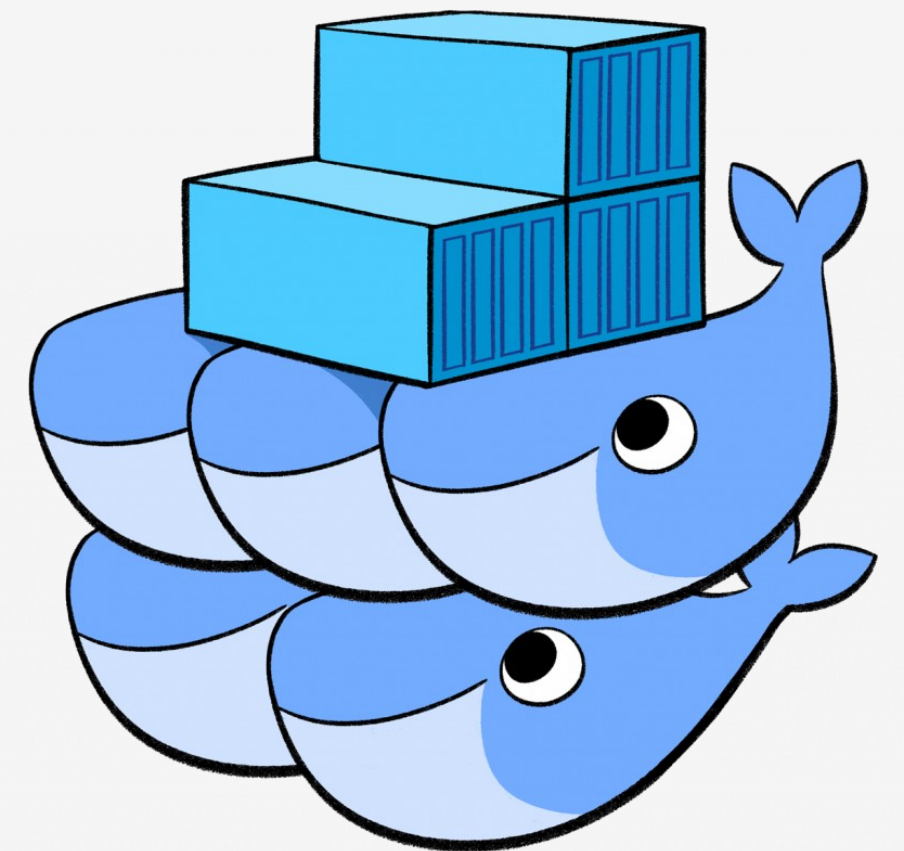
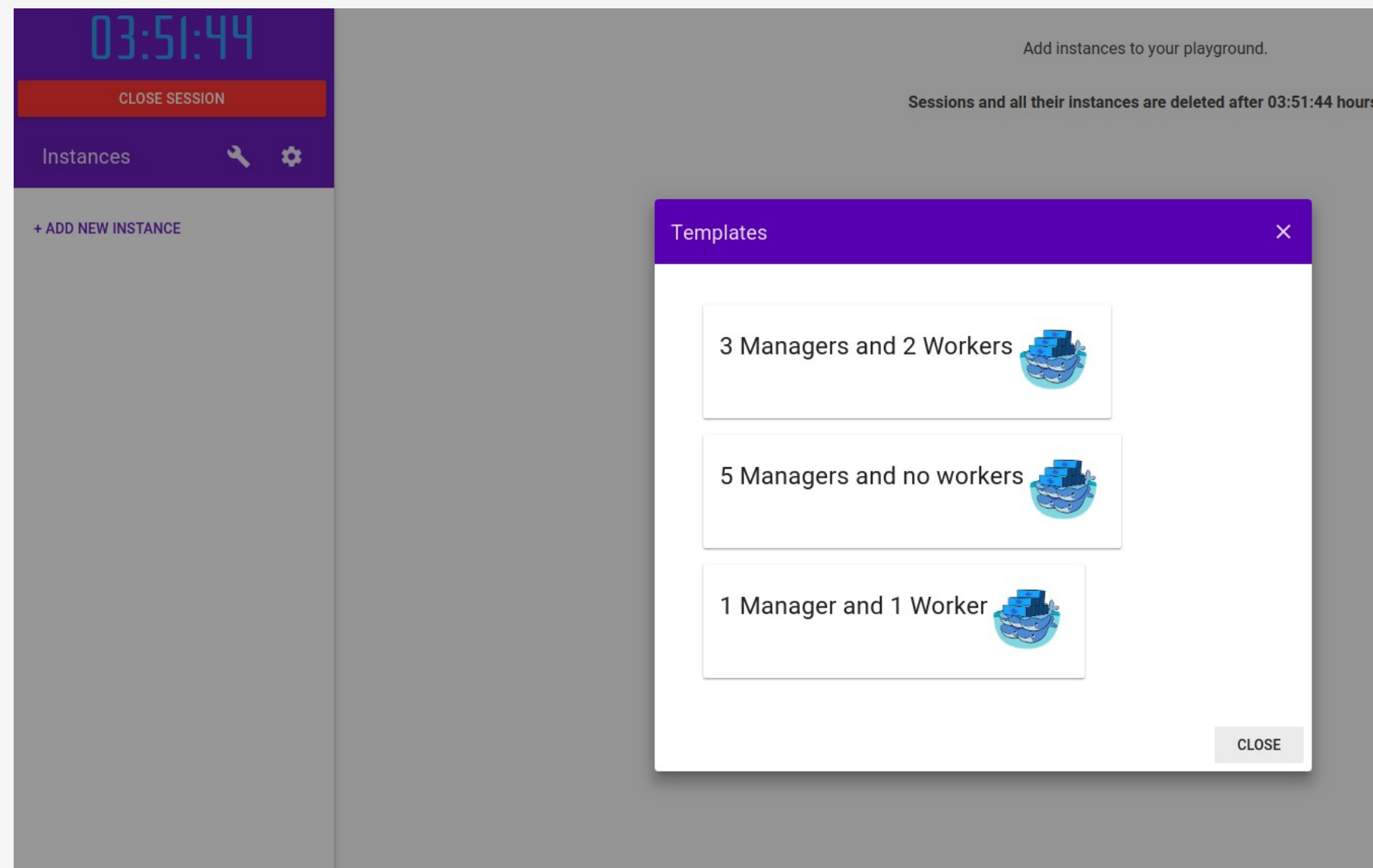


<Configurando um cluster para Docker>



Começando com Docker

Utilizando templates do PWD



<Configurando um cluster para Docker>



Começando com Docker

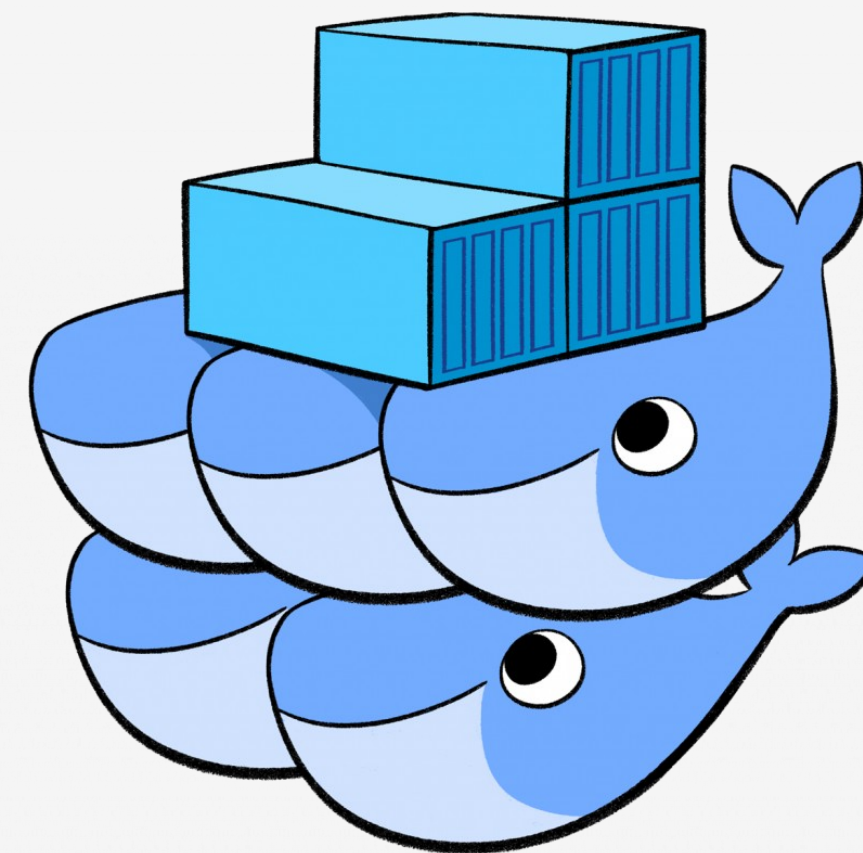
> Configurando Manualmente

Adicione instancias no PWD, nosso exemplo 3 e após:

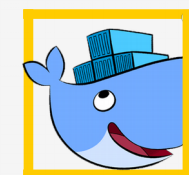
```
$ docker swarm init --advertise-addr SEU-IP
```

Copie a chave parecida com a abaixo:

```
$ docker swarm join --token SWMTKN-1-  
3hl3chvop1dt12oz35acync7ig7s0mzck23zu8  
jutre7ltqhne-34rr4vbcr5fa41g5tt59am9lg  
192.168.0.13:2377
```



<Configurando um cluster para Docker>



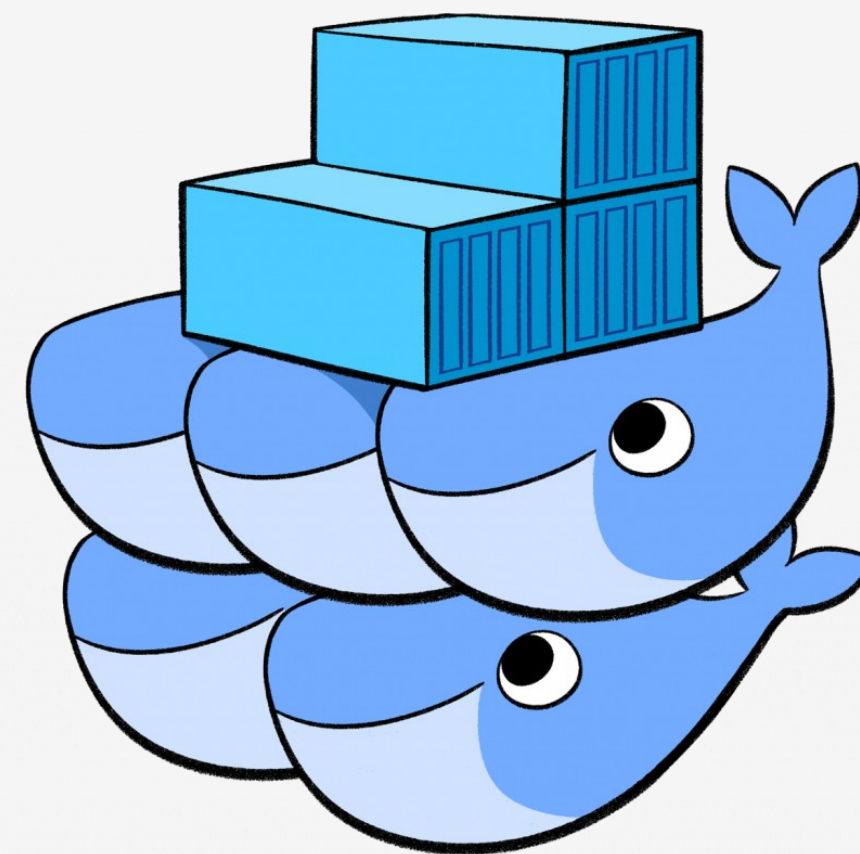
Começando com Docker

> Verificando chave para adicionar um worker:

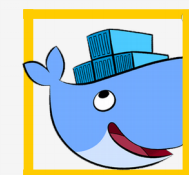
```
$ docker swarm join-token worker
```

> Verificando chave para adicionar um manager:

```
$ docker swarm join-token manager
```



<Configurando um cluster para Docker>



Começando com Docker

> Verificando os nodos:

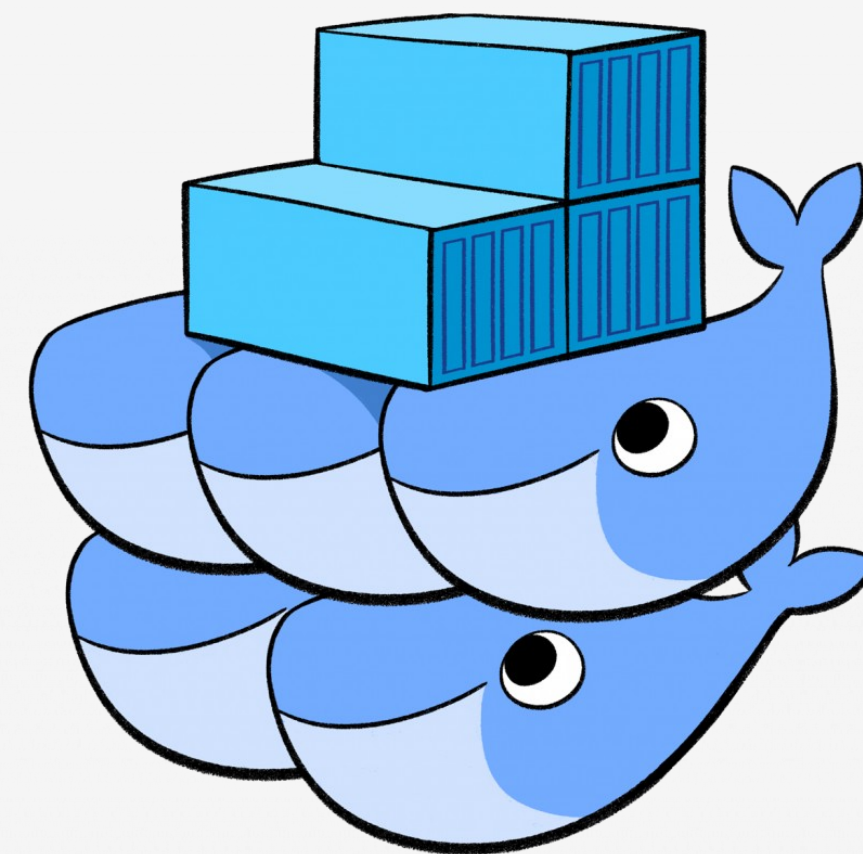
```
$ docker node ls
```

> Promovendo um nodo:

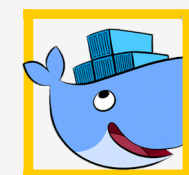
```
$ docker node promote id
```

> Des-promovendo um nodo:

```
$ docker node demote id
```



<Configurando um cluster para Docker>



Começando com Docker

> Inspecionando um nodo:

```
$ docker node inspect self --pretty
```

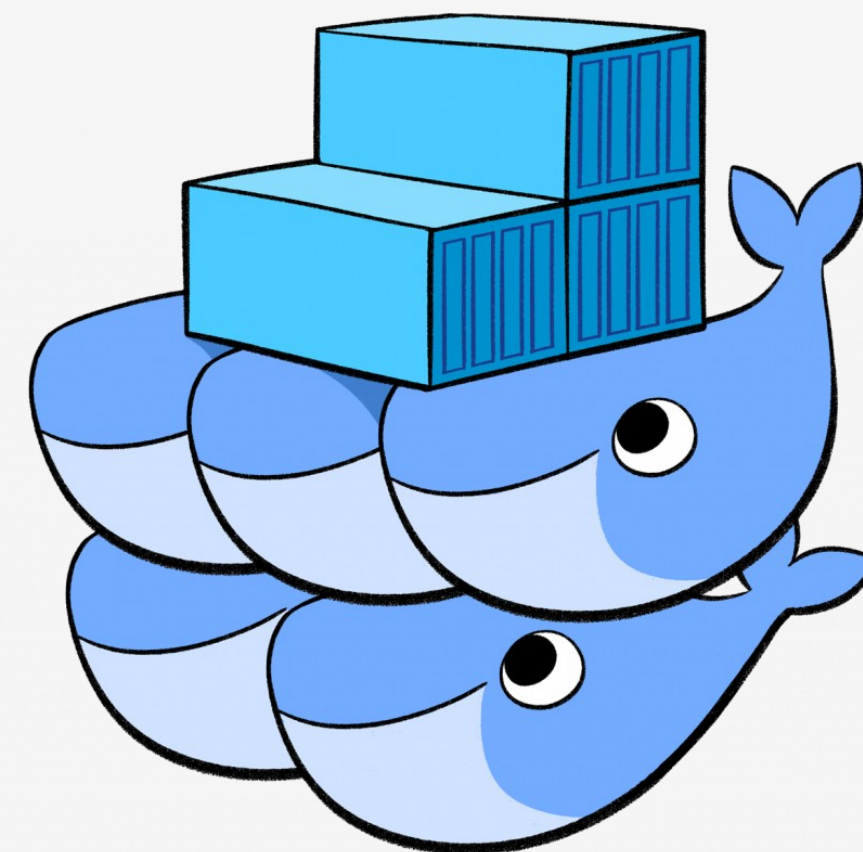
```
$ docker node inspect id
```

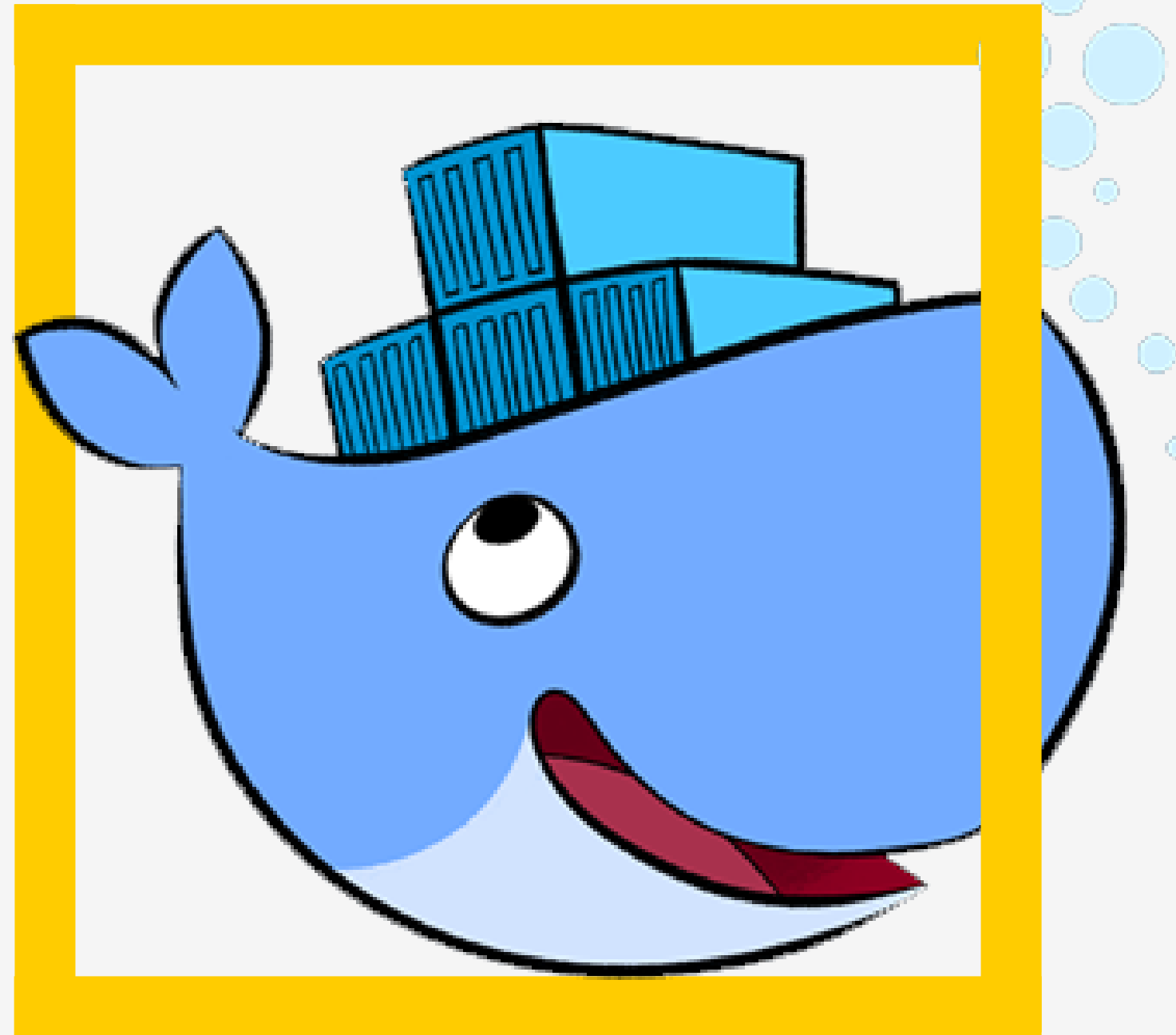
> Desconectando de um nodo do cluster:

```
$ docker swarm leave
```

> Removendo um nodo:

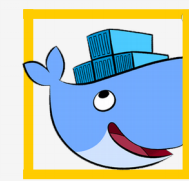
```
$ docker node rm id
```





O que é
Docker-Service

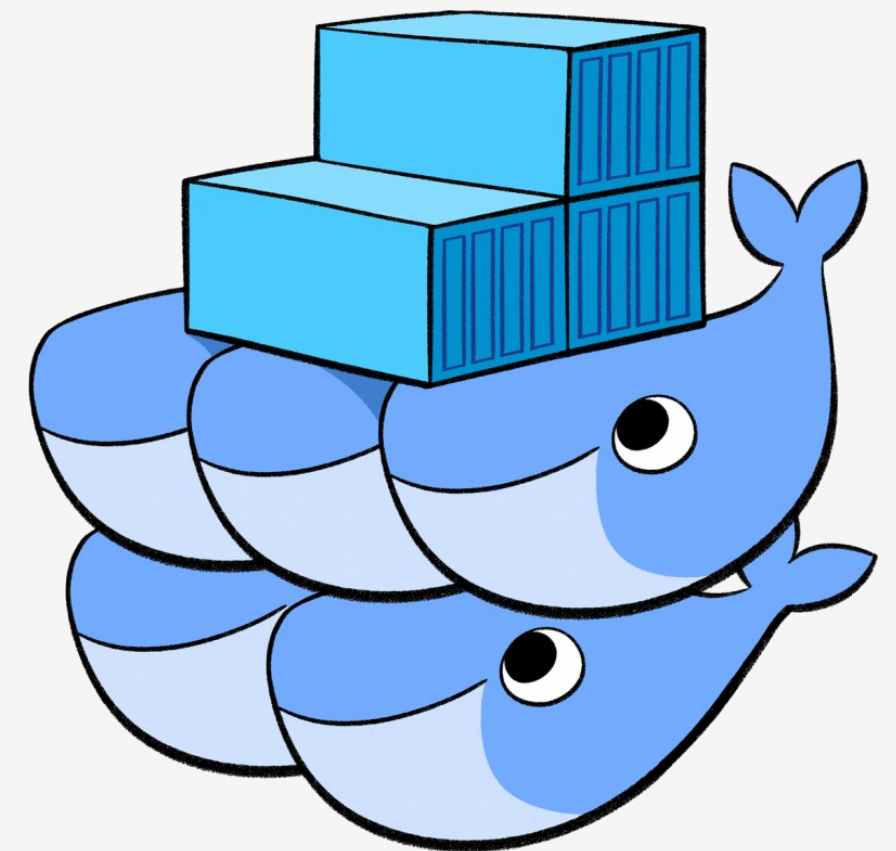
<O que é Docker Service>



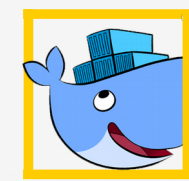
Começando com Docker

O que é Docker Service:

Docker Service é uma feature que foi incorporada pela engine Docker que permite ao administrador criar e administrar sua stack de serviço dentro de um cluster Swarm, sem precisar utilizar uma segunda ferramenta para isso.



<O que é Docker Service>

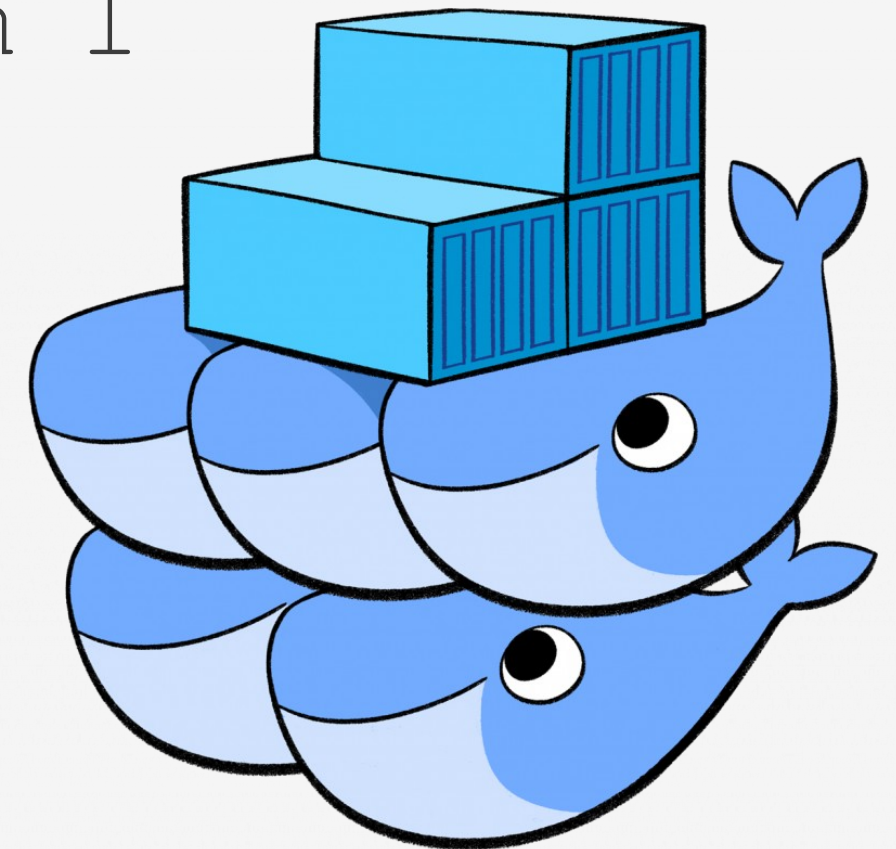


Começando com Docker

> Primeiro Docker Service

```
$ docker pull nginx
```

```
$ docker service create --replicas 1  
--update-delay 10s --update-parallelism 1  
--name nginx nginx
```



<O que é Docker Service>



Começando com Docker

> No terminal execute

```
$ docker service ls
```

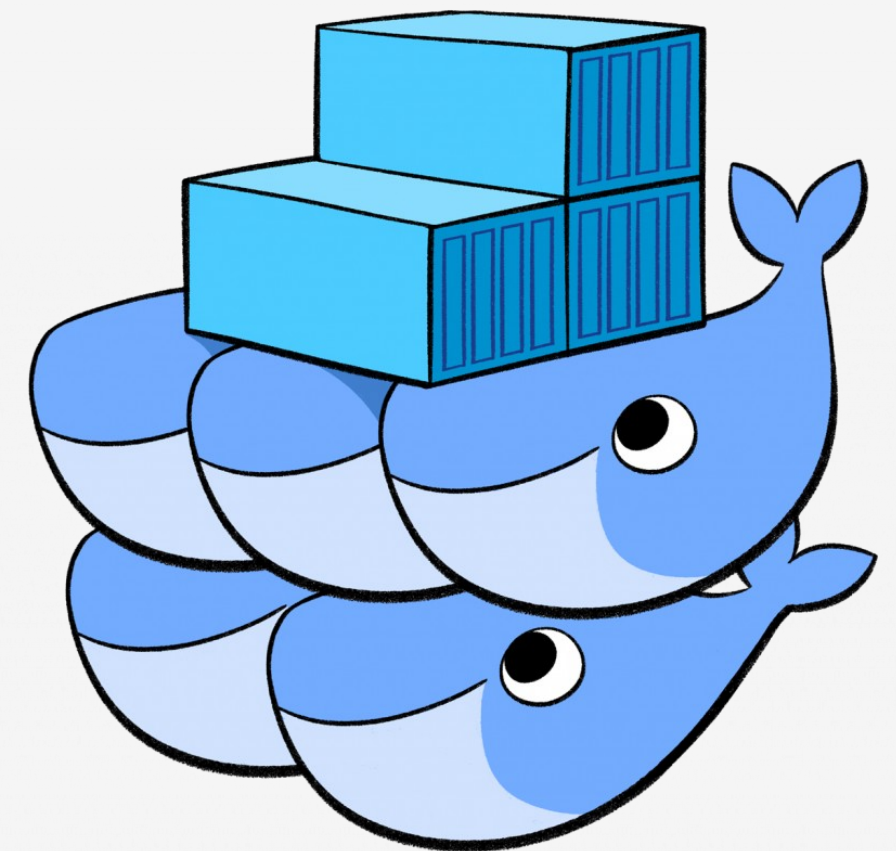
```
$ docker service create
```

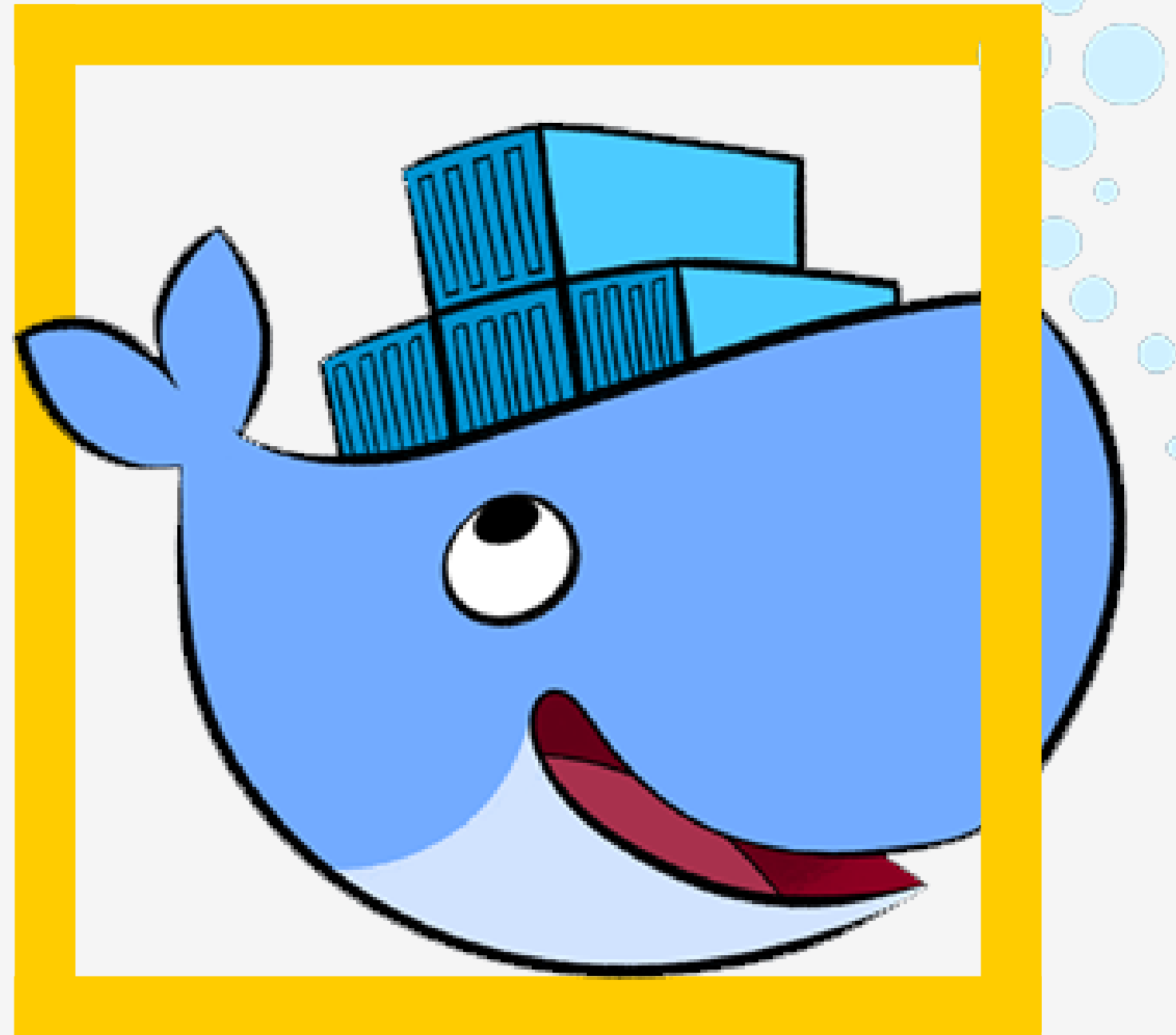
```
$ docker service scale name=x
```

```
$ docker service inspect
```

```
$ docker service ps
```

```
$ docker service rm
```





O que é
Docker Stack

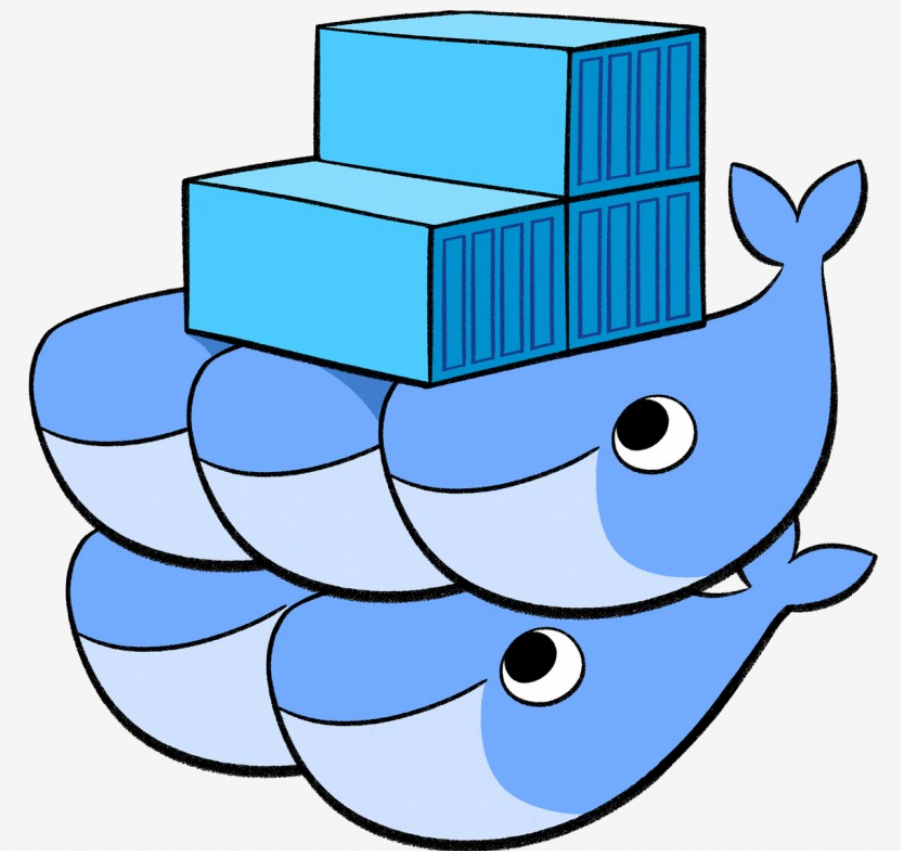
<O que é Docker Stack>

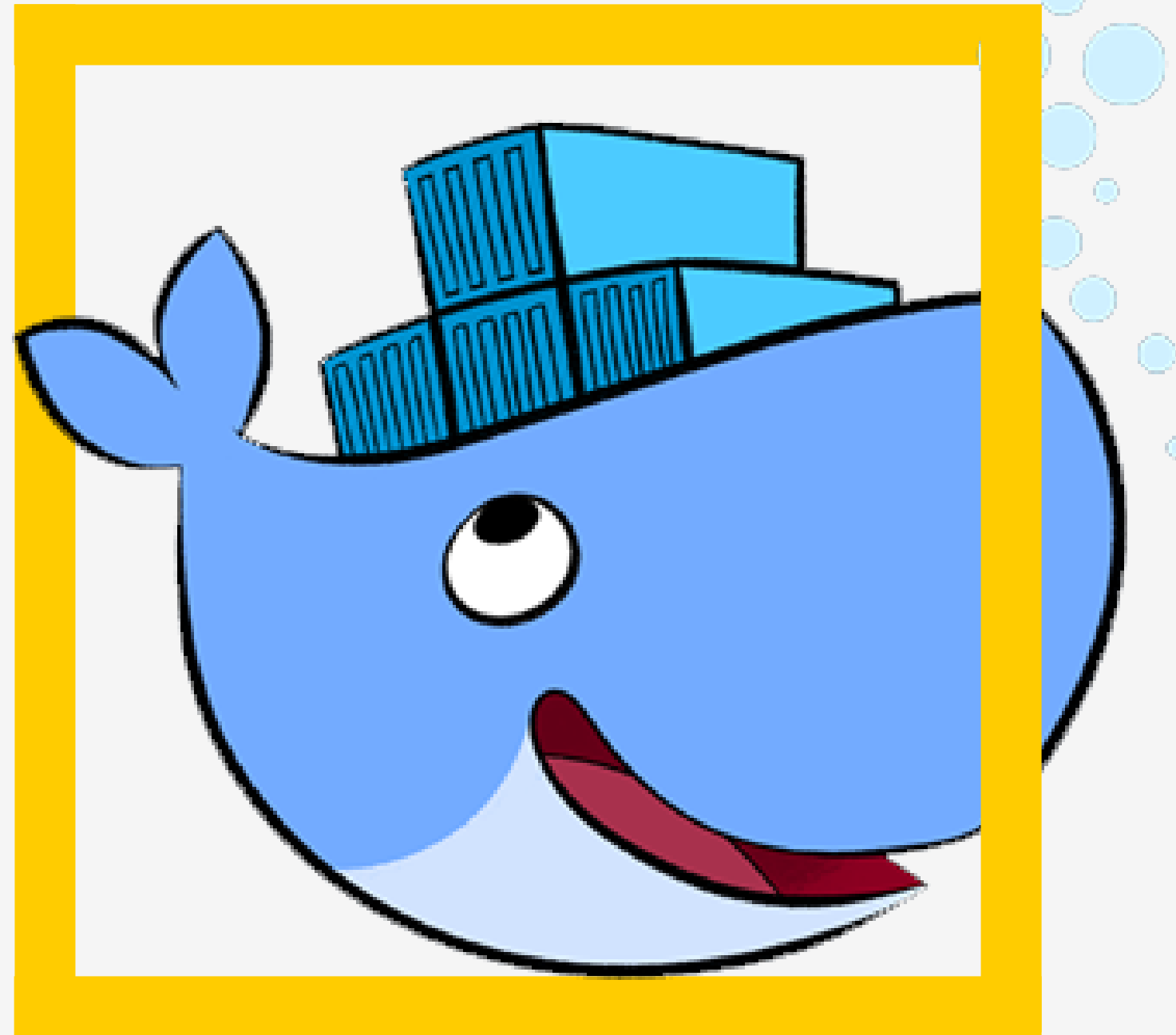


Começando com Docker

A Motivação:

Com Docker Stack é possível você portar suas aplicações do compose para o Swarm, e isso graças a funcionalidade de deploy disponível na engine. Seu funcionamento é bem simples, basta você informar na execução do comando o diretório de onde está o seu arquivo compose, e o nome da aplicação





Criando a primeira
Docker Stack

<Criando a primeira Docker Stack>



Começando com Docker

> No terminal execute

```
$ git clone
```

```
https://github.com/diogoab/curso_docker.git
```

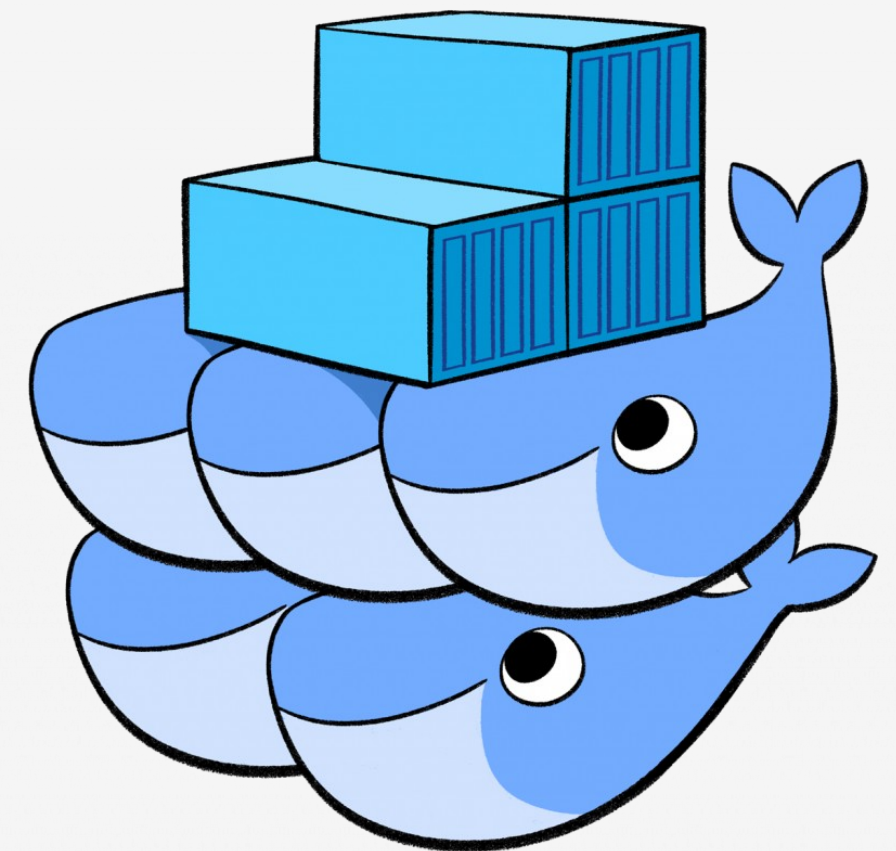
```
$ cd curso_docker/Aula 04/stackdemo
```

```
$ docker build . -t
```

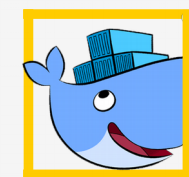
```
127.0.0.1:5000/stackdemo
```

```
$ docker stack deploy --compose-file
```

```
docker-compose.yml stackdemo
```



<Criando a primeira Docker Stack>



Começando com Docker

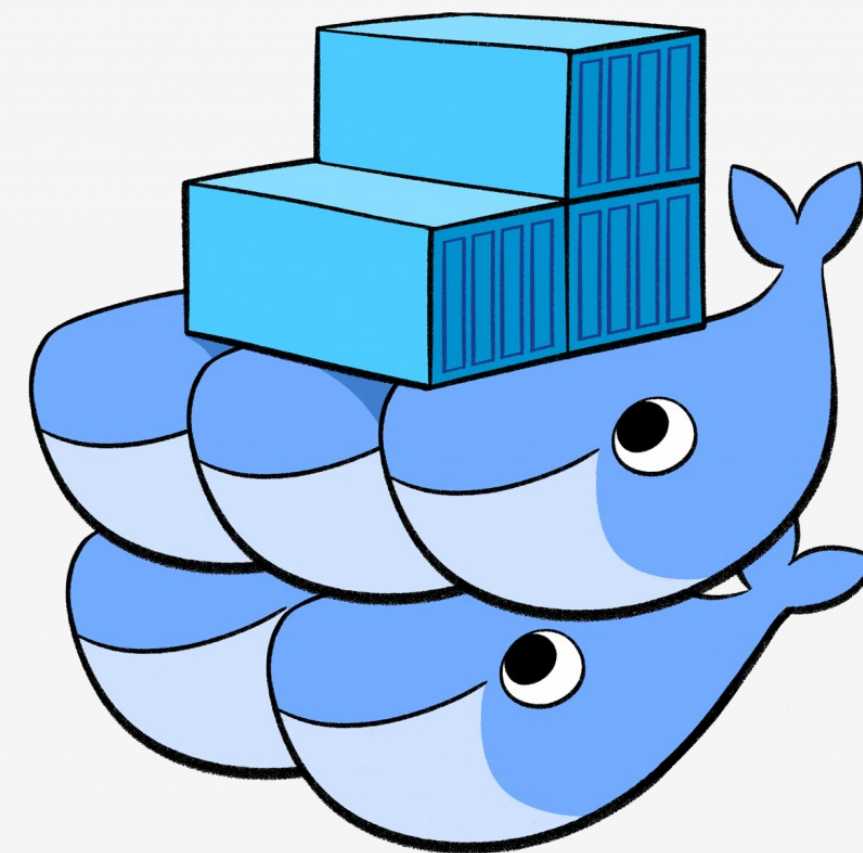
> No terminal execute

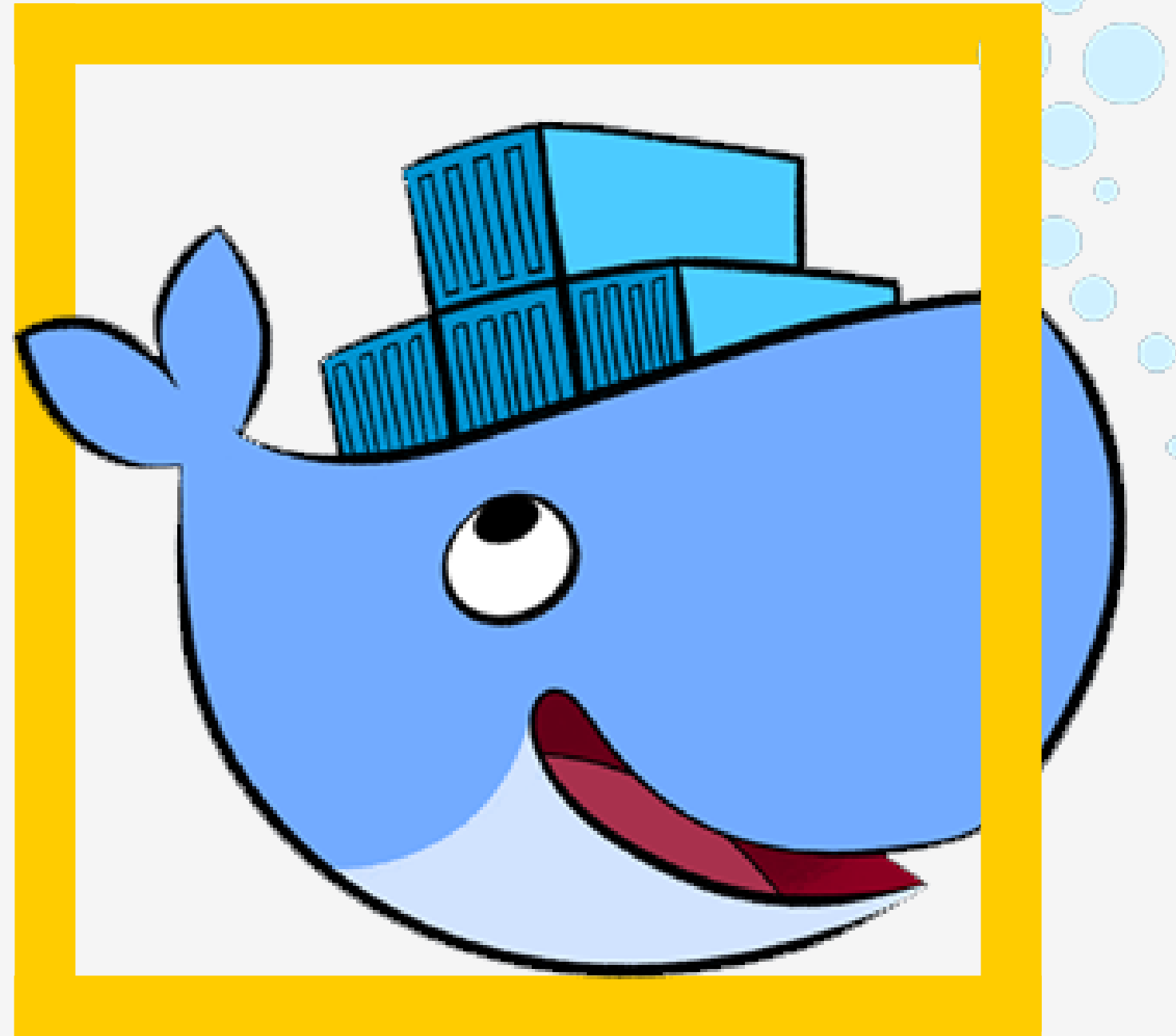
```
$ docker stack ls
```

```
$ docker stack ps
```

```
$ docker stack service ls
```

```
$ docker stack rm
```





Hand-ons



<Hand-ons>

>Vamos lá?

\$ Criar um cluster com 3 manager e 2 workers com serviços web, banco e cache instalados e escalado no cluster.

\$ Sugestão utilizar aplicações como Redis, Nginx, MYSQL, APACHE e etc...



Obrigado



<https://www.linkedin.com/in/diogoalvesbarbosa/>
<https://www.instagram.com/diogo.alvesoficial/>
diogo.alves.barbosa@gmail.com