



Serviço de Contêineres

Cloud Container Engine

Center of Electrical Engineering and Informatics
Federal University of Campina Grande

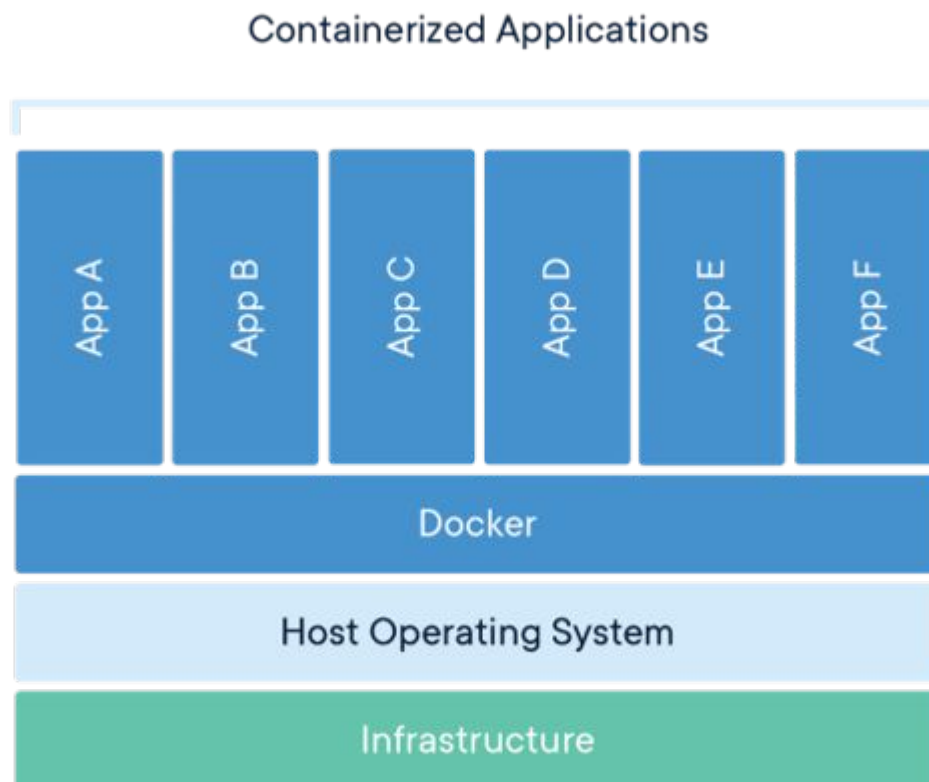


- Introdução ao Docker
 - O que é?
 - Qual a vantagem?
 - Conceitos (contêiner, orquestrador, repositório)
 - Como usar?
- Introdução ao Docker HUB
- Cloud Container Engine
 - O que é?
 - Como usar?

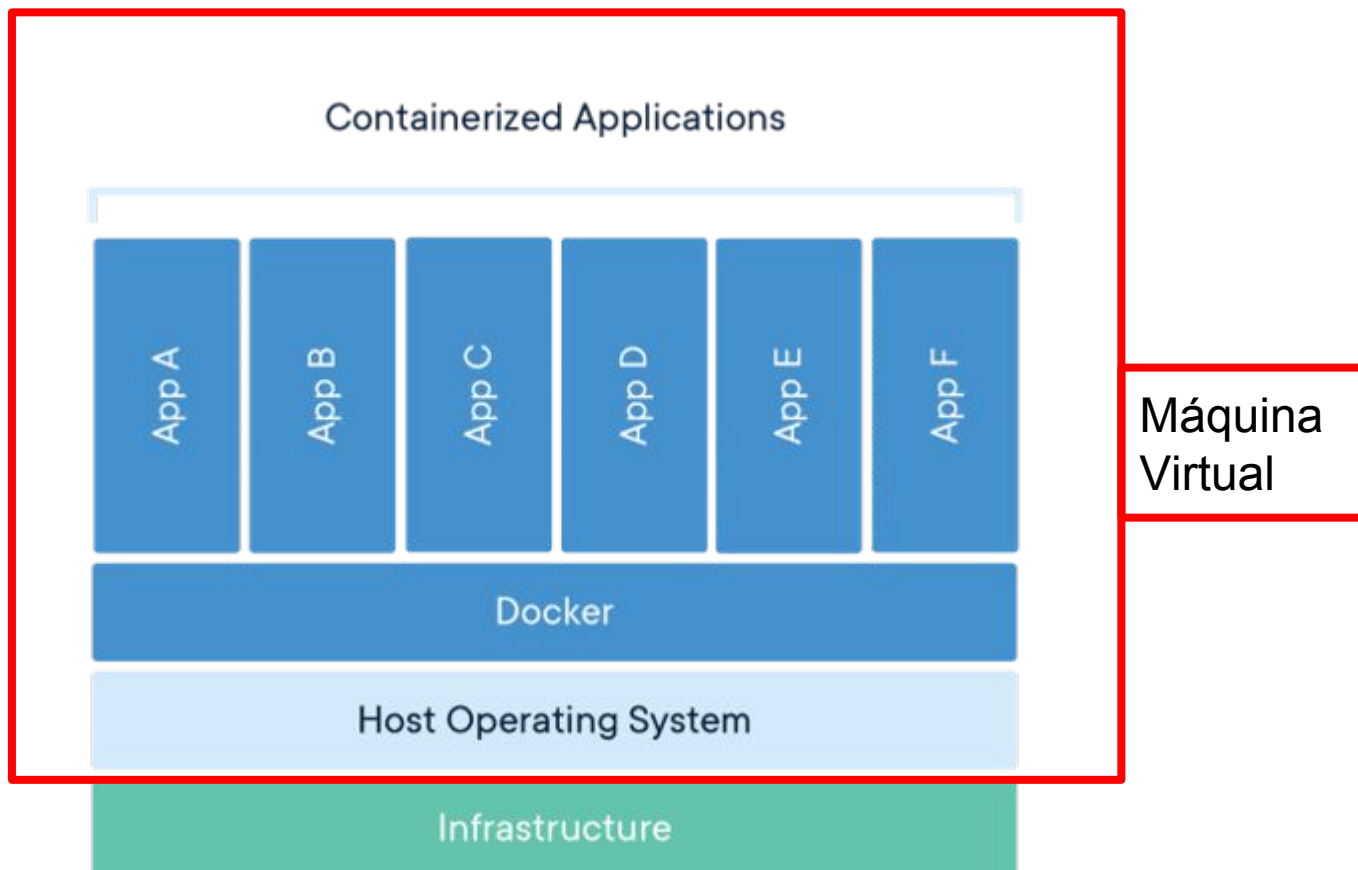
- Introdução ao Docker
 - O que é?
 - Qual a vantagem?
 - Conceitos (contêiner, orquestrador, repositório)
 - Como usar?
- Introdução ao Docker HUB
- Cloud Container Engine
 - O que é?
 - Como usar?

- Docker é uma tecnologia que fornece contêineres de aplicação ao desenvolvedor
- Contêiner é uma abstração do ambiente de execução de uma aplicação e da aplicação
- Contêineres **não** são Máquinas Virtuais
- Contêineres são extremamente leves
- Contêineres permitem independência entre o ambiente de desenvolvimento e ambiente de publicação
- Contêineres Docker executam sobre um *daemon* (**dockerd**) extremamente leve e com uma API Rest bem definida
- O contêiner é definido por um único arquivo (**Dockerfile**)

Docker - O que é?



Docker - O que é?



- **Introdução ao Docker**
 - O que é?
 - **Qual a vantagem?**
 - Conceitos (contêiner, orquestrador, repositório)
 - Como usar?
- **Introdução ao Docker HUB**
- **Cloud Container Engine**
 - O que é?
 - Como usar?

- Ambiente de desenvolvimento auto contido. Permitindo ao desenvolvedor começar a trabalhar instalando somente o **Docker**.
- Divisão clara entre os papéis de desenvolvedor e gerente de deploy. (Como desenvolvedor, você não quer se preocupar com as configurações da máquina e com atualizações de *softwares*)
- Reutilização de imagens prontas. **Cuidado!**

- **Introdução ao Docker**
 - O que é?
 - Qual a vantagem?
 - **Conceitos (contêiner, orquestrador, repositório)**
 - Como usar?
- **Introdução ao Docker HUB**
- **Cloud Container Engine**
 - O que é?
 - Como usar?

- Contêiner

- Pacote com a definição das bibliotecas e *frameworks* utilizados pela aplicação, assim como a aplicação.
- É de responsabilidade do desenvolvedor.
- O acesso ao “mundo externo” é definido através de variáveis de ambiente e volumes montados

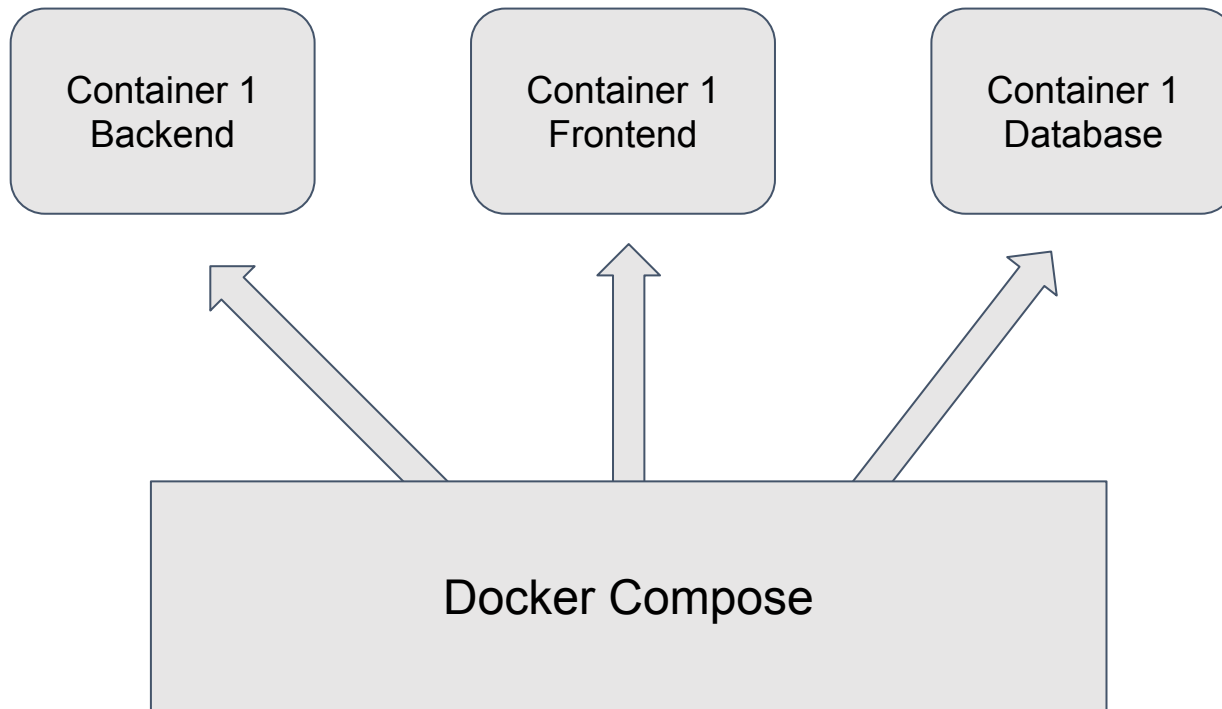
- Orquestrador

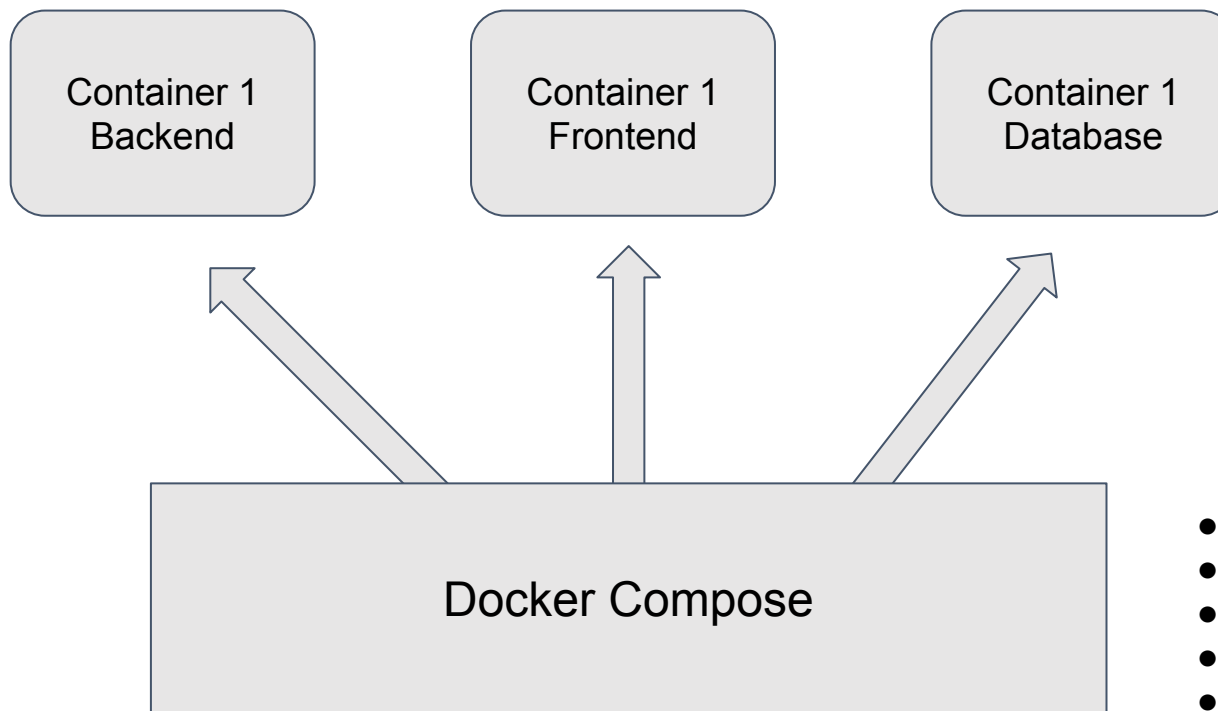
- Aplicação responsável por iniciar contêineres e estabelecer a comunicação entre os contêineres.
- O orquestrador padrão é o **swarm**, mas um muito utilizado é o **kubernetes**

- Repositórios
 - Local para armazenar contêineres prontos para execução
 - Pode ser público ou privado
 - Permite o envio e recuperação de contêineres através de um cliente de console
 - docker push, docker pull

- **Introdução ao Docker**
 - O que é?
 - Qual a vantagem?
 - Conceitos (contêiner, orquestrador, repositório)
 - **Como usar?**
- **Introdução ao Docker HUB**
- **Cloud Container Engine**
 - O que é?
 - Como usar?

- Instalar os aplicativos **docker** e **docker-compose**
- Garantir que o usuário atual está no grupo **docker**, recém criado pela instalação do **docker**
- Criar o arquivo **Dockerfile** na raiz do projeto
- Criar um arquivo **.dockerignore** com as exceções do arquivo **Dockerfile**
- Criar um arquivo **docker-compose.yml** que define como o orquestrador inicia e gerencia o contêiner Docker, estabelecendo comunicação entre contêineres e ordem de inicialização





- Define variáveis de ambiente
- Volumes de dados
- Portas exportas
- Sequência de inicialização
- Configurar redes

- A prática vai ser dividida em 3 etapas:
 - Primeiro vamos criar um contêiner Docker usando a aplicação my-project como fonte
 - Depois vamos enviar o novo contêiner docker para o repositório privado da Vivo
 - Por fim, vamos iniciar uma aplicação usando o **Cloud Container Engine** como orquestrador

Vamos criar um contêiner usando como base a aplicação
my-project

- Acesse a instância ECS que contém a aplicação **my-project** usando **ssh**
 - `ssh -i KeyPair.pem linux@0.0.0.0`
 - Onde **0.0.0.0**, é o IP público da instância ECS
- Instale e habilite o **docker** usando os comando abaixo

```
sudo apt install docker  
sudo apt install docker.io  
sudo usermod -a -G docker linux
```

- O comando `usermod` é necessário para adicionar o usuário **linux** no grupo **docker** e ter acesso ao daemon **dockerd** (pode ser necessário efetuar `logout` e `login` na máquina)

- Arquivo **Dockerfile**
- Criar um arquivo **Dockerfile** na raiz do projeto
 - O arquivo Dockerfile é dividido em **steps** e cada step gera um **layer** como saída
 - Cada **layer** é identificado por um **hash** único
 - **Layers** idênticos possuem o mesmo **hash**
 - Isso permite uma reutilização de **layers** em diferentes contêineres e economiza espaço em disco no repositório

- Arquivo **Dockerfile** que define como o contêiner deve ser construído

```
FROM node:8
RUN npm install sails -g
ENV HOME=/home/node
WORKDIR $HOME/huawei-pratica
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 80
CMD [ "sails", "lift", "--port=80" ]
```

- Arquivo **.dockerignore**
- Arquivo que define quais arquivos e pastas devem ser ignorados pelo arquivo **Dockerfile**
- No nosso caso ignoramos a pasta **node_modules** pois a mesma é criada durante o comando **npm install** já definido no arquivo **Dockerfile**

```
node_modules
```

- Após o arquivo **Dockerfile** e **.dockerignore** serem criados, execute o comando abaixo raiz do projeto

```
docker build . -t huawei:v1.0
```

- “docker” - Executável do docker
- “build” - Criar a imagem do contêiner
- “.” - Use o arquivo Dockerfile que está no diretório atual
- “-t huawei:v1.0” - Crie uma *tag* para identificar a imagem
- Execute então o comando abaixo para iniciar o contêiner e iniciar a aplicação, exportando a porta 80

```
docker run -p 80:80 huawei:v1.0
```

- Com a imagem criada, vamos criar o arquivo do orquestrador **docker-compose** para gerenciar como o contêiner vai ser iniciado
 - O docker-compose permite que variáveis de ambiente sejam passadas ao contêiner
 - Também permite que pastas do contêiner sejam montadas em pastas do *host* (o computador que está executando o **docker-compose**)

- Arquivo **docker-compose.yml**

```
version: '2'
services:
  huawei-app:
    volumes:
      - ./data:/home/node/huawei/data
      - ./huawei:/home/node/huawei
      - /home/node/huawei/node_modules
    ports:
      - "1338:1338"
    build:
      context: huawei
      dockerfile: Dockerfile
    container_name: huawei-container
```


- Introdução ao Docker
 - O que é?
 - Qual a vantagem?
 - Conceitos (contêiner, orquestrador, repositório)
 - Como usar?
- Introdução ao Docker HUB
- Cloud Container Engine
 - O que é?
 - Como usar?

- O Docker HUB é o repositório público oferecido pela empresa Docker
- Pode ser visto como um MarketPlace de contêineres
- Grandes empresas e comunidades open source disponibilizam soluções prontas de contêineres:
 - Microsoft, MySQL, PostgreSQL, Node
- Você pode até mesmo disponibilizar seu contêiner para a comunidade baixar
- Mas **cuidado!** Alguns membros da comunidade disponibilizam contêineres maliciosos.

- Introdução ao Docker
 - O que é?
 - Qual a vantagem?
 - Conceitos (contêiner, orquestrador, repositório)
 - Como usar?
- Introdução ao Docker HUB
- Cloud Container Engine
 - O que é?
 - Como usar?

- Introdução ao Docker
 - O que é?
 - Qual a vantagem?
 - Conceitos (contêiner, orquestrador, repositório)
 - Como usar?
- Introdução ao Docker HUB
- Cloud Container Engine
 - O que é?
 - Como usar?

- Cloud Container Engine (CCE) é a ferramenta da Vivo que permite o gerenciamento de contêineres Docker
- Permite ao desenvolvedor iniciar uma aplicação sem se preocupar com o sistema operacional onde o mesmo está sendo executado
- Faz o papel de orquestrador da aplicação Docker, substituindo o **docker-compose**
- É baseado no projeto **Kubernetes**
- Permite a criação de clusters de dockers que podem utilizar outros serviços da Vivo:
 - **AS** - Escalando automaticamente o cluster, **ELB** - Permitindo o acesso de um único IP ao cluster, **EIP** - Definindo um IP público, **ECS** - Configurando os nós do cluster

- Registro de Contêineres
 - Repositório privado de contêineres disponibilizado pela Vivo. Permite que sua aplicação docker seja enviada para a cloud da Vivo diretamente da linha de comando.
- Cluster
 - O conjunto de computadores onde sua aplicação vai executar. Define o ponto de entrada único da sua aplicação.
- Nó
 - Os computadores que fazem parte do seu cluster são chamados de nós.

- Modelo de Componente
 - Definição dos moldes de uma aplicação. Configuração mínima exigida, repositório docker usado, portas e variáveis de ambiente necessárias.
- Aplicativo
 - A aplicação em execução no cluster. Referencia um modelo de componente.

- Introdução ao Docker
 - O que é?
 - Qual a vantagem?
 - Conceitos (contêiner, orquestrador, repositório)
 - Como usar?
- Introdução ao Docker HUB
- Cloud Container Engine
 - O que é?
 - Como usar?

Vamos enviar nosso contêiner Docker que está na instância
ECS ao repositório privado da Vivo

- Os passos para carregar uma imagem docker no registro de contêineres da Vivo exige uma configuração adicional no computador host (onde o **docker** está em execução).
- Os passos a seguir são executados na interface web do **Cloud Container Engine** e na linha de comando da instância ECS onde o contêiner **Docker** foi configurado anteriormente.

- Acesse **Lista de Serviços > Cloud Container Engine > Registro de Contêineres** (Clique em **Carregar Imagem de Contêiner**)
- Baixe o arquivo indicado e anote o **Endereço do registro de contêineres**

certificado são válidos durante 1 ano por padrão. Se um arquivo de certificado expirar, baixar um novo arquivo de certificado. [Saiba mais](#)

Carregar imagem de contêiner **Excluir arquivos residuais**

1. Conectar com Registro de contêineres privado

Baixar arquivo de certificado e depois autorizar **Download do arquivo dockercfg.txt** [Saiba mais](#)

2. Carregar imagem de contêiner

Antes de você carregar uma imagem de contêiner, especifique uma marca para ela. Uma marca contém o endereço do registro de contêineres privado para o que a imagem vai ser carregada. [Saiba mais](#)

Se a imagem de contêiner precisar de autenticação da assinatura, especificar variáveis de ambiente relacionadas antes de você carregar a imagem do contêiner. [Saiba mais](#)

Endereço do registro de contêineres: **200.121.128.26:443** **container_registry_address**

Endereço do servidor de assinatura de imagem: <https://200.121.128.26:4443>

Inserir um nome de imagem de cor 

- Na instância ECS, execute os comandos abaixo
- Edite ou crie o arquivo **/etc/docker/daemon.json** e insira as linhas abaixo:

```
{  
  "insecure-registries": ["container_registry_address"]  
}
```

- Trocando **container_registry_address** pelo endereço do registro de contêiner anotado anteriormente
- Execute os comandos abaixo

```
sudo systemctl daemon-reload  
sudo service docker restart
```

- Faça download do arquivo **dockercfg.txt**. O conteúdo do arquivo deve ser similar ao abaixo:

```
{"auths":{"200.121.128.26:443":{"auth":"X2F1dGhfdG9rZW46YjI5NDQ0NjNiZmI4NGI2YjgzNTlkODBlYjAxNTcwZjMtVlc1NkVaSjBKV  
FVSUk9MNVJZSFMtMjAxOTA5MjkxMjA2MDQtZTE3MDJjZThlYzZjMD  
FkMWYxMmM3YjJmODg4ZTEyMDliNjI5MDJjMGNhMjQ0MzY5NWU0  
YmFjMmE0M2M3M2FIMQ==","email":""}}}
```

- Execute o comandos abaixo, substituindo **{auth}** pela chave **auths.auth** do arquivo **dockercfg.txt**

```
echo -n {auth}| base64 -d
```

- A saída deste comando será similar ao apresentado abaixo, salve-a:

```
_auth_token:1c1ea38bc3dd4910879170b154ac5eed-MO3WATY4W  
ASMHTWDTOV6-20180824102142-c56d5ccc9dcdd8da9345ddfa02  
b7354425e6e1c0e2a2f7bcbace0ff0c1f20966
```

- Execute o comando abaixo, substituindo **_auth_token** pelo valor retornado pelo comando anterior e **container_registry_address** pelo endereço do registro recuperado no primeiro passo

```
docker login -u _auth_token -p _auth_token container_registry_address
```

- Se o comando retornar **Login Succeeded**, você estará logado no repositório de contêineres privados da Vivo e poderá fazer upload de seus contêineres dockers

- Para fazer upload de um contêiner para o repositório privado execute os seguintes comandos no diretório da aplicação **my-project**:

```
docker build . -t container_registry_address/domain_name/image_name:version
```

```
docker push container_registry_address/domain_name/image_name:version
```

- **{container_registry_address}** é o endereço do registro
- **{domain_name}** é o nome do domínio (*huawei-ufcg em nosso caso*)
- **{image_name}** é o nome único da imagem de sua escolha
- **{version}** é a versão da imagem, pode ser qualquer texto

- Enquanto o upload da imagem ocorre, a tela abaixo deve ser visualizada

```
mysql/mysql-server
mf@mf-desktop-home:~/Projects/huawei-pratica/huawei$ docker push 200.121.128.26:443/huawei-ufcg/huawei-mf
The push refers to repository [200.121.128.26:443/huawei-ufcg/huawei-mf]
f02b31985de0: Pushed
6d5854339533: Pushed
00d03ec47a1c: Pushed
8180fa9bef63: Pushed
bf20ddc400c0: Pushed
3a0e5f45ee0e: Pushed
d9fdc5af195e: Pushed
245ce6af2e7b: Pushed
373fc5310302: Pushed
25494c62cf78: Pushed
d714f65bc280: Pushing [=====>] 54.09MB/320.1MB
fd6060e25706: Pushing [=====>] 39.13MB/123MB
d7ed640784f1: Pushed
1618a71a1198: Pushed
```


- A imagem **docker** foi carregada

As imagens do contêiner privadas são armazenadas no registro de contêineres do CCE. Apenas são acessíveis para determinar o arquivo de certificado expirar, baixar um novo arquivo de certificado. [Saiba mais](#)

Carregar imagem de contêiner

Excluir arquivos residuais

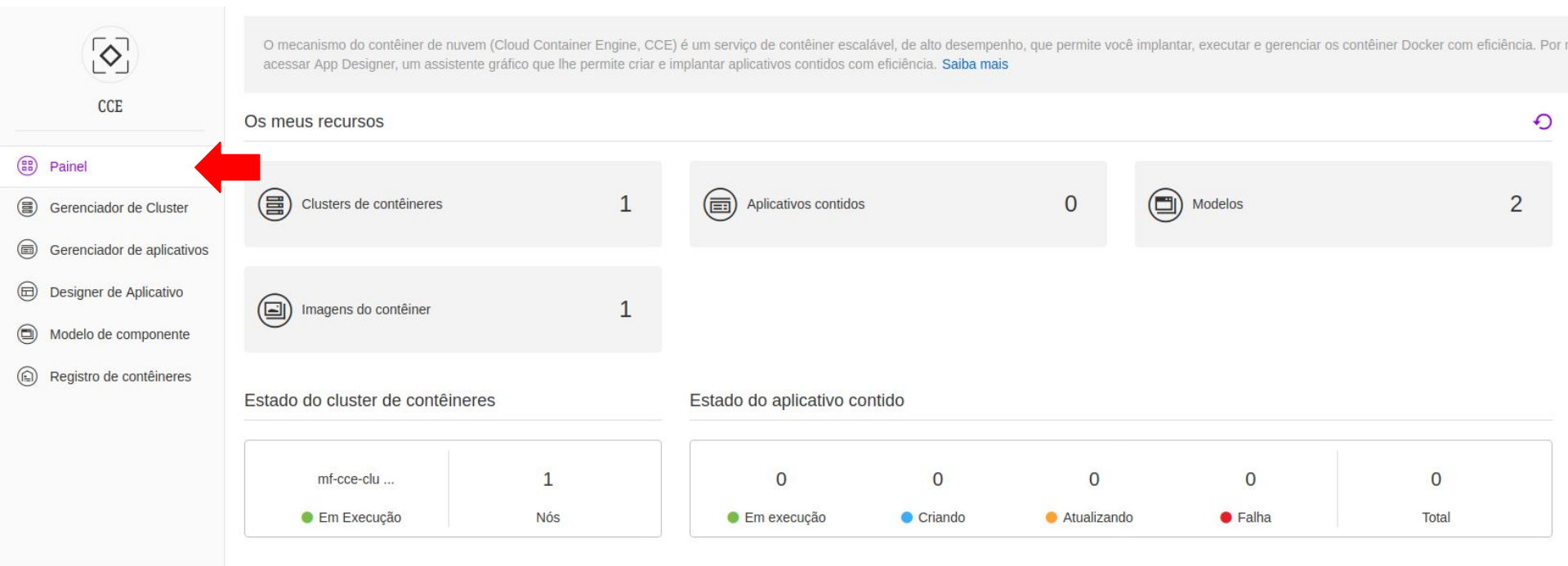
Endereço do registro de contêineres: 200.121.128.26:443 

Endereço do servidor de assinatura de imagem: https://200.121.128.26:4443 

Nome	Descrição
huawei-ufcg/huawei-mf	-

- Vamos passar agora pelas principais opções da interface gráfica do Cloud Container Engine
 - Painel de Resumo
 - Gerenciador de Cluster
 - Gerenciador de Aplicativos
 - Modelo de Componente
 - Registro de Contêineres

- Um resumo da configuração atual é mostrado nesta página



O mecanismo do contêiner de nuvem (Cloud Container Engine, CCE) é um serviço de contêiner escalável, de alto desempenho, que permite você implantar, executar e gerenciar os contêiner Docker com eficiência. Por m acessar App Designer, um assistente gráfico que lhe permite criar e implantar aplicativos contidos com eficiência. [Saiba mais](#)

Os meus recursos

Recurso	Quantidade
Clusters de contêineres	1
Aplicativos contidos	0
Modelos	2
Imagens do contêiner	1

Estado do cluster de contêineres

Nome do Cluster	Estado	Quantidade
mf-cce-clu ...	Em Execução	1
	Nós	

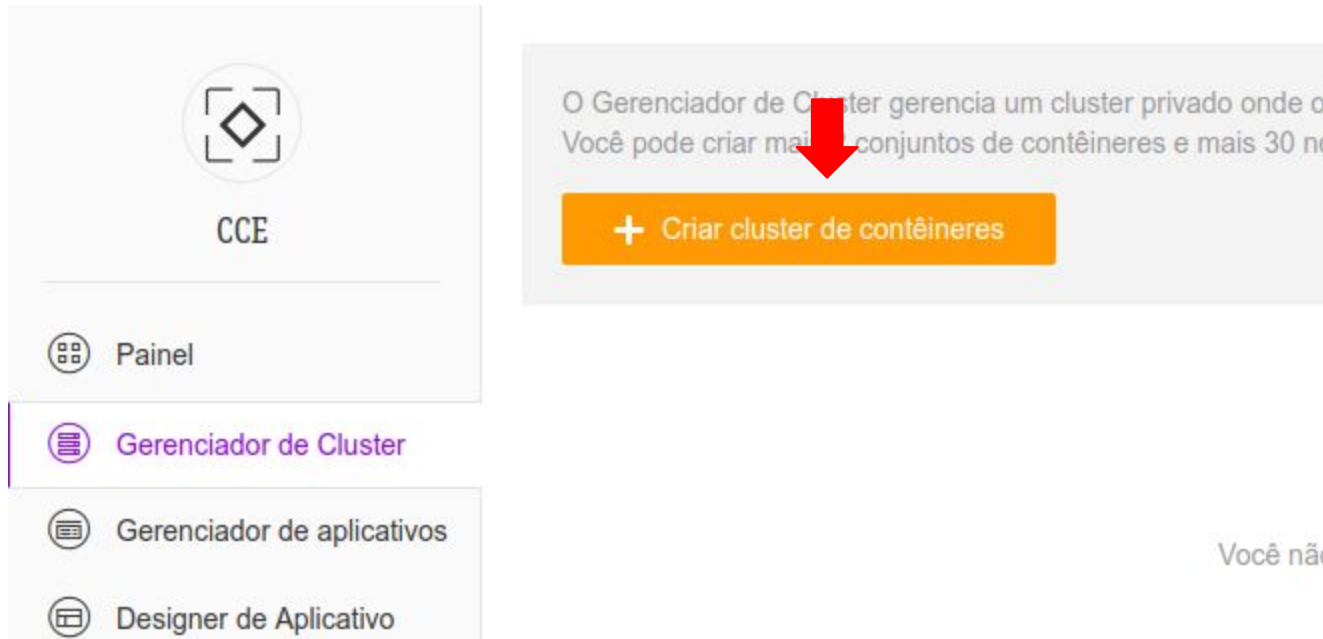
Estado do aplicativo contido

Estado	Quantidade
Em execução	0
Criando	0
Atualizando	0
Falha	0
Total	0

Sidebar de Navegação:

- CCE
- Painel
- Gerenciador de Cluster
- Gerenciador de aplicativos
- Designer de Aplicativo
- Modelo de componente
- Registro de contêineres

- Aqui é possível criar um cluster de contêiner



- Defina os parâmetros do novo cluster:
 - VPC
 - Sub-rede
 - Cluster Type:
 - Single Master
 - HA Master
- Clique em OK

Criar cluster de contêineres

X

* Nome:

?

* VPC:

vpc-mf

▼

↺

* Sub-rede:

subnet-mf--172.16.0.0/24

▼

Security Group:

☒ Automático

☐ Custom

Cluster Type:

☒ Single Master

☐ HA Master

?

Descrição:

Inserir no máximo 200 caracteres.

Nota:

Antes de você criar um cluster de contêineres no console do CCE, carregue o seu arquivo AK/SK. Sem um arquivo AK/SK válido, o cluster de contêineres não pode ser autorizado a usar imagens do contêiner privadas ou ELB. Para uma experiência do usuário ideal e a segurança da conta, é recomendável você carregar o seu arquivo AK/SK e concluir a autenticação do usuário.

OK

Cancelar

- Criar o primeiro nó do cluster
 - Como dito anteriormente, os nós são os computadores que fazem parte do cluster. Criar o primeiro nó significa criar a instância ECS que fará parte do cluster




Adicionar nó para usar o seu cluster de contêineres.

- Criar o primeiro nó do cluster


Progress bar: Especificar detalhes (active) | Confirmar especificações

Informações básicas

- * AZ: ☒ AZ1 ☐ AZ2
- * Tipo de nó: ☒ Para Fins Gerais ☐ Computação I ☐ Computação II ☐ Memória otimizada
- * Especificações:

1 vCPUs 4 GB s1.medium	2 vCPUs 8 GB s1.large	4 vCPUs 16 GB	8 vCPUs 32 GB	16 vCPUs 64 GB	32 vCPUs 128 GB
---------------------------	--------------------------	---------------	---------------	----------------	-----------------
- * Quantidade: 
- * Disco:

Disco do sistema	I/O Comum	40 GB
Disco de dados	I/O Comum	100 GB

[+ Adicionar um disco de dados](#) Você pode anexar mais 9 discos.
- * EIP: ☒ Sim ☐ Não
- * Logon
- * Par de chaves: 

[Adicionar agora](#)

Pode adicionar até 15 computadores de uma vez no cluster

- Criar o primeiro nó do cluster

Progress bar: Especificar detalhes (active) | Confirmar especificações

Informações básicas

- * AZ: ☒ AZ1 ☐ AZ2
- * Tipo de nó: ☒ Para Fins Gerais ☐ Computação I ☐ Computação II ☐ Memória otimizada
- * Especificações:

1 vCPUs 4 GB s1.medium	2 vCPUs 8 GB s1.large	4 vCPUs 16 GB	8 vCPUs 32 GB	16 vCPUs 64 GB	32 vCPUs 128 GB
---------------------------	--------------------------	---------------	---------------	----------------	-----------------
- * Quantidade: 1 Você pode
- * Disco:

Disco do sistema	I/O Comum	40 GB ?
Disco de dados	I/O Comum	100 GB
- [+ Adicionar um disco de dados](#) Você pode anexar mais 9 discos.
- * EIP: ☒ Sim ☐ Não ?
- * Logon
- * Par de chaves: '-Selecionar-'

[Adicionar agora](#)

Um disco de dados é obrigatório

←

- Criar o primeiro nó do cluster

Progress bar: Especificar detalhes (active) | Confirmar especificações

Informações básicas

- * AZ: ☒ AZ1 ☐ AZ2
- * Tipo de nó: ☒ Para Fins Gerais ☐ Computação I ☐ Computação II ☐ Memória otimizada
- * Especificações:

1 vCPUs 4 GB s1.medium	2 vCPUs 8 GB s1.large	4 vCPUs 16 GB	8 vCPUs 32 GB	16 vCPUs 64 GB	32 vCPUs 128 GB
---------------------------	--------------------------	---------------	---------------	----------------	-----------------
- * Quantidade: 1 Você pode
- * Disco:

Disco do sistema	I/O Comum	40 GB ?
Disco de dados	I/O Comum	100 GB

[+ Adicionar um disco de dados](#) Você pode anexar mais 9 discos.
- * EIP: ☒ Sim ☐ Não ?
- * Logon
- * Par de chaves: '-Selecionar-'

[Adicionar agora](#)

Vamos configurar sem um EIP

- Criar o primeiro nó do cluster

Progress bar: Especificar detalhes (active) | Confirmar especificações

Informações básicas

- * AZ: ☒ AZ1 ☐ AZ2
- * Tipo de nó: ☒ Para Fins Gerais ☐ Computação I ☐ Computação II ☐ Memória otimizada
- * Especificações:

1 vCPUs 4 GB s1.medium	2 vCPUs 8 GB s1.large	4 vCPUs 16 GB	8 vCPUs 32 GB	16 vCPUs 64 GB	32 vCPUs 128 GB
---------------------------	--------------------------	---------------	---------------	----------------	-----------------
- * Quantidade: 1 Você pode
- * Disco:

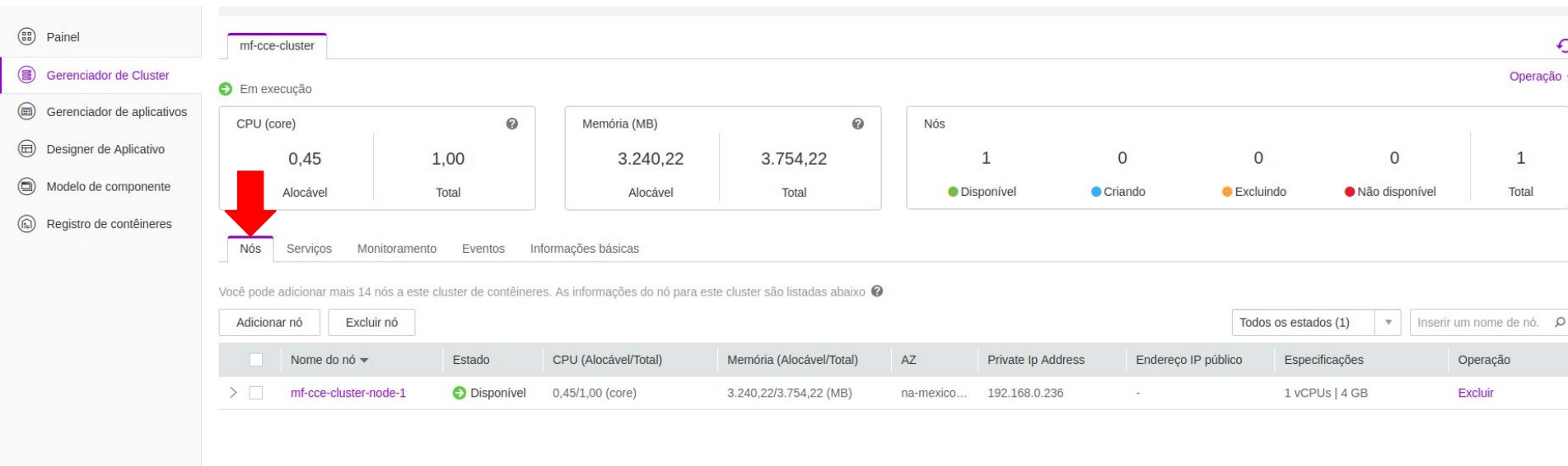
Disco do sistema	I/O Comum	40 GB ?
Disco de dados	I/O Comum	100 GB

[+ Adicionar um disco de dados](#) Você pode anexar mais 9 discos.
- * EIP: ☒ Sim ☐ Não ?
- * Logon
- * Par de chaves: '-Selecionar-' ?

Adicionar agora

Selecione um KeyPair que você tenha em seu computador

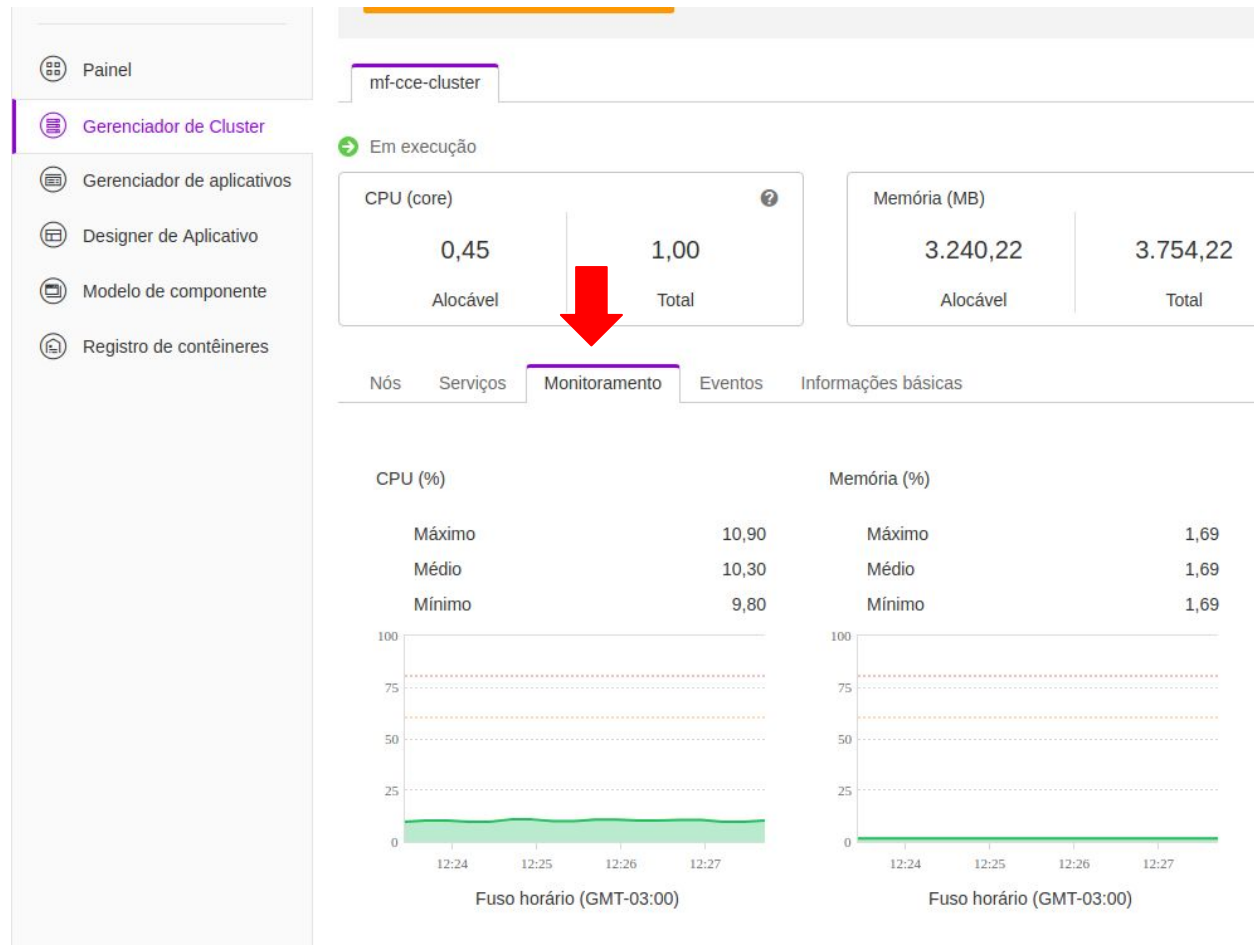
- Configurar nós (computadores) que fazem parte do cluster



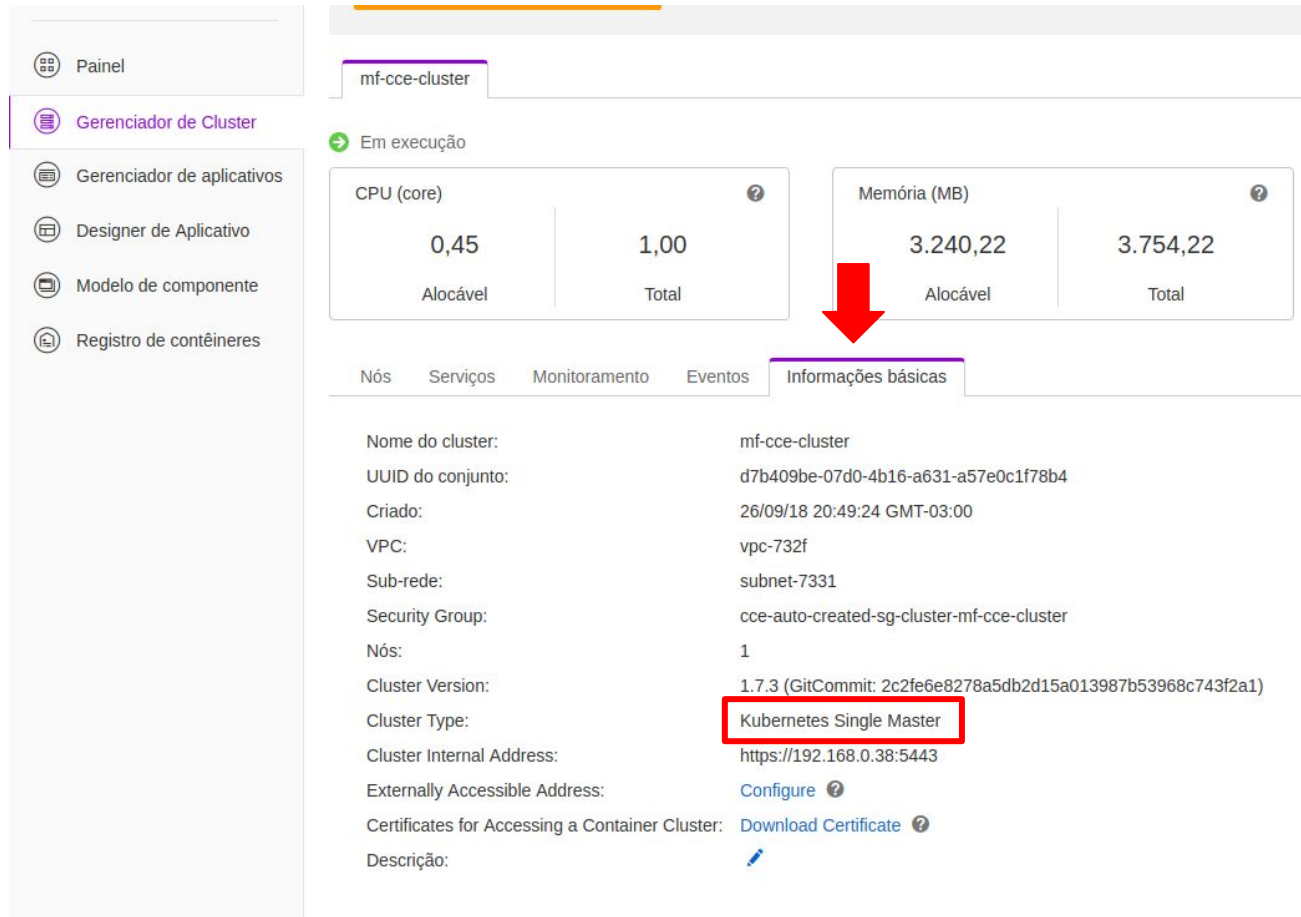
The screenshot displays the CCE Cluster Manager interface. On the left is a sidebar with navigation options: Painel, Gerenciador de Cluster (selected), Gerenciador de aplicativos, Designer de Aplicativo, Modelo de componente, and Registro de contêineres. The main area shows the configuration for a cluster named 'mf-cce-cluster'. It includes summary cards for CPU (0.45 allocatable, 1.00 total), Memory (3,240,22 allocatable, 3,754,22 total), and a node status overview (1 available, 0 creating, 0 excluding, 0 unavailable, 1 total). Below these is a tabbed interface with 'Nós' (Nodes) selected. A red arrow points to the 'Nós' tab. The nodes section shows a message: 'Você pode adicionar mais 14 nós a este cluster de contêineres. As informações do nó para este cluster são listadas abaixo'. There are buttons for 'Adicionar nó' and 'Excluir nó'. A table lists the nodes with columns: Nome do nó, Estado, CPU (Alocável/Total), Memória (Alocável/Total), AZ, Private Ip Address, Endereço IP público, Especificações, and Operação. One node, 'mf-cce-cluster-node-1', is listed as 'Disponível' (Available) with 0.45/1.00 CPU and 3,240,22/3,754,22 MB memory. The 'Operação' column for this node has an 'Excluir' (Exclude) link.

Nome do nó	Estado	CPU (Alocável/Total)	Memória (Alocável/Total)	AZ	Private Ip Address	Endereço IP público	Especificações	Operação
> mf-cce-cluster-node-1	Disponível	0,45/1,00 (core)	3.240,22/3.754,22 (MB)	na-mexico...	192.168.0.236	-	1 vCPUs 4 GB	Excluir

- Monitorar a CPU e memória usadas por um cluster



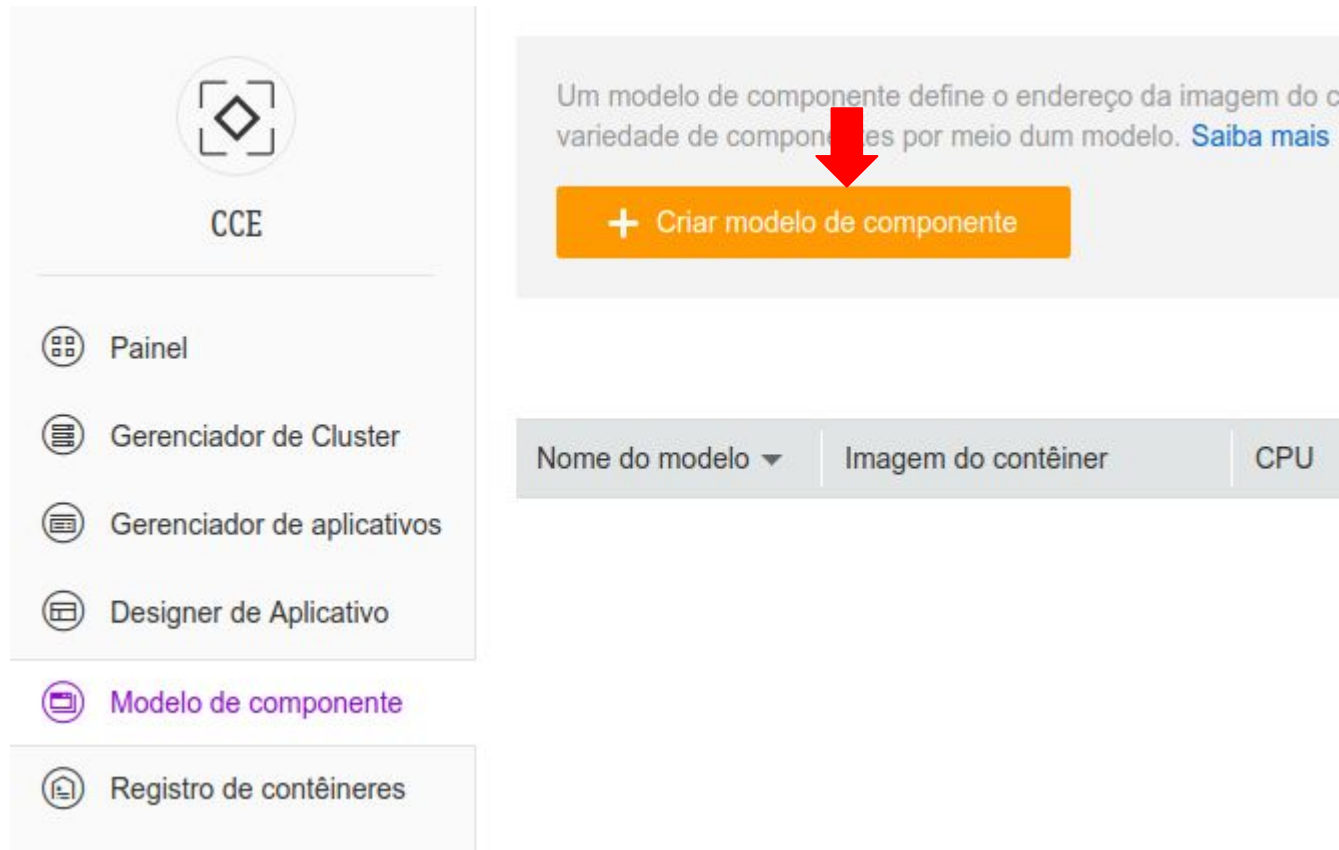
- Monitorar a CPU e memória usadas por um cluster



The screenshot displays the CCE (Cloud Container Engine) dashboard. On the left, a sidebar contains navigation links: Painei, Gerenciador de Cluster (highlighted), Gerenciador de aplicativos, Designer de Aplicativo, Modelo de componente, and Registro de contêineres. The main content area shows the 'mf-cce-cluster' status as 'Em execução' (Running). Below this, two summary cards are visible: 'CPU (core)' showing 0,45 Alocável and 1,00 Total, and 'Memória (MB)' showing 3.240,22 Alocável and 3.754,22 Total. A red arrow points from the memory card to the 'Informações básicas' tab. This tab displays various cluster details:

Nome do cluster:	mf-cce-cluster
UUID do conjunto:	d7b409be-07d0-4b16-a631-a57e0c1f78b4
Criado:	26/09/18 20:49:24 GMT-03:00
VPC:	vpc-732f
Sub-rede:	subnet-7331
Security Group:	cce-auto-created-sg-cluster-mf-cce-cluster
Nós:	1
Cluster Version:	1.7.3 (GitCommit: 2c2fe6e8278a5db2d15a013987b53968c743f2a1)
Cluster Type:	Kubernetes Single Master
Cluster Internal Address:	https://192.168.0.38:5443
Externally Accessible Address:	Configure
Certificates for Accessing a Container Cluster:	Download Certificate
Descrição:	

- Se a imagem estiver carregada na Vivo, o próximo passo é criar um modelo de componente a partir desta imagem




The screenshot displays the CCE (Componente de Criação de Elementos) interface. On the left, a sidebar menu lists several options: 'Painel', 'Gerenciador de Cluster', 'Gerenciador de aplicativos', 'Designer de Aplicativo', 'Modelo de componente' (highlighted in purple), and 'Registro de contêineres'. The main area on the right contains a text box explaining that a component model defines the image address for a variety of components, with a link to 'Saiba mais'. Below this text is an orange button labeled '+ Criar modelo de componente'. A red arrow points from the text in the slide to this button. At the bottom of the main area, there is a table with three columns: 'Nome do modelo', 'Imagem do contêiner', and 'CPU'.

Nome do modelo ▼	Imagem do contêiner	CPU
------------------	---------------------	-----

- Forneça a informação necessária

* Nome do modelo: ?

* Imagem do contêiner: ☒ Imagem de contêiner interna  ☐ Imagem de contêiner externa ?

Nome da imagem : Rótulo ?

* Rede: ?

* Porta: ?

Memória: MB ?

CPU: mcore ☒ Aceitar adequação de CPU ?

Volumes: ?

Variáveis de ambiente: ?

CCE - Criando modelo de componente



- Forneça a informação necessária

* Nome do modelo: ?

* Imagem do contêiner: ☒ Imagem de contêiner interna ☐ Imagem de contêiner externa ?

Nome da imagem : Rótulo ?

* Rede: ?

* Porta: ?

Memória: MB ?

CPU: mcore ☒ Aceitar adequação de CPU ?

Volumes: ?

Variáveis de ambiente: ?

CCE - Criando a aplicação contida



CCE



Painel



Gerenciador de Cluster



Gerenciador de aplicativos



Designer de Aplicativo



Modelo de componente



Registro de contêineres

O gerenciador de aplicativos cria, atualiza, exclui e gerencia aplicativos contidos no seu cluster de contêineres privado. Após um ser acessado por meio dos endereços de serviço fornecidos por CCE. [Saiba mais](#)

+ Criar aplicativo contido



Nome do aplicativo ▼

Estado ▼

Cluster de contêineres ▼

Namespace ▼

C

Não há dados disponíveis.

CCE - Criando a aplicação contida



- Forneça as informações da aplicação
 - Escolha **Modelo de componente**

The screenshot shows the CCE (Container Cloud Edge) interface for creating a contained application. On the left is a sidebar with navigation options: CCE (selected), Painel, Gerenciador de Cluster, Gerenciador de aplicativos (highlighted with a purple bar), Designer de Aplicativo, Modelo de componente, and Registro de contêineres. The main area contains a form with the following fields:

- Template Type:** Two radio buttons. 'Modelo de aplicativo' is unselected, and 'Modelo de componente' is selected (indicated by a red arrow).
- Cluster de contêineres:** A dropdown menu showing 'mf-cce-cluster-1'.
- Nome do aplicativo:** An empty text input field.
- Componente de aplicativo:** A button labeled 'Criar componente'.

At the bottom of the form are two buttons: 'OK' and 'Cancelar'.

CCE - Criando a aplicação contida



- Forneça as informações da aplicação
 - Escolha um nome único para a aplicação

CCE

Painel

Gerenciador de Cluster

Gerenciador de aplicativos

Designer de Aplicativo

Modelo de componente

Registro de contêineres

* Template Type: ☐ Modelo de aplicativo ☒ Modelo de componente ?

* Cluster de contêineres: mf-cce-cluster-1 ▼

* Nome do aplicativo:

* Componente de aplicativo:

CCE - Criando a aplicação contida



- Forneça as informações da aplicação
 - Crie um novo componente de aplicativo

CCE

Painel

Gerenciador de Cluster

Gerenciador de aplicativos

Designer de Aplicativo

Modelo de componente

Registro de contêineres

* Template Type: ☐ Modelo de aplicativo ☒ Modelo de componente ?

* Cluster de contêineres: mf-cce-cluster-1 ▼

* Nome do aplicativo:

* Componente de aplicativo:

OK Cancelar

CCE - Criando a aplicação contida



- Forneça as informações do componente
 - Um nome

Criar componente

X

* Nome do modelo:

mf-cce-peru-componente

▼

* Nome do componente:

Descrição do componente:

Inserir no máximo 200 caracteres.

AZ:

☒ Automático ☐ Custom ?

Nó:

Automático

▼

?

* Instâncias:

1

▲

▼

* Nome do serviço:

Nome do serviço usado para a comunicação do componente.

* Serviço privado:

☒ ?

* Serviço público:

☐ Sim ☒ Não ?

OK

Cancelar

CCE - Criando a aplicação contida



- Forneça as informações do componente
 - O nó (deve ter somente um no cluster)

Criar componente

X

* Nome do modelo:

mf-cce-peru-componente

▼

* Nome do componente:

Descrição do componente:

Inserir no máximo 200 caracteres.

AZ:

☒ Automático ☐ Custom ?

Nó:

Automático

▼

* Instâncias:

1

▲▼

* Nome do serviço:

Nome do serviço usado para a comunicação do componente.

* Serviço privado:

☒ ?

* Serviço público:

☐ Sim ☒ Não ?

OK

Cancelar



CCE - Criando a aplicação contida



- Forneça as informações do componente
 - Deixe 1 instância

Criar componente

X

* Nome do modelo:

mf-cce-peru-componente

▼

* Nome do componente:

Descrição do componente:

Inserir no máximo 200 caracteres.

AZ:

☒ Automático ☐ Custom ?

Nó:

Automático

▼

?

* Instâncias:

1

▲▼

* Nome do serviço:

Nome do serviço usado para a comunicação do componente.

* Serviço privado:

☒ ?

* Serviço público:

☐ Sim ☒ Não ?

OK

Cancelar

CCE - Criando a aplicação contida



- Forneça as informações do componente
 - Marque serviço público como **Sim**

Criar componente

X

* Nome do modelo:

mf-cce-peru-componente

▼

* Nome do componente:

Descrição do componente:

Inserir no máximo 200 caracteres.

AZ:

☒ Automático ☐ Custom ?

Nó:

Automático

▼

?

* Instâncias:

1

▲

▼

* Nome do serviço:

Nome do serviço usado para a comunicação do componente.

* Serviço privado:

☒ ?

* Serviço público:

☐ Sim ☒ Não ?

OK

Cancelar

CCE - Criando a aplicação contida



- Forneça as informações do componente
 - Defina um LoadBalancer para controlar o acesso à aplicação. Caso não tenha um LoadBalancer, será necessário configurar um.
 - Redirecione a porta do contêiner para uma porta externa livre.

Tipo de serviço: LoadBalancer ▼

Balanceador de carga: elb-alessandro; 200.121.128.24; 5Mbps B ▼ ↺ ⓘ

Parâmetros de serviço:

Protocolo de rede: TCP ▼ Porta de contêiner: 80 ▼ Porta ELB: 81 +

OK

Cancelar

- A imagem já está no repositório privado de cada região, vamos criar aplicações com base nesta imagem:
 - Um representante de cada região vai fazer o papel de gerente operacional e criar o **cluster de contêineres** e ao menos um **nó** para execução das aplicações
 - Todos os integrantes da região vão criar um **modelo de componente** e uma **aplicação** cada um, todos dentro do mesmo cluster criado anteriormente
 - Todas as aplicações vão estar conectadas no mesmo **ELB** e cada um vai ser hospedado em uma porta específica (81,82,83 e 84)
 - Utilize os slides anteriores como guia para cada etapa da criação da aplicação



Contact

Angelo Perkusich, D.Sc.

Professor, CEO

angelo.perkusich@embedded.ufcg.edu.br

+55 83 8811.9545

Hyggo Almeida, D.Sc.

Professor, CTO

hyggo.almeida@embedded.ufcg.edu.br

+55 83 8875.1894

Rohit Gheyi

Professor, Program Manager

rohit.gheyi@embedded.ufcg.edu.br

+55 83 8811 3339

