



Anotações

Métodos de instalação do Cluster Kubernetes

Objetivos da Aula

- Conhecer métodos de instalação do Cluster Kubernetes;
- Requisitos de instalação do Cluster Kubernetes.

Competências da Aula

Esta aula possui conteúdo que atende aos objetivos da seguinte certificação:

Certified Kubernetes Administrator (CKA)

Installation, Configuration & Validation (13% of exam);



Conhecer métodos de instalação do Cluster Kubernetes

Anotações

Métodos de instalação do Cluster Kubernetes

Instalação manual

- 1** – Instale manualmente;
- 2** – Configure seu próprio ambiente de rede;
- 3** – Localize os binários de lançamento;
- 4** – Construa suas próprias imagens;
- 5** – Comunicação de cluster seguro.



Métodos de Instalação

Você pode ter boas razões para executar em um ambiente que você controla de perto (segurança, regulamentação, conformidade, desempenho e custo). Se este for o caso, os provedores de nuvem podem não ser iniciantes. Mas você ainda pode aproveitar todos os benefícios do Kubernetes, executando você mesmo. Você tem o conhecimento, as habilidades e a experiência para gerenciar a infraestrutura subjacente, já que já está fazendo isso.

Métodos de instalação do Cluster Kubernetes

Instalação pré-compilação

- ✓ Minikube;
- ✓ Minishift;
- ✓ MicroK8s;
- ✓ Ubuntu on LXD;
- ✓ Google Cloud;
- ✓ Azure;
- ✓ AWS.



Métodos de Instalação

- ✓ **Minikube:** Ferramenta que facilita a execução local do Kubernetes. O Minikube executa um cluster de Kubernetes de nó único, dentro de uma Máquina Virtual (VM) em seu laptop para usuários que queiram experimentar o Kubernetes ou desenvolvê-lo no dia-a-dia;
- ✓ **Minishift:** Ferramenta que ajuda você a executar o OKD localmente, iniciando um cluster OKD de nó único, dentro de uma máquina virtual. Com o Minishift, você pode experimentar o OKD ou desenvolvê-lo no dia a dia em sua máquina local;
- ✓ **MicroK8s:** Implementa Kubernetes a montante, certificada pela CNCF que é executada inteiramente na sua estação de trabalho. Sendo um snap, ele executa todos os serviços do Kubernetes de forma nativa (ou seja, sem máquinas virtuais), enquanto empacota todo o conjunto de bibliotecas e binários necessários;
- ✓ **Ubuntu on LXD:** O CDK é uma distribuição do Kubernetes empacotada como um pacote de charms para o Juju, o modelador de aplicativos de código aberto;
- ✓ **Google Cloud, Azure e AWS:** O Kubernetes lhe dá a oportunidade de substituir várias camadas de gerenciamento, monitoramento e segurança que você precisava para construir, integrar e manter-se com uma experiência engenhosa. Na verdade, existem alguns provedores de nuvem compatíveis com o Kubernetes, como exemplo: GKE do Google, Microsoft AKS e Amazon's EKS.

Métodos de instalação: Provedores de Nuvem

Anotações

Métodos de instalação do Cluster Kubernetes

Google GKE

Microsoft Azure AKS

Amazon AWS EKS

✓ O Kubernetes, claro, veio do Google. O GKE é a oferta gerenciada do Kubernetes pelo Google. Os Google SREs gerenciarão o plano de controle do Kubernetes para você e obterão atualizações automáticas. Como o Google tem muita influência no Kubernetes e o utiliza como solução de orquestração de contêineres da plataforma de nuvem do Google desde o primeiro dia, seria muito estranho se não tivesse a melhor integração.

Google GKE

Da mesma forma, você pode confiar que o GKE seja o mais atualizado. Muito mais testes de novos recursos e recursos do Kubernetes acontecem no GKE do que em outros provedores de nuvem. No GKE, você não precisa pagar pelo plano de controle do Kubernetes. O Google cobrirá você e você paga apenas pelos nós do trabalhador. Você também obtém o GCR (Google Container Registry), o registro central integrado e o monitoramento, por meio do Stackdriver Logging e do Stackdriver Monitoring. Se você estiver interessado em uma integração ainda maior com o pipeline de IC / CD, poderá usar o Google Code Build.

Para saber mais sobre o Google GKE, acesse o seguinte link:

<https://cloud.google.com/kubernetes-engine/>

Métodos de instalação do Cluster Kubernetes

Google GKE

Microsoft Azure AKS

Amazon AWS EKS

✓ O AKS é muito semelhante ao GKE. Também gerencia um cluster do Kubernetes para você, gratuitamente. Ele também é certificado pelo CNCF como Kubernetes em conformidade (sem cortes personalizados). A Microsoft investiu muito no Kubernetes em geral e no AKS em particular. Há uma forte integração com o ActiveDirectory para autenticação e autorização, monitoramento e registro integrados e armazenamento do Azure. Você também obtém registro interno de contêiner, rede e nós habilitados para GPU.

Microsoft Azure AKS

Um dos recursos mais interessantes do AKS, é o uso do projeto virtual-kublet para integração com o ACI (Azure Container Instances). A ACI elimina a necessidade de provisionar nós para o cluster, o que é uma grande dificuldade se você estiver lidando com uma carga altamente variável.

Para saber mais sobre o Microsoft Azure AKS, acesse o seguinte link:

<https://azure.microsoft.com/pt-br/services/kubernetes-service/>

Métodos de instalação do Cluster Kubernetes

Google GKE

Microsoft Azure AKS

Amazon AWS EKS

✓ A Amazon sempre teve sua própria plataforma de orquestração de contêineres ECS (Elastic Container Service). Mas a demanda do cliente era que o Kubernetes fosse esmagador. Muitas organizações executavam seus clusters Kubernetes no EC2, usando Kops ou similares. A AWS decidiu fornecer suporte adequado com integrações oficiais. A EKS hoje se integra ao IAM para gerenciamento de identidades, balanceadores de carga da AWS, rede e várias opções de armazenamento.

Amazon AWS EKS

Quanto ao desempenho, o EKS está em algum lugar no meio. Leva de 10 a 15 minutos para iniciar um cluster. Obviamente, o desempenho de sistemas distribuídos complexos é muito diversificado e não pode ser capturado por uma única métrica. Dito isso, o EKS em si ainda é relativamente novo e pode levar algum tempo até que ele possa tirar total proveito da base robusta da AWS.

Para saber mais sobre o Amazon AWS EKS. acesse o seguinte link:

<https://aws.amazon.com/pt/eks/>

Requisitos de instalação do Cluster Kubernetes

Anotações

Requerimentos para Nó Master

- ✓ Sistema físico/virtual/instância em uma IaaS pública ou privada;
- ✓ RHEL 7.3+/RHEL Atomic Host 7.4.5+/CentOS7+/Ubuntu 16 Server+;
- ✓ Mínimo 2/4 vCPU (adicionais são altamente recomendados);
- ✓ Mínimo 4/8/16GB RAM (memória adicional, se o etcd estiver no Master);
- ✓ Mínimo de 40GB de espaço em disco para o /var;
- ✓ Mínimo de 1GB de espaço em disco para o /usr/local/bin.

Anotações

Requerimentos para os Nodes

- ✓ Sistema físico/virtual/instância em uma IaaS pública ou privada.
- ✓ RHEL 7.3+/RHEL Atomic Host 7.4.5+/CentOS7+/Ubuntu 16 Server+.
- ✓ Mínimo 2 vCPU.
- ✓ Mínimo 4/8 RAM.
- ✓ Mínimo de 80GB de espaço em disco para o /var
- ✓ Mínimo de 1GB de espaço em disco para o /usr/local/bin.

Anotações

Recapitulando

- Conhecer métodos de instalação do Cluster Kubernetes;
- Requisitos de instalação do Cluster Kubernetes.

Anotações



Anotações

Objetivos da Aula

- Migração do Docker UCP / Swarm para Kubernetes;
- Instalar pacotes do Docker e Kubernetes.

Competências da Aula

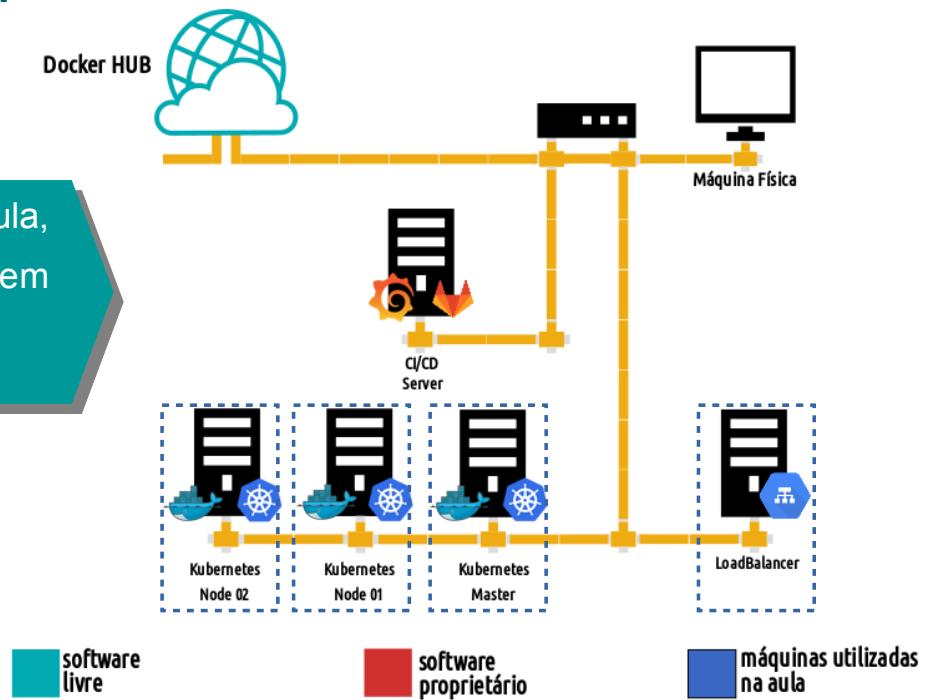
Esta aula possui conteúdo que atende aos objetivos da seguinte certificação:

Certified Kubernetes Administrator (CKA)

Installation, Configuration & Validation (13% of exam);



Para realizar esta aula,
ligue as máquinas em
destaque:



Anotações

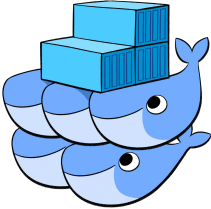
This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

Migração do Docker UCP / Swarm para Kubernetes


Anotações

Conceitos: Docker UCP para Kubernetes

- ✓ Manager Node
- ✓ Worker Node
- ✓ Service
- ✓ Task
- ✓ Replicas
- ✓ DNS-based Service Discovery
- ✓ Ingress
- ✓ PublishedPort
- ✓ Volumes
- ✓ Overlay Network



- ✓ Master
- ✓ Node
- ✓ Deployment
- ✓ Pod
- ✓ Replica Set
- ✓ DNS-based Service Discovery
- ✓ Service
- ✓ Endpoint
- ✓ Persisted Volumes
- ✓ Flat Networking space



Anotações

	Swarm	Kubernetes
Controller	Manager	Master
Slave	Worker	Nodes
Workload Definition	Service	Deployment
Deployment Unit	Task	Pod
Scale-out Definition	Replicas	Replica Set
Service Discovery	DNS	DNS
Load Balancing	Ingress	Service
Port	PublishedPort	Endpoint
Storage	Volumes	Persistent Volumes / Claims
Network	Overlay	Flat Networking Space

Anotações

ETAPA 1 – Backup e desinstalação do Docker UCP

- 1. Obter o ID do Cluster UCP:
\$ sudo docker container run <opções> docker/ucp id
- 2. Realizar o backup do Docker UCP:
\$ sudo docker container run <opções> docker/ucp backup > bkp_ucp.tar
- 3. Desinstalação do Docker UCP:
\$ sudo docker container run <opções> uninstall-ucp

Anotações

ETAPA 2 – Backup e desinstalação do Cluster Swarm

1. Parar o serviço do Docker no Manager e fazer backup da pasta do Swam:

```
$ sudo systemctl stop docker

$ sudo tar cvzf "/tmp/swarm-$(ENGINE)-$(hostname -s)-$(date +%s%z).tgz"
/var/lib/docker/swarm/
```

2. Remover o nó Worker do Cluster Swarm:

```
$ sudo swarm leave
```

3. Remover o nó Master do Cluster Swarm:

```
$ sudo swarm leave --force
```

Anotações

Instalar o Kubernetes no Ubuntu Server

Anotações

Configuração do Docker – ETAPA 1

1

Logue com usuário suporte e instale os pacotes para ativar o suporte à repositórios https:
\$ `sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common`

2

Em seguida, baixe a chave pública para configurar o repositório do Docker:
\$ `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`



Kube Master, Kube Node 1 e 2

Desinstalar versões antigas

Versões mais antigas do Docker eram chamadas docker, docker.io ou docker-engine. Se estes estiverem instalados, desinstale-os.

Distribuições baseadas no RedHat:

\$ `sudo yum erase docker docker-engine docker.io containerd runc`

Distribuições baseadas no Debian:

\$ `sudo apt-get remove docker docker-engine docker.io containerd runc`

Configuração do Docker – ETAPA 2

3

Use o seguinte comando para configurar o repositório estável para a distribuição Ubuntu:

```
$ sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable"
```

4

Atualize a lista de pacotes e instale o Docker versão 18.06.1, recomendado para o Cluster:

```
$ sudo apt-get update
$ sudo apt-get install docker-ce=18.06.1~ce~3-0~ubuntu -y
```

5

Adicione o usuário suporte ao grupo "docker" para ter acesso ao comando "docker":

```
$ sudo usermod -aG docker suporte
```



Kube Master, Kube Node 1 e 2

Desinstalar versões atuais

Se o ambiente está sendo migrado de um Docker UCP/Swarm que utiliza versões mais atuais do Docker, desinstale-os.

Distribuições baseadas no RedHat:

```
$ sudo yum erase docker-ce docker-ce-cli containerd.io -y
```

Distribuições baseadas no Debian:

```
$ sudo apt purge --remove docker-ce docker-ce-cli containerd.io -y
```


Desativar SWAP

1

Edite o arquivo `/etc/fstab` e remova (ou comente) a linha do Swap:

```
$ sudo vim /etc/fstab
```

```
....
```

```
#/dev/mapper/centos-swap swap swap defaults 0 0
```

2

Desative o Swap, sem precisar reiniciar o servidor:

```
$ sudo swapoff -a
```

3

Confirme se o Swap está desativado no servidor:

```
$ sudo swapon -s
```



Kube Master, Kube Node 1 e 2

Por que desativar o Swap?

O Kubernetes é um sistema distribuído, projetado para operar em escala. Ao executar um grande número de contêineres em uma vasta frota de máquinas, você deseja previsibilidade e consistência. Desativar o swap é a abordagem correta. É melhor matar um único contêiner do que vários contêineres em uma máquina numa taxa imprevisível, provavelmente lenta.

Portanto, o kubelet não foi projetado para lidar com situações de troca. Espera-se que a demanda por carga de trabalho se encaixe na memória do host.

Baixe a chave pública para configurar o repositório do Kubernetes:

```
$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
```

Baixe a chave pública para configurar o repositório do Kubernetes:

```
$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
```

Use o seguinte comando para configurar o repositório estável para a distribuição Ubuntu:

```
$ cat << EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
EOF
```

Em seguida, atualize a lista de pacotes:

```
$ sudo apt-get update
```

Kube Master, Kube Node 1 e 2

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

Instalação do Kubernetes – ETAPA 2

4

Instale o Kubernetes na versão 1.15.1, a partir do seguinte comando:
\$ sudo apt install -y kubelet=1.15.1-00 kubeadm=1.15.1-00 kubectl=1.15.1-00

5

Verifique qual é a versão do kubeadm que esta instalada:
\$ sudo kubeadm version



Kube Master, Kube Node 1 e 2

Versão do Kubernetes

Em nossa infraestrutura não estamos instalando a versão mais atual do Kubernetes. Isso é proposital, a fim de criar um ambiente que será atualizado futuramente.

Recapitulando

- Migração do Docker UCP / Swarm para Kubernetes;
- Instalar pacotes do Docker e Kubernetes.

Anotações



Anotações

Objetivos da Aula

- Configurar Cluster Kubernetes;
- Listar componentes do Cluster;
- Ativar rede Calico no Cluster.

Competências da Aula

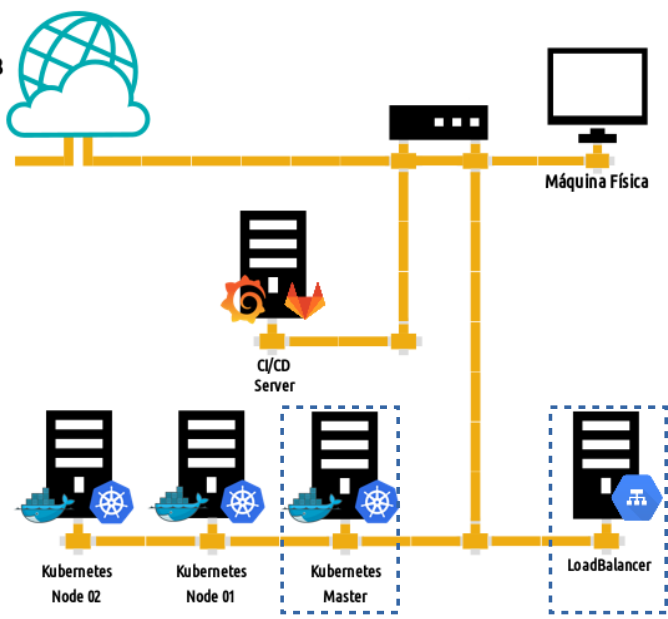
Esta aula possui conteúdo que atende aos objetivos da seguinte certificação:

Certified Kubernetes Administrator (CKA)

Installation, Configuration & Validation (13% of exam);



Para realizar esta aula,
ligue as máquinas em
destaque:

[illegible]

Instalação do Cluster

1

Logue com usuário suporte e verifique as opções do comando kubeadmin:

```
$ sudo kubeadm --help
```

2

Execute o comando para listar e extrair as imagens que o kubeadm vai utilizar:

```
$ sudo kubeadm config images pull --kubernetes-version=1.15.1
```

3

Para configurar o cluster kubernetes, execute o comando kubeadmin init:

```
$ sudo kubeadm init --pod-network-cidr=192.168.0.0/16
```



Kube Master

Comando kubeadmin

O kubeadm é uma ferramenta CLI que instala e configura os vários componentes de um cluster de maneira padrão. O subcomando **init** executa primeiro uma série de pré-verificações, para garantir que a máquina esteja pronta para executar o Kubernetes. Esses pré-controles, expõem avisos e saem sem erros.

A opção **--pod-network-cidr** especifica a sub-rede privada, da qual os IPs do pod serão atribuídos.

Configuração de acesso ao Cluster

- | | |
|---|--|
| 1 | Crie o diretório que armazenará os arquivos de configuração do Cluster Kubernetes:
<code>\$ mkdir -p \$HOME/.kube</code> |
| 2 | Copie o arquivo de configuração admin.conf para o diretório criado no comando anterior:
<code>\$ sudo cp -i /etc/kubernetes/admin.conf \$HOME/.kube/config</code> |
| 3 | Ajuste a permissão de acesso de configuração do Cluster ao usuário suporte:
<code>\$ sudo chown \$(id -u):\$(id -g) \$HOME/.kube/config</code> |



Anotações

Listar componentes do Cluster – ETAPA 1

1

Verifique as opções do comando kubectl:

```
$ kubectl --help
```

2

Para listar os namespaces, execute o seguinte comando:

```
$ kubectl get namespaces
```

3

Para listar os Pods e Deployments, execute os seguintes comandos:

```
$ kubectl get pods
```

```
$ kubectl get deployments
```



Kube Master

Descrição dos comandos

O kubectl é uma ferramenta CLI usada para emitir comandos para o cluster, através do seu servidor de API. O subcomando **get** exibe uma tabela com as informações mais importantes sobre os recursos especificados.

Listar componentes do Cluster – ETAPA 2

4

Para listar os serviços, execute o seguinte comando:

```
$ kubectl get services
```

4

Para listar os Pods do namespace kube-system, execute os seguintes comandos:

```
$ kubectl get pods -n kube-system
```

6

Para listar os Deployments e Serviços do namespace kube-system, execute:

```
$ kubectl get deployments -n kube-system
```

```
$ kubectl get services -n kube-system
```



Kube Master

Descrição dos comandos

Ao utilizar o subcomando get, você verá apenas resultados do namespace default e/ou seu namespace atual, a menos que a flag **-n** seja utilizada.

Em nosso exemplo, estamos listando Pods, Deployments e Serviços do namespace **kube-system**. Para listar recursos de todos os namespaces, utilize a opção **--all-namespaces**.

Exemplo:

```
$ kubectl get pods --all-namespaces
```

Ativar rede Calico no Cluster

1

Antes de aplicar as configurações de rede, verifique o arquivo exemplo na seguinte pasta:

```
$ls aula02/network
```

2

Aplique as configurações da rede Calico no cluster, através do seguinte comando:

```
$kubectl apply -f aula02/network/calico.yaml
```

3

Verifique se os containers da rede Calico estão presentes no Cluster Kubernetes:

```
$kubectl get pods -n kube-system
```



Kube Master

Rede Calico

O Calico permite a política de rede nos clusters do Kubernetes em toda a nuvem. O Calico funciona em todos os lugares - em todos os principais provedores de nuvem pública e em nuvem privada também.

O Calico usa uma malha de rede IP pura para fornecer rede de alto desempenho, e seu mecanismo de política testado em batalha, reforça a política de rede de alto nível, que é voltada para a intenção. Juntos, o Calico e o Kubernetes fornecem uma plataforma nativa e segura na nuvem, que pode dimensionar sua infraestrutura para centenas de milhares de cargas de trabalho.

O subcomando **apply -f** aplica uma configuração a um recurso por nome de arquivo ou stdin. Em nosso arquivo, estamos aplicando as configurações do arquivo **calico.yaml**.




Para aplicar as configurações a partir do site do projeto Calico, utilize o seguinte comando:

```
$ kubectl apply -f https://docs.projectcalico.org/v3.7/manifests/calico.yaml
```

Ajustes em todos os Nós do Cluster, caso utilize Rede Flannel:

```
$ cat << EOF | sudo tee /etc/sysctl.conf
net.bridge.bridge-nf-call-iptables = 1
EOF
$ sudo sysctl -p
```

Recapitulando

-  Configurar Cluster Kubernetes;
-  Listar componentes do Cluster;
-  Ativar rede Calico no Cluster.

Anotações



Anotações

Objetivos da Aula

- Gerenciar Tokens no Cluster;
- Adicionar nó do Cluster Kubernetes;
- Remover nó do Cluster Kubernetes.

Competências da Aula

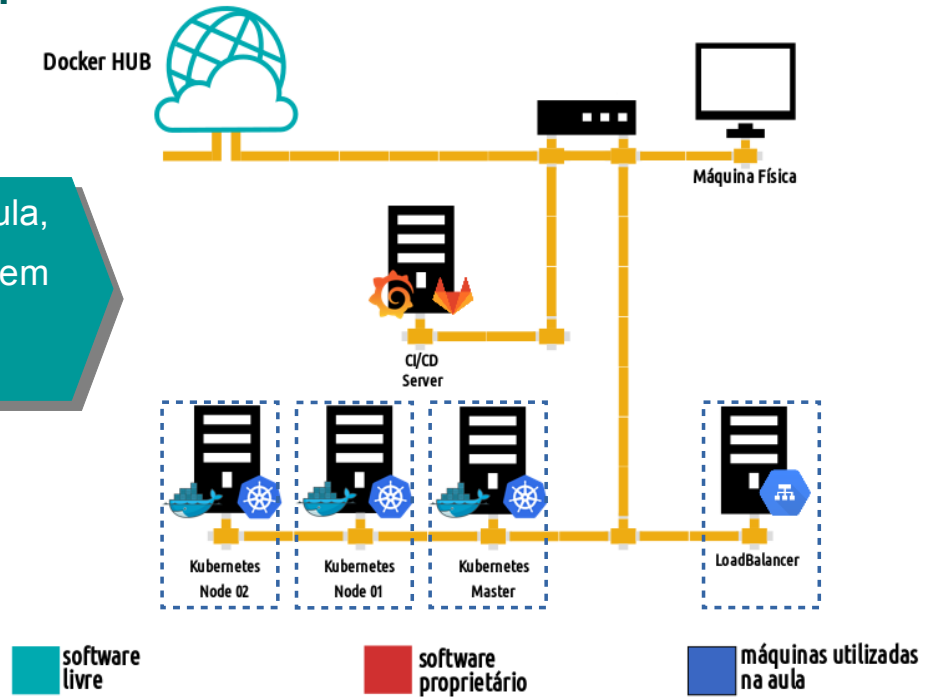
Esta aula possui conteúdo que atende aos objetivos da seguinte certificação:

Certified Kubernetes Administrator (CKA)

Installation, Configuration & Validation (13% of exam);



Para realizar esta aula,
ligue as máquinas em
destaque:



Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

Tokens de acesso

Em suma, os tokens de acesso são usados para estabelecer confiança bidirecional entre um cliente e um servidor. Um token de bootstrap pode ser usado quando um cliente (por exemplo, um nó que está prestes a entrar no cluster) precisa confiar no servidor, com o qual ele está falando.



Anotações

Criar Token de acesso ao Cluster

1

Logue com o usuário suporte e verifique as opções do comando `kubeadm token`:

```
$ sudo kubeadm token --help
```

2

Crie um token de acesso ao cluster com tempo de vida de 1 minuto e execute o comando:

```
$ sudo kubeadm token create --print-join-command --ttl 1m --description="Teste"
```

3

Para listar os tokens de acesso disponível no Cluster, execute o comando:

```
$ sudo kubeadm token list
```



Kube Master

O que é um token de acesso mais exatamente?

- ✓ É um Segredo no namespace do sistema kube do tipo "bootstrap.kubernetes.io/token";
- ✓ Um token de bootstrap deve estar no formato "[a-z0-9] {6}. [A-z0-9] {16}". A parte anterior é o ID do token público, enquanto o último é o Token Secret e deve ser mantido privado em todas as circunstâncias!
- ✓ O nome do Segredo deve ser chamado de "bootstrap-token- (token-id);

Descrição dos comandos

- ✓ **kubeadm token create**: Criar um token de bootstrap no servidor;
- ✓ **--print-join-command**: Em vez de imprimir apenas o token, exibe o comando completo 'kubeadm join', necessário para ingressar um node no Cluster usando o token;
- ✓ **--ttl 1m**: A duração antes do token é automaticamente excluída (por exemplo, 1s, 2m, 3h). Se definido como '0', o token nunca expirará (padrão 24h0m0s);
- ✓ **kubeadm token list**: Lista os tokens de bootstrap no servidor.

Criar Token de acesso ao Cluster

1

Para armazenar somente o Token em uma variável, execute o seguinte comando:

```
$TK=$(sudo kubeadm token list | grep Teste | awk -F" " '{print $1}')
$echo $TK
```

2

Para remover o Token, execute o subcomando token delete e a variável:

```
$sudo kubeadm token delete $TK
$sudo kubeadm token list
```

3

Crie um token de acesso ao cluster com tempo de vida de 30 minutos, antes de continuar:

```
$sudo kubeadm token create --print-join-command --ttl 30m
$sudo kubeadm token list
```



Kube Master

Descrição dos comandos

- ✓ **kubeadmin token delete:** Remove um token de bootstrap no servidor.

Adicionar Nó no Cluster

1

Execute o comando somente nas máquinas Kube Node 1 e 2, para ingressar no Cluster:

```
$ sudo kubeadm join 200.100.50.100:6443 --token <TOKEN_GERADO> \
--discovery-token-ca-cert-hash <DISCOVERY_TOKEN_GERADO>
```

2

Na máquina Kube Master, verifique se os Kube Node 1 e 2 possuem o STATUS como Ready:

```
$ kubectl get nodes
```

3

Verifique se containers da rede Calico e Kube Proxy, estão sendo executados nos Nodes:

```
$ kubectl get pods -n kube-system
```



Kube Master, Kube Node 1 e 2

Ingressar novos Nodes no Cluster

- ✓ Ao ingressar um node em um cluster, precisamos estabelecer confiança bidirecional. Isso é dividido em descoberta (com o nó confiando no plano de controle do Kubernetes) e na inicialização do TLS (com o plano de controle do Kubernetes, confiando no nó);
- ✓ Existem dois esquemas principais para a descoberta. A primeira é usar um token compartilhado junto com o endereço IP do servidor da API. A segunda é fornecer um arquivo, um subconjunto do arquivo kubeconfig padrão. Este arquivo pode ser um arquivo local ou baixado por meio de um URL HTTPS.

Descrição dos comandos

- ✓ **kubeadm join**: Permite ingressar uma máquina a um Cluster Kubernetes;
- ✓ **--token string**: Define o token de descoberta e tls-bootstrap-token, quando esses valores não forem fornecidos;
- ✓ **--discovery-token-ca-cert-hash strings**: Para detecção baseada em token, valide se a chave pública da CA raiz corresponde a esse hash;
- ✓ **--ignore-preflight-errors strings**: Define uma lista de verificações, cujos erros serão mostrados como avisos;
- ✓ **kubectl get nodes**: Lista e exibe os estados de todas as máquinas que participam do cluster.

Obter informações sobre os Nós do Cluster

1

Para listar as máquinas que participam do Cluster Kubernetes, execute o seguinte comando:

```
$ kubectl get nodes
```

2

Para exibir informações detalhadas sobre uma máquina, execute:

```
$ kubectl describe node kube-node01.dexter.com.br
```

3

Repita o comando para exibir informações sobre a máquina Kube Node 2:

```
$ kubectl describe node kube-node02.dexter.com.br
```



Kube Master

Descrição dos comandos

- ✓ **kubectl describe node:** Exibe informações sobre uma máquina que participa do cluster.

Remover Nó no Cluster

1

Para remover uma máquina que participa do Cluster Kubernetes, execute:

```
$ kubectl delete node kube-node02.dexter.com.br
```

2

Para validar o comando anterior, liste as máquinas que participam do Cluster Kubernetes:

```
$ kubectl get nodes
```



Kube Master

Descrição dos comandos

- ✓ **kubectl delete node:** Remove uma máquina que participa do cluster.

Adicionar novamente Nó no Cluster

Execute novamente o comando na máquina Kube Node 2, para ingressar no Cluster:

```
$ sudo kubeadm join 200.100.50.100:6443 --token <TOKEN_GERADO> \
--discovery-token-ca-cert-hash <DISCOVERY_TOKEN_GERADO> \
--ignore-preflight-errors=all
```

Na máquina Kube Master, verifique se a máquina Kube Node 2 possui o STATUS Ready:




```
$ kubectl get nodes
```



Anotações

[illegible]

Recapitulando

-  Gerenciar Tokens no Cluster;
-  Adicionar nó do Cluster Kubernetes;
-  Remover nó do Cluster Kubernetes.

Anotações



Anotações

Atualizando o Cluster Kubernetes

Objetivos da Aula

- Atualizar o kubeadmin e kubelet no Master;
- Realizar o planejamento de atualização;
- Atualizar o kubectl em todos os Nós do Cluster.

Competências da Aula

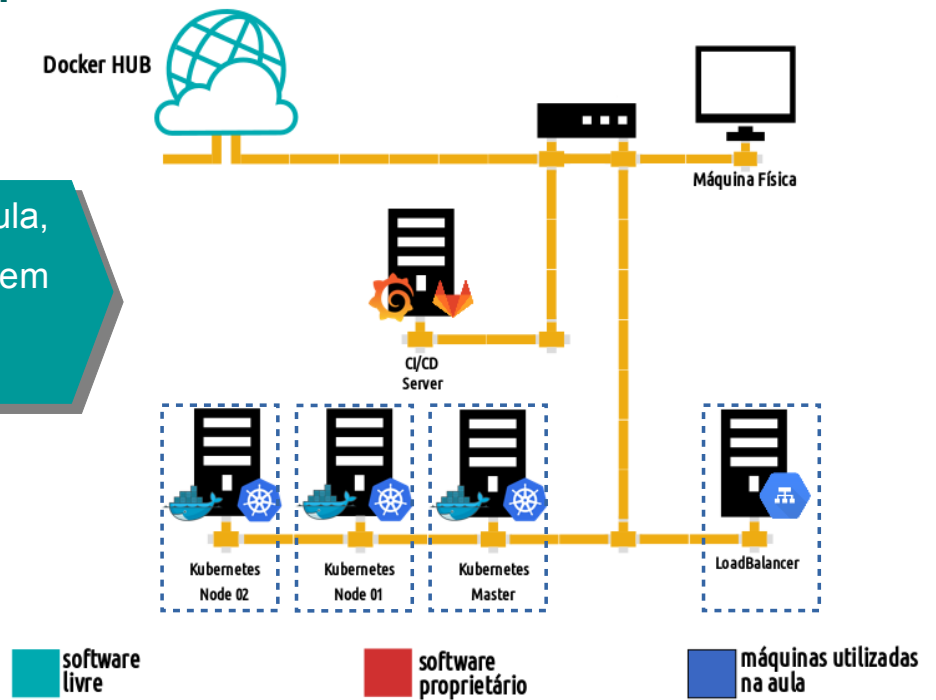
Esta aula possui conteúdo que atende aos objetivos da seguinte certificação:

Certified Kubernetes Administrator (CKA)

Installation, Configuration & Validation (13% of exam);



Para realizar esta aula,
ligue as máquinas em
destaque:



Anotações

[illegible]

Atualizar ferramentas e realizar plano de atualização

Anotações

Atualizar o kubeadmin no Master – ETAPA 1

- | | |
|---|---|
| 1 | Logue com usuário suporte e acesse a pasta tmp para realizar a atualização do Cluster:
<code>\$ cd /tmp</code> |
| 2 | Armazene em variáveis a última versão das ferramentas do Cluster e a arquitetura:
<code>\$ export VERSION=v1.15.6</code>
<code>\$ export ARCH=amd64</code> |
| 3 | Realize o download da ferramenta kubeadm, utilizando as informações das variáveis:
<code>\$ curl -sSL https://dl.k8s.io/release/\${VERSION}/bin/linux/\${ARCH}/kubeadm > kubeadm</code> |



Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

Atualizar o kubeadmin no Master – ETAPA 2

4

Instale o kubeadm na pasta /usr/bin com permissão 0775, usuário e grupo do root:

```
$sudo install -o root -g root -m 0755 ./kubeadm /usr/bin/kubeadm
```

5

Teste a atualização da ferramenta, exibindo a sua versão:

```
$sudo kubeadm version
```



Kube Master

Anotações

Realizar planejamento de atualização do cluster Kubernetes

1

Para realizar o plano de atualização do cluster, execute o seguinte comando:

```
$ sudo kubeadm upgrade plan
```

```
COMPONENT  CURRENT  AVAILABLE
Kubelet    3 x v1.15.1  v1.15.6
```

2

A partir do resultado do planejamento, atualize o cluster informando o número da release:

```
$ sudo kubeadm upgrade apply v1.15.6
```

```
[upgrade/versions] Cluster version: v1.15.6
```

```
[upgrade/versions] kubeadm version: v1.15.6
```

```
[upgrade/confirm] Are you sure you want to proceed with the upgrade? [y/N]:y
```



Kube Master

Descrição dos comandos

- ✓ **kubeadm upgrade plan:** Verifica quais versões estão disponíveis para atualizar e validar se seu cluster atual é atualizável;
- ✓ **kubeadm upgrade apply:** Atualiza o cluster do Kubernetes para a versão especificada.

Atualizar o kubelet no Master

1

Realize o download da ferramenta kubelet, utilizando as informações das variáveis:

```
$ curl -sSL https://dl.k8s.io/release/${VERSION}/bin/linux/${ARCH}/kubelet > kubelet
```

2

Instale o kubelet na pasta /usr/bin com permissão 0775, usuário e grupo do root:

```
$ sudo install -o root -g root -m 0755 ./kubelet /usr/bin/kubelet
```

3

Reinicie o serviço do kubelet para aplicar as configurações:

```
$ sudo systemctl restart kubelet
```



Kube Master

Anotações

Atualizar o kubectl no Master

Realize o download da ferramenta kubectl, utilizando as informações das variáveis:

```
$ curl -sSL https://dl.k8s.io/release/${VERSION}/bin/linux/${ARCH}/kubectl > kubectl
```

Instale o kubectl na pasta /usr/bin com permissão 0775, usuário e grupo do root:

```
$ sudo install -o root -g root -m 0755 ./kubectl /usr/bin/kubectl
$
```



Anotações

Atualizar kubectl em todos os nós do Cluster

Anotações

Atualizar o kubectl em todos os Nodes do Cluster

Armazene em variáveis a última versão das ferramentas do Cluster e a arquitetura:

```
$ export ARCH=amd64
```

Realize o download da ferramenta kubectl, utilizando as informações das variáveis:

```
$ curl -sSL https://dl.k8s.io/release/${VERSION}/bin/linux/${ARCH}/kubectl > kubectl
```

Instale o kubectl na pasta /usr/bin com permissão 0775, usuário e grupo do root:

```
$ sudo install -o root -g root -m 0755 ./kubectl /usr/bin/kubectl
```



Anotações

Atualizar o kubelet em todos os Nodes do Cluster

- | | |
|---|---|
| 1 | <pre>\$ curl -sSL https://dl.k8s.io/release/\${VERSION}/bin/linux/\${ARCH}/kubelet > kubelet</pre> |
| 2 | <p>Instale o kubelet na pasta /usr/bin com permissão 0775, usuário e grupo do root:</p> <pre>\$ sudo install -o root -g root -m 0755 ./kubelet /usr/bin/kubelet</pre> |
| 3 | <p>Reinicie o serviço do kubelet para aplicar as configurações:</p> <pre>\$ sudo systemctl restart kubelet</pre> |



Anotações

Testar se a atualização foi bem sucedida

Liste as máquinas que participam do Cluster Kubernetes, após a atualização:

```
$ kubectl get nodes
```

Verifique se os Pods estão em execução no Cluster, após a atualização:




```
$ kubectl get pods -n kube-system
```



Anotações

[illegible]

Recapitulando

-  Atualizar o kubeadmin e kubelet no Master;
-  Realizar o planejamento de atualização;
-  Atualizar o kubectl em todos os Nós do Cluster.

Anotações



Anotações

Objetivos da Aula

- Gerenciar Pods;
- Gerenciar Deployments.

Competências da Aula

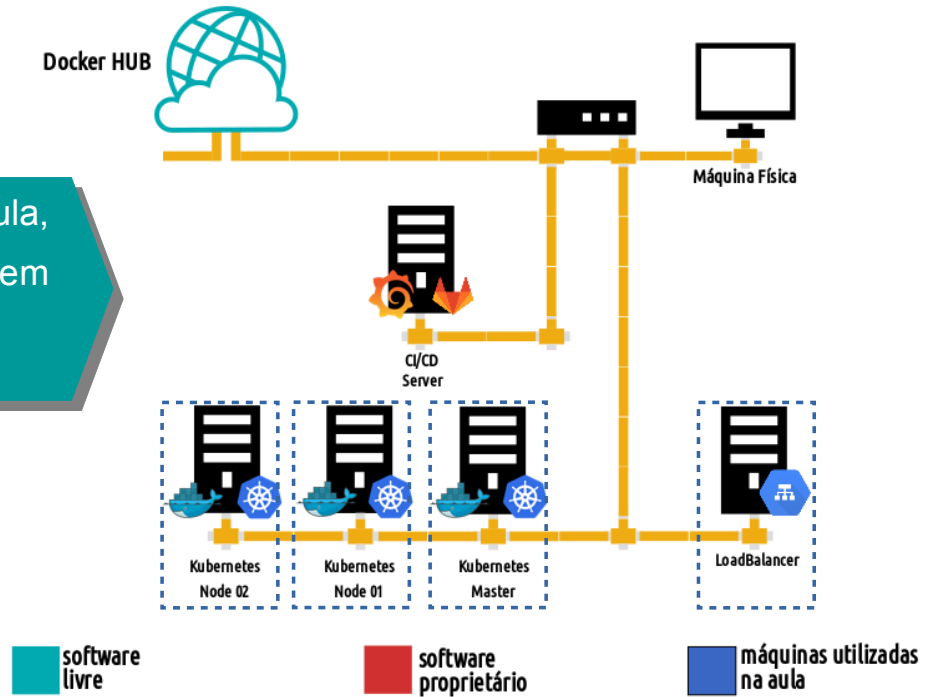
Esta aula possui conteúdo que atende aos objetivos da seguinte certificação:

Certified Kubernetes Administrator (CKA)

Installation, Configuration & Validation (13% of exam);



Para realizar esta aula,
ligue as máquinas em
destaque:



Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

Gerenciar Pods

Anotações

Executar Pods

1

Logue com o usuário suporte e liste os Pods presentes no Cluster:

```
$ kubectl get pods
```

2

Verifique as opções de comando para criar um Pod com a imagem nginx:

```
$ kubectl run --help
```

```
$ kubectl run nginx --generator=run-pod/v1 --image=nginx
```

3

Verifique em qual Node o Pod está em execução e também a descrição do mesmo:

```
$ kubectl get pods -o wide
```

```
$ kubectl describe pod nginx
```



Kube Master

Descrição dos comandos

- ✓ **kubectl run**: Crie e execute uma imagem específica, possivelmente replicada;
- ✓ **--generator=run-pod/v1**: O nome do gerador de API a ser utilizado;
- ✓ **--image**: O nome da imagem Docker;
- ✓ **kubectl get pods -o wide**: Lista todos os pods no formato de saída ps com mais informações (como nome do nó);
- ✓ **kubectl describe pod**: Exibe informações sobre um Pod.

Permitir acesso e executar comandos em Pods

1

Para encaminhar uma porta para acesso local de um Pod, execute o seguinte comando:

```
$ kubectl port-forward nginx 8081:80 &
<TECLE ENTER>
```

2

Teste o acesso à aplicação na porta 8081 no endereço local da máquina:

```
$ curl http://127.0.0.1:8081
```

3

Para executar um comando no Pod, utilize o comando kubectl exec:

```
$ kubectl exec -it nginx -- uname -r
$ kubectl exec -it nginx -- nginx -v
```



Kube Master

Descrição dos comandos

- ✓ **kubectl port-forward**: Encaminha uma ou mais portas locais para um pod;
- ✓ **Kubectl exec**: Executa comando em um Pod.

Permitir acesso externo ao Pod

4

Traga para o primeiro plano e cancele com CTRL + C o comando `kubectl port-forward`:

```
$fg
CTRL + C
```

5

Para encaminhar uma porta para acesso remoto de um Pod, execute o seguinte comando:

```
$kubectl port-forward --address 200.100.50.100 nginx 8081:80 &
```

6

Teste o acesso à aplicação na porta 8081 no endereço remoto da máquina:

```
$curl http://200.100.50.100:8081
```



Kube Master

Descrição dos comandos

- ✓ **`kubectl port-forward --address`**: Encaminha uma ou mais portas em um determinado endereço.

Deletar um Pod no Cluster

7

Traga para o primeiro plano e cancele com CTRL + C o comando kubectl port-forward:
\$fg
 CTRL + C

8

Para remover um Pod do cluster, execute o seguinte comando:
\$ kubectl delete pod nginx

9

Verifique se os Pods foram removidos do Cluster:
\$ kubectl get pods



Kube Master

Descrição dos comandos

- ✓ **kubectl delete pod**: Remove um Pod do cluster.

Gerenciar Deployments

Anotações

Criar Deployments no Cluster

1

Antes de criar um Deploy no Cluster, verifique as opções do comando:
\$ kubectl create deployment --help

2

Crie um Deploy no Cluster, através do seguinte comando:
\$ kubectl create deployment nginx --image=nginx

3

Liste os Deploy no Cluster e também a descrição do mesmo:
\$ kubectl get deployments
\$ kubectl describe deployment nginx



Kube Master

Descrição dos comandos

- ✓ **kubectl create**: Cria e executa um Deploy;
- ✓ **--image**: O nome da imagem Docker;
- ✓ **kubectl describe deployments**: Exibe informações sobre um Deploy.

Escalar aplicações e remover deployments no Cluster

4

Liste os Pods presentes no Cluster para validar a criação do Deploy:

```
$ kubectl get pods
```

5

Para criar 2 réplicas do Pod no Deploy, execute os seguintes comandos:

```
$ kubectl scale deployment/nginx --replicas=2
```

```
$ kubectl get pods
```

6

Para deletar e listar um Deploy no cluster, execute os seguintes comandos:

```
$ kubectl delete deployment nginx
```

```
$ kubectl get deployments
```



Kube Master

Descrição dos comandos

✓ **kubectl scale**: Escala um replicaset em um Deploy.

Recapitulando



Gerenciar Pods;



Gerenciar Deployments.

Anotações



Anotações

Objetivos da Aula

- Realizar backup do cluster Kubernetes;
- Remover nós e resetar a configuração do Cluster;
- Realizar restore do cluster Kubernetes.

Competências da Aula

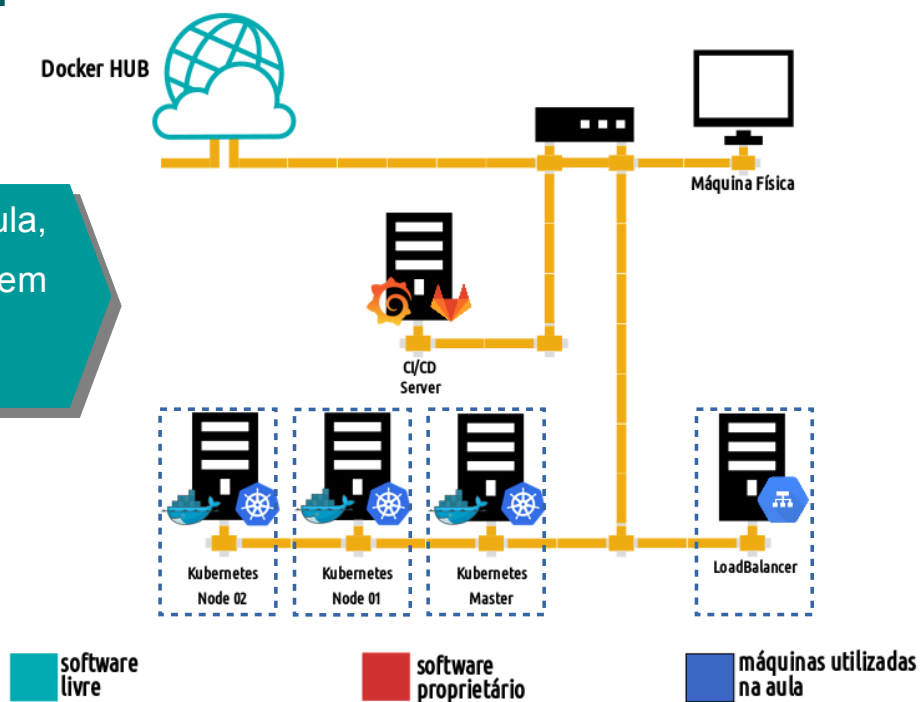
Esta aula possui conteúdo que atende aos objetivos da seguinte certificação:

Certified Kubernetes Administrator (CKA)

Installation, Configuration & Validation (13% of exam);



Para realizar esta aula,
ligue as máquinas em
destaque:



Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

Realizar backup do cluster Kubernetes

Anotações

Tipos de configuração do Cluster

Cluster interno do etcd

Cluster do etcd externo

✓ Significa que você está executando seu cluster do etcd na forma de contêineres / conjuntos, dentro do cluster do Kubernetes e é responsabilidade do Kubernetes gerenciar esses pods.

Anotações

Tipos de configuração do Cluster

Cluster interno do etcd

Cluster do etcd externo

- ✓ O cluster do Etcd que você está executando fora do cluster do Kubernetes, principalmente na forma de serviços do Linux e fornecendo seus pontos de extremidade ao cluster do Kubernetes para gravar.

Anotações

Instalação da ferramenta etcdctl

1

Logue com o usuário suporte e alterne para a conta do usuário root:

```
$ sudo su -
```

2

Descompacte o arquivo etcd, através do comando TAR:

```
# ls
# tar xvf etcd-v3.3.13-linux-amd64.tar.gz
```

3

Mova os binários do etcd para o diretório /usr/local/bin e declare a versão do API:

```
# mv etcd-v3.3.13-linux-amd64/etcd* /usr/local/bin/
# export ETCDCTL_API=3
```



Kube Master

Ferramenta etcdctl

O etcdctl é um cliente de linha de comando para o etcd. Certifique-se de definir a variável de ambiente ETCDCTL_API=3.

Para obter o arquivo utilizado em nosso exemplo, acesse o seguinte link:

<https://github.com/coreos/etcd/releases/download/v3.3.13/etcd-v3.3.13-linux-amd64.tar.gz>

Realizar backup com a ferramenta etcdctl

1

Verifique as opções de comando da ferramenta etcdctl com o subcomando snapshot save:
etcdctl snapshot save --help

2

Grave um snapshot pontual do banco de dados backend do etcd em um arquivo:
**# etcdctl snapshot save snapshot.db **
**--cacert=/etc/kubernetes/pki/etcd/ca.crt **
**--cert=/etc/kubernetes/pki/etcd/healthcheck-client.crt **
--key=/etc/kubernetes/pki/etcd/healthcheck-client.key

3

Liste as informações do snapshot que estão armazenadas no arquivo snapshot.db:
etcdctl --write-out=table snapshot status snapshot.db



Kube Master

Descrição dos comandos

- ✓ **etcdctl snapshot save**: Grava um snapshot pontual do banco de dados backend do etcd em um arquivo;
- ✓ **--cacert**: Arquivo de autoridade de certificação TLS para cluster de destino;
- ✓ **--cert**: Arquivo de certificado TLS para cluster de destino;
- ✓ **--key**: Arquivo de chave TLS para cluster de destino;
- ✓ **etcdctl --write-out=table snapshot status**: Lista informações sobre um determinado arquivo instantâneo de banco de dados de back-end.

Realizar backup de configurações do cluster

1

Crie um diretório para armazenar as configurações do Cluster:

```
#mkdir -p /data/backup
```

2

Realize um backup do diretório /etc/kubernetes para a pasta criada no comando anterior:

```
#cp -r /etc/kubernetes /data/backup/kubernetes.bak
```



Kube Master

Anotações

Remover nós e resetar configuração do Cluster

Anotações

Remover Nós do Cluster

Retorne a conta do usuário suporte e liste os Nodes presentes no Cluster:

\$ kubectl get nodes

Remova os Nodes kube-node 01 e 02, através dos seguintes comandos:

```
$ kubectl delete node kube-node02.dexter.com.br
```

Confirme se os Nodes foram removidos, listando os mesmos:

\$ kubectl get nodes



Anotações

Resetar configuração e limpar regras de Firewall do Cluster

1

Altere para a conta do usuário root e reverta as alterações feitas no Cluster:

```
$ sudo su -
```

```
# kubeadm reset
```

```
[reset] Are you sure you want to proceed? [y/N]: y <TECLE ENTER>
```

2

Em seguida, limpe todas as regras de Firewall através dos seguintes comandos:

```
# iptables -F
```

```
# iptables -t nat -F
```

```
# iptables -t mangle -F
```

```
# iptables -X
```



Kube Master

Descrição dos comandos

- ✓ **kubeadm reset**: Permite reverter quaisquer alterações feitas no host pelos comandos 'kubeadm init' ou 'kubeadm join';
- ✓ **iptables -F**: Exclui todas as regras em cadeia da tabela Filter;
- ✓ **iptables -t nat -F**: Exclui todas as regras em cadeia da tabela NAT;
- ✓ **iptables -t mangle -F**: Exclui todas as regras em cadeia da tabela Mangle;
- ✓ **iptables -X**: Exclui uma cadeia definida pelo usuário.

Realizar restore do cluster Kubernetes

Anotações

Realizar restore do Cluster – ETAPA 1

1

Antes de restaurar os arquivos de backup, remova o conteúdo de /etc/kubernetes:

```
#rm -rf /etc/kubernetes/*
```

2

Restaure os arquivos de configuração, localizados na pasta /data/backup/kubernetes.bak:

```
#cp -r /data/backup/kubernetes.bak/* /etc/kubernetes/
```

3

Restaure o banco de dados, através do subcomando snapshot restore:

```
#export ETCDCTL_API=3
```

```
#etcdctl snapshot restore snapshot.db
```



Kube Master

Descrição dos comandos

- ✓ **etcdctl snapshot restore:** Cria um diretório de dados do etcd para um membro de cluster do etcd, a partir de um snapshot de banco de dados de backend e uma nova configuração de cluster.

Realizar restore do Cluster – ETAPA 2

```
#rm -rf /var/lib/etcd/*
#mv default.etcd/member /var/lib/etcd/
```

```
#systemctl restart kubelet
#kubeadm init --ignore-preflight-errors=all
```




```
#exit
$ kubectl get nodes,pods -n kube-system
```



Anotações

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

Recapitulando

-  Realizar backup do cluster Kubernetes;
-  Remover nós e resetar a configuração do Cluster;
-  Realizar restore do cluster Kubernetes.

Anotações
