
DEVOPS
DIA 9 DE 10

Augusto Nunes

PSM, MCT, OCA, OCP, MCSD

augustonunes

Augusto Nunes

augusto.cbn


KUBERNETES

KUBERNETES – SUMÁRIO

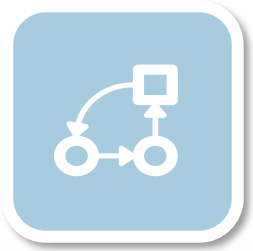
- Visão Geral
- Auto-Hospedagem
- Serviços Gerenciados e Ferramentas
- Soluções Turnkey
- Instaladores
- Principais Comandos
- Laboratórios

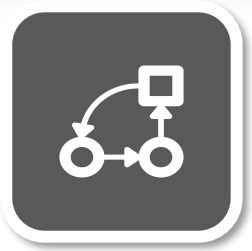
KUBERNETES

VISÃO GERAL

KUBERNETES - VISÃO GERAL

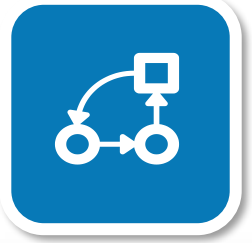
Sistema operacional nativo de nuvem.

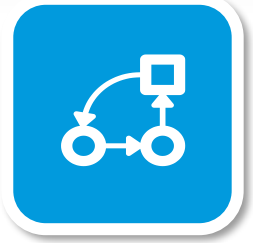




Plataforma confiável e escalável para executar cargas de trabalho em contêineres.

Arquitetura em cluster.

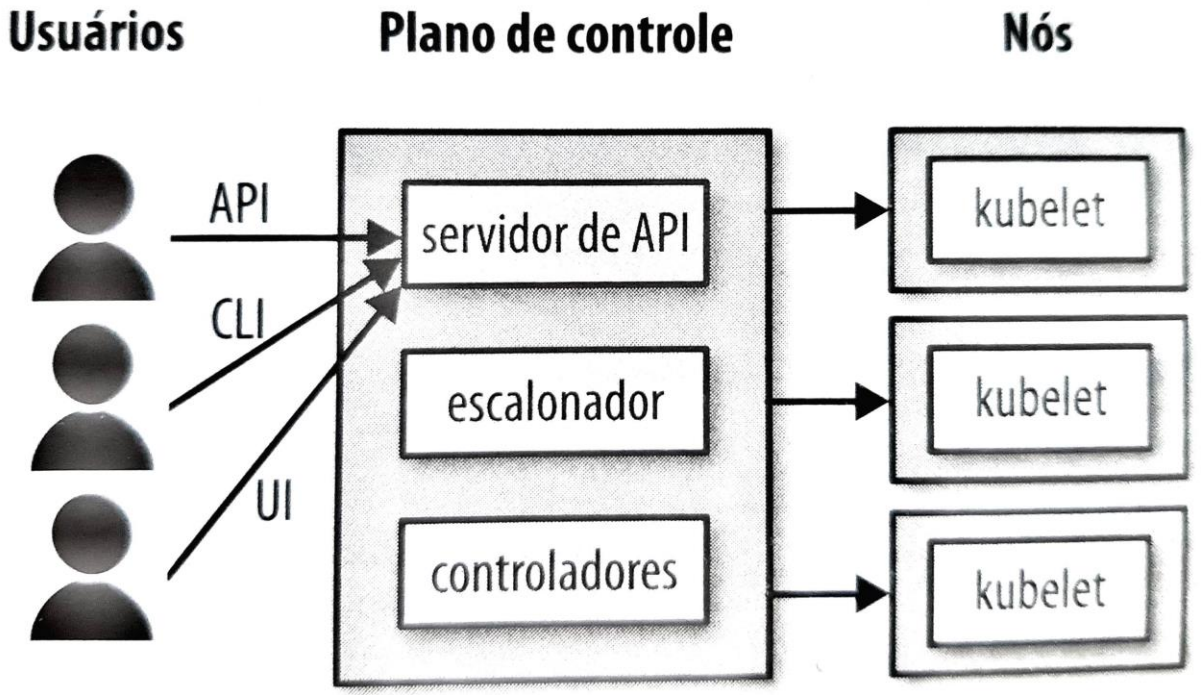




Auto-hospedagem ou serviço gerenciado.

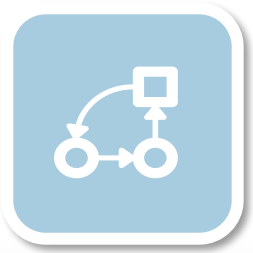
KUBERNETES – PLANO DE CONTROLE

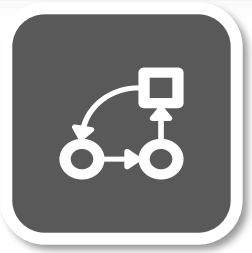
O plano de controle é a principal peça do Kubernetes. Ele executa todas as tarefas necessárias a realização dos trabalhos: escalona contêineres, administra serviços e atende requisições de API.



KUBERNETES – PLANO DE CONTROLE

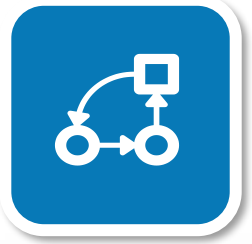
Kube-apiserver: servidor de frontend do plano de controle, trata requisições destinadas a API.

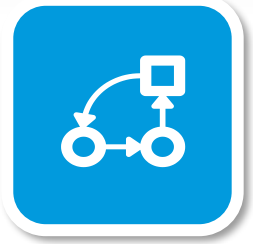




Etcd: banco de dados do Kubernetes. Armazena informações sobre os nós, recursos do cluster, etc.

Kube-scheduler: decide onde executar os Pods criados.



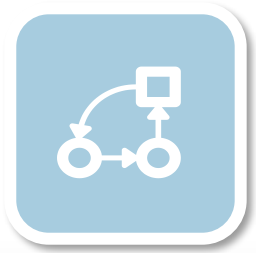


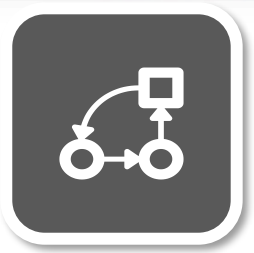
Kube-controller-manager: responsável por executar os controladores de recurso (ex.: deployments).

Cloud-controller-manager: Interage com o provedor de nuvem.

KUBERNETES – PLANO DE CONTROLE

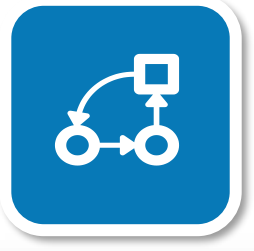
Deve possuir vários nós mestres, a fim de, garantir alta disponibilidades.

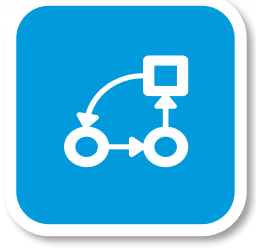




O banco de dados é replicado entre os vários nós, garantindo sobrevida em caso de falhas.

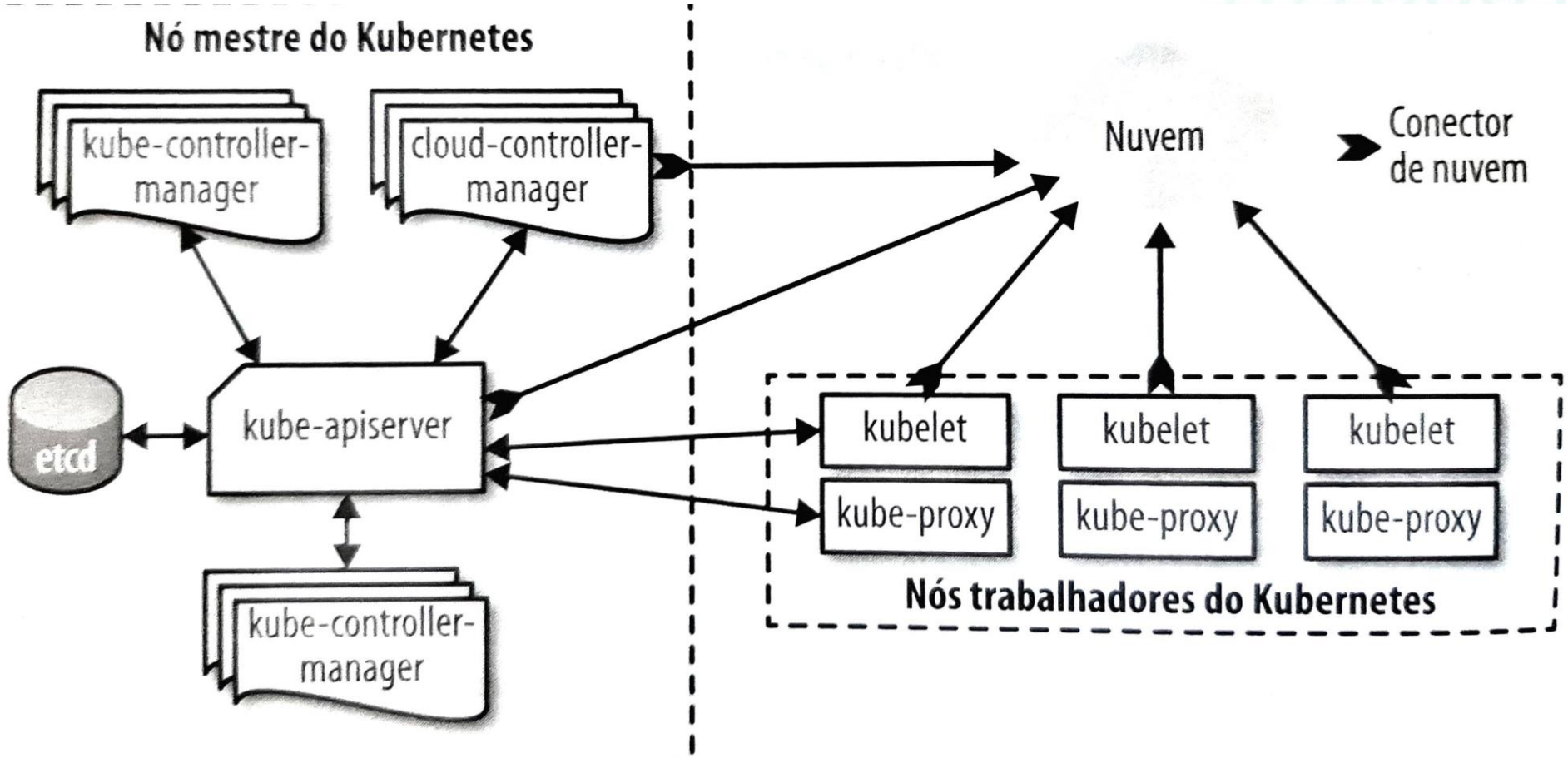
Uma falha geral nos nós mestres pode interromper a execução do cluster.





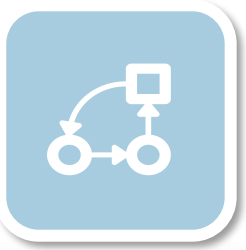
Falhas nos nós trabalhores serão tratadas pelo Kuberntes, escalonando os Pods em nós saudáveis.

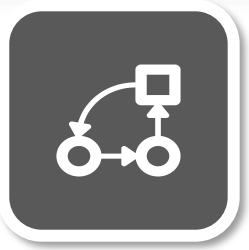
KUBERNETES – COMPONENTES DE UM NÓ



KUBERNETES – ESTRUTURA DOS NÓS

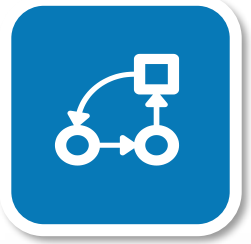
Kubelet: responsável por orientar o contêiner a iniciar cargas de trabalho escalonadas no nó e monitorar o status.

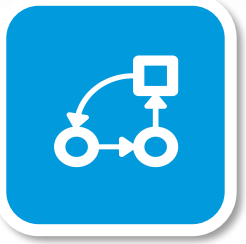




Runtime do contêiner: Inicia e finaliza os contêineres, além de tratar de suas comunicações.

Kube-proxy: Roteia as requisições entre os Pods e nós distintos, e também entre os Pods e internet.

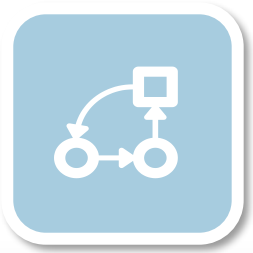


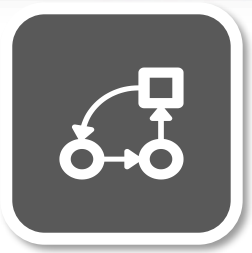


Nós mestres e nós trabalhadores.

KUBERNETES – DEPLOYMENTS

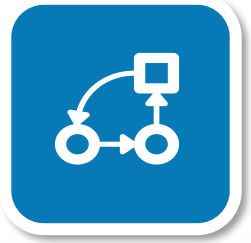
Programa supervisor que verifica, de forma recorrente, se o contêiner está executando.

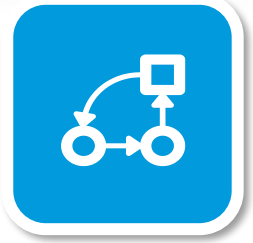




Em caso de paradas, ele reinicia imediatamente o contêiner.

Responsável por iniciar ou parar a execução de um contêiner.

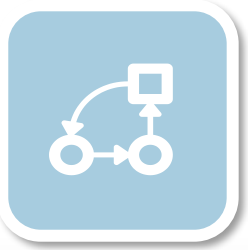


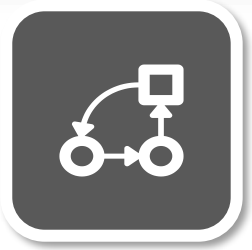


Responsável pela criação de controladores e gestores de replicas (ReplicaSets).

KUBERNETES – PODS

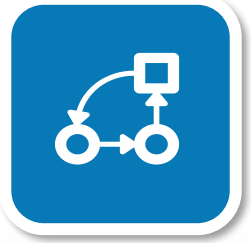
Objeto Kubernetes que representa um grupo de um ou mais contêineres.

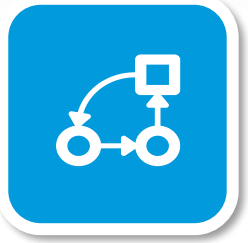




Garantia de que um grupo de contêineres será executado em um mesmo nó.

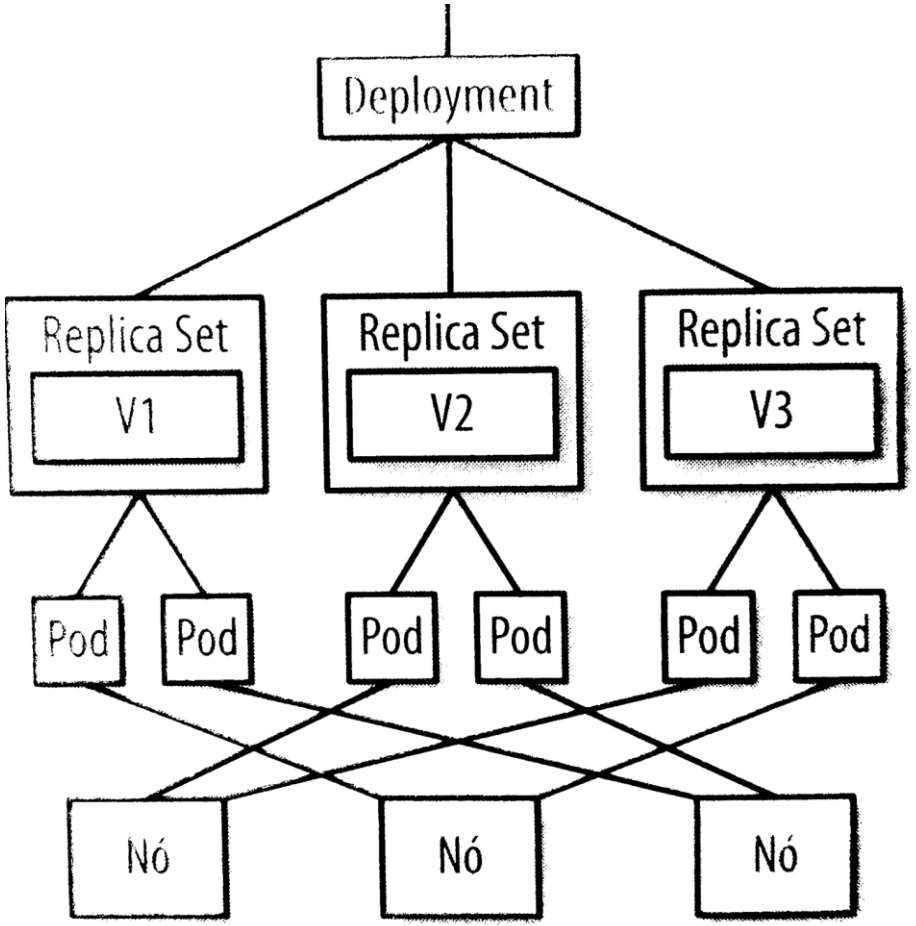
Facilita o escalonamento de grupos de contêineres.





Em diversos cenários o Pod conterá apenas um contêiner.

KUBERNETES – REPLICASET

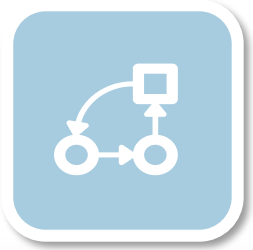


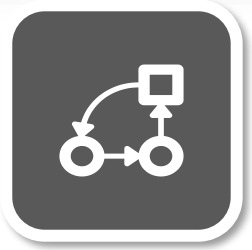
KUBERNETES

AUTO-HOSPEDAGEM

KUBERNETES – AUTO-HOSPEDAGEM

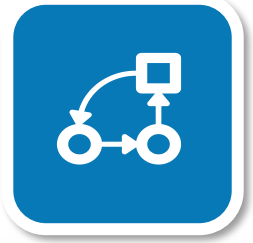
Máximo de flexibilidade e controle..





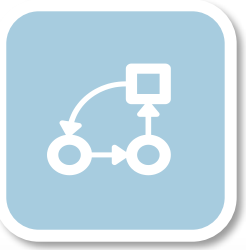
Equipe dedicada para instalação e manutenção da ferramenta.

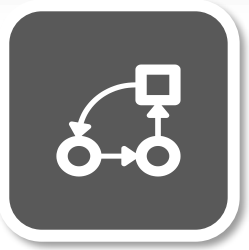
Controle total sobre atividades de upgrade.



KUBERNETES – AUTO-HOSPEDAGEM - DESAFIOS

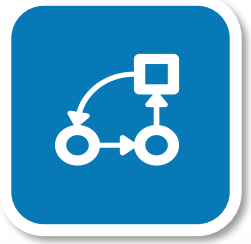
O plano de controle foi configurado para alta disponibilidade.

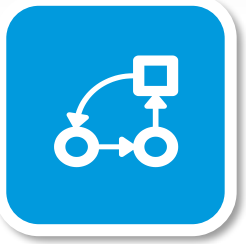




Os nós trabalhadores foram configurados para alta disponibilidade? Organizados geograficamente?

O cluster está seguro? A comunicação interna ocorre via canais seguros? Os contêineres possuem as permissões adequadas?

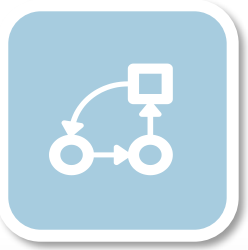


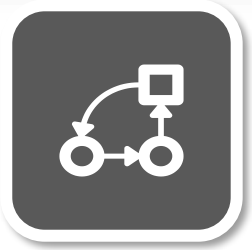


O acesso a base de dados (etcd) é autenticado e monitorado?

KUBERNETES – AUTO-HOSPEDAGEM - DESAFIOS

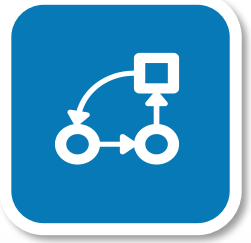
Todos os serviços do cluster estão seguros? Os acessos externos são autenticados e autorizados?

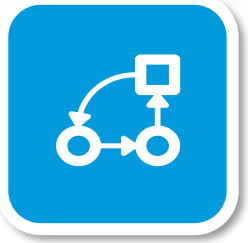




O acesso a API do Kubernetes foi devidamente limitado?

O cluster está em compliance? Atende a Cloud Native Computing Foudation?

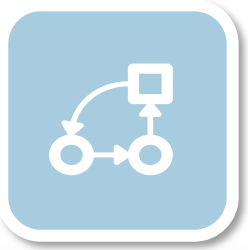


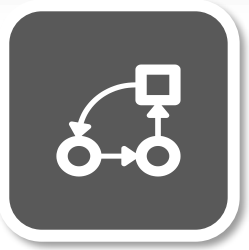


Manter o sistema operacional e o Kubernetes atualizados em cada um dos nós.

KUBERNETES – AUTO-HOSPEDAGEM - DESAFIOS

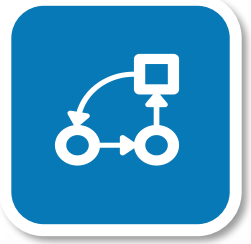
Os backups das bases de dados são feitos corretamente? Armazenados em local adequado? A restauração é testada com frequência?

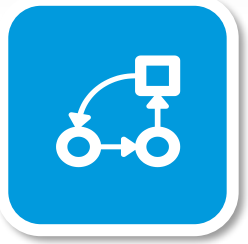




Como o cluster será mantido? Como novos nós serão incrementados?

Como o cluster será escalado diante de uma demanda crescente?





Como será feito o rollout de atualizações entre os nós?

KUBERNETES – AUTO-HOSPEDAGEM - DESAFIOS

Cindy Sridharan
@copyconstruct

At a previous job, we decided against doing K8s. A disgruntled engineer spent the entirety of their tenure being grumpy about this decision.

It takes well over a million dollars just in engineer salary to get K8s up and running from scratch. And you still might not get there.

Traduzir Tweet

> **K10g** @KarlKFI · 21 de jul de 2018
> Em resposta a @ibuildthecloud e @drnic
> Even with kubeadm and a team of 6 you can't get upstream K8s production ready in 6 months. We tried.
>
> Hard Problems:
> - High Availability
> - Single Sign On
> - Multitennancy
> - Resource Isolation
> - Permission Management
> - Upgrades
> - Backups
> - Package Management
> - CI/CD Integration

12:57 AM · 22 de jul de 2018 · Twitter for iPhone

KUBERNETES – AUTO-HOSPEDAGEM - DESAFIOS

Cindy Sridharan is a distributed systems engineer. She's the author of a book on Distributed Systems Observability with O'Reilly and the co-author of an upcoming book on distributed systems engineering in the cloud. She runs the Prometheus user group in San Francisco, has been a reviewer of several technical books and on the program committee of leading industry conferences on systems engineering. She lives in San Francisco and in her spare time enjoys hiking the gorgeous outdoors of the Bay Area, reading way too many papers and occasionally blogging about building resilient and maintainable systems.

Find Cindy Sridharan at

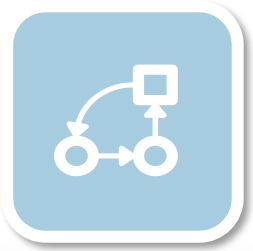
@copyconstruct

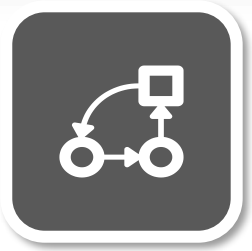
KUBERNETES

PRINCIPAIS FERRAMENTAS

KUBERNETES – GOOGLE KUBERNETES ENGINE (GKE)

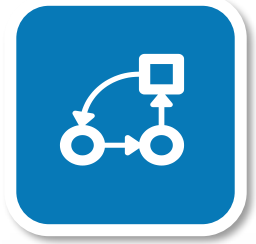
Possibilita a criação de clusters com poucos cliques.

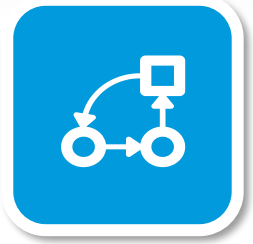




Plano de controle pronto. Garantia de alta disponibilidade.

Canais de lançamento, reparos automáticos e upgrade automático.

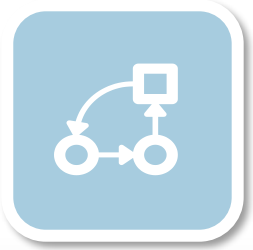


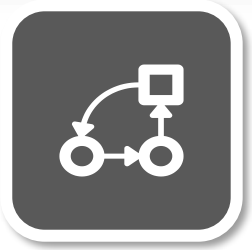


Seguro por padrão (verificação de vulnerabilidades e criptografia de dados). Monitoramento completo.

KUBERNETES – AMAZON FOR KUBERNETES (EKS)

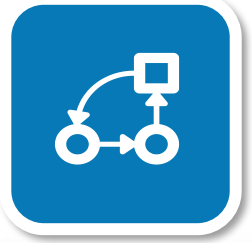
Experiência limitada em relação ao GKE.

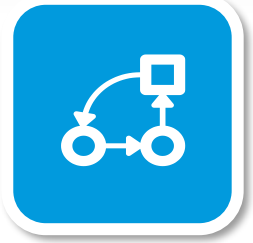




Esforços adicionais para configuração do serviço.

Cobrança pelos nós mestres e também pelo restante da infraestrutura.

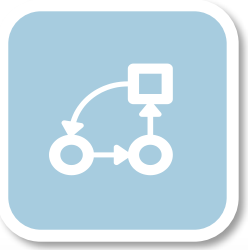


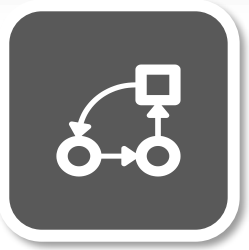


Geralmente mais caro que o serviço similar da Google e da Microsoft.

KUBERNETES – SERVIÇO DE KUBERNETES DO AZURE (AKS)

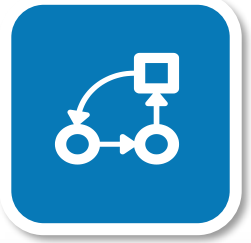
Provisionamento elástico, sem a necessidade de gerenciar a infraestrutura.

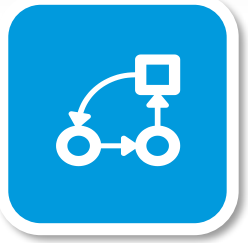




Monitramento visual e CLI.

Segurança aprimorada com o Azure Active Directory.

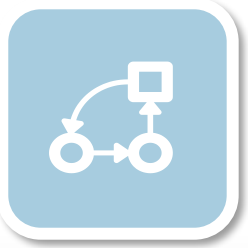


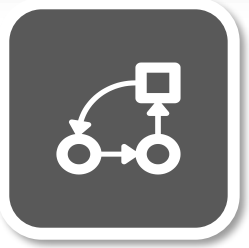


Ferramental para os desenvolvedores.

KUBERNETES – OPENSHIFT

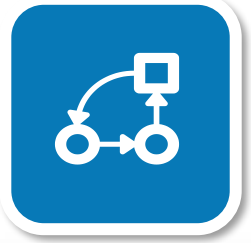
Serviços de gerenciamento Kubernetes, entregue na modalidade PaaS.

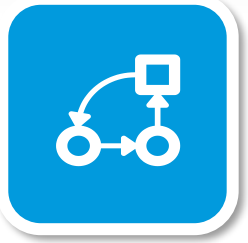




Features de ALM.

Suite de testes.

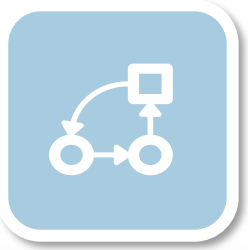


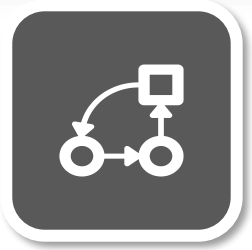


Funciona em ambientes heterogêneos : Nuvem pública, privada, máquinas virtuais, etc.

KUBERNETES – IBM CLOUD KUBERNETES SERVICE

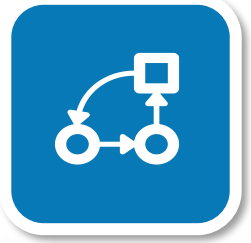
Simples e organizado. Ideal para construção de clusters mais simples.





Manutenção do cluster por CLI e interface gráfica.

Interessante apenas se a organização já estiver na cloud da IBM.

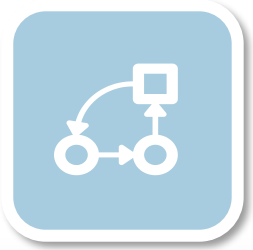


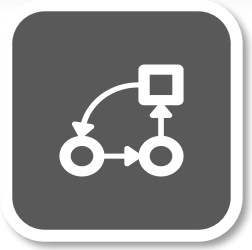
KUBERNETES

SOLUÇÕES TURNKEY

KUBERNETES – STACKPOINT

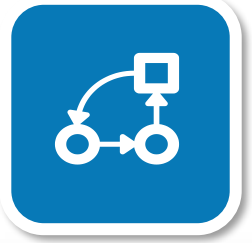
Poucos cliques para implantar um cluster Kubernetes em uma nuvem pública.

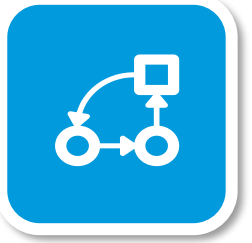




Suporte a grandes marcas: AWS, Google, Microsoft, etc.

Portal web de gerenciamento.

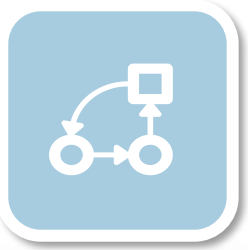


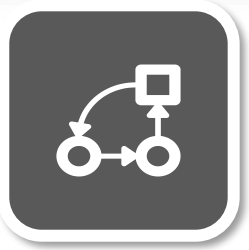


Planos a partir de 50 dólares por mês.

KUBERNETES – CONTAINERSHIP KUBERNETES ENGINE (CKE)

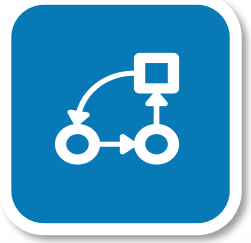
Suporte a nuvens públicas e privadas e ambientes on-premisse.

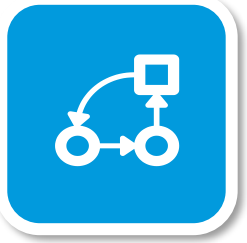




Configurações padrões e recomendadas para o plano de controle.

Flexível, pois, suporta a customização do cluster.





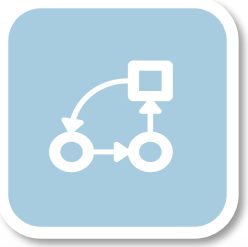
Provisionamento de um novo cluster através de poucos cliques.

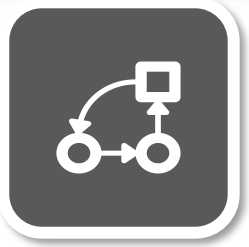
KUBERNETES

INSTALADORES

KUBERNETES – INSTALADORES - KOPS

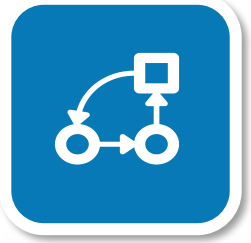
Ferramenta para provisionamento automático de clusters Kubernetes.

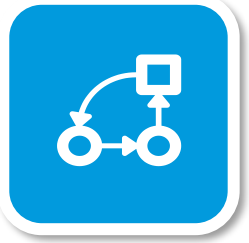




Suporte para AWS e GC.

Suporte a construção de clusters para alta disponibilidade.

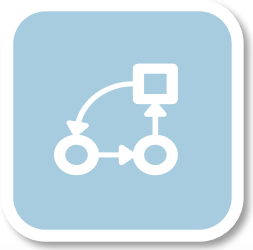


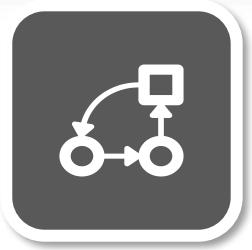


Ferramenta madura e sofisticada. Faz parte do projeto Kubernetes.

KUBERNETES – INSTALADRES - KUBESPRAY

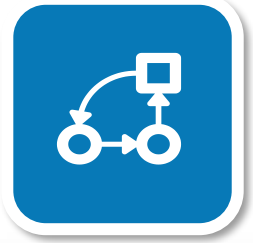
O projeto faz parte da família Kubernetes.

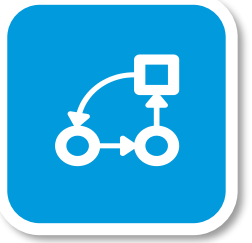




Suporte a servidores on-premisse.

Suporte e serviços de nuvem.

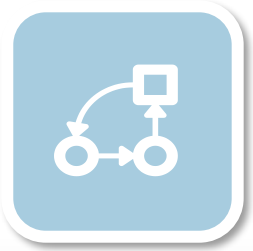


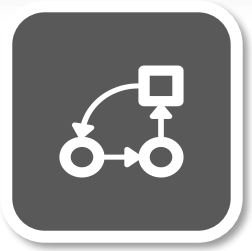


Configura clusters de alta disponibilidade.

KUBERNETES – INSTALADORES – TK8

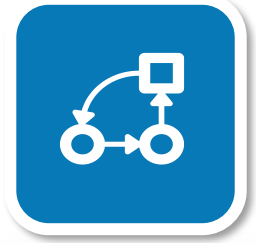
Ferramenta de linha de comando. Faz uso do Terraform e Kubespray.





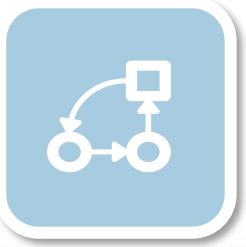
Suporte para AWS, Azure e GC.

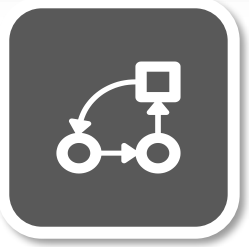
Instala plugins opcionais para manutenção do ALM.



KUBERNETES – INSTALADORES - KUBEADM

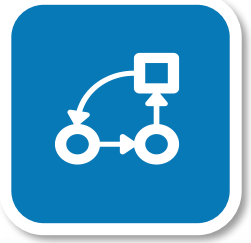
Ferramenta integrada a distribuição do Kubernetes.

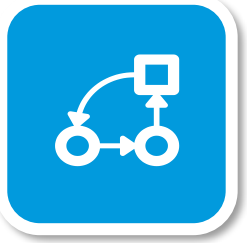




Suporte a instalação, a fim de, garantir que as melhores práticas sejam adotadas.

Não faz o provisionamento da infraestrutura do cluster.





Conformidade com o Kubernetes Conformance Test.

KUBERNETES

PRINCIPAIS COMANDOS

KUBERNETES – PRINCIPAIS COMANDOS – CREATE DEPLOYMENT

```
c:> kubectl create deployment [nome] --image [imagem docker]
```

--image nome da imagem a ser executada

Cria um deployment e consequentemente instância o pod relacionado.

KUBERNETES – PRINCIPAIS COMANDOS – LOGS

```
c:> kubectl logs [nome do pod]
```

Exibe o log de uma instanciação.

KUBERNETES – PRINCIPAIS COMANDOS – PORT FORWARD

```
c:> kubectl port-forward type/name 0000:9999
```

type deploy, pod, service, etc.
0000 porta do host
9999 porta exposta no container

Realiza o encaminhamento de portas, entre o host e o container.

KUBERNETES – PRINCIPAIS COMANDOS – GET NODES

```
c:> kubectl get nodes
```

Exibe a lista de nós vinculados ao cluster Kubernetes.

KUBERNETES – PRINCIPAIS COMANDOS – GET DEPLOYMENTS

```
c:> kubectl get deployments
c:> kubectl get deployment [nome do deployment] --output yaml
```

Lista os deployments ativos no namespace corrente.

KUBERNETES – PRINCIPAIS COMANDOS – DESCRIBE

```
c:> kubectl describe [tipo do recurso]/[nome do recurso]
```

Exibe os detalhes de um recurso específico do Kubernetes.

KUBERNETES – PRINCIPAIS COMANDOS – GET PODS

```
c:> kubectl get pods
c:> kubectl get pods --selector app=[nome do deployment]
```

Obtém os pods instânciados para determinado deployment.

KUBERNETES – PRINCIPAIS COMANDOS – DELETE PODS

```
c:> kubectl delete pods --selector app=[nome do deployment]
c:> kubectl delete all --selector app=[nome do deployment]
```

all remove o deployment e as réplicas.

Desativa os pods do deployment.

KUBERNETES – PRINCIPAIS COMANDOS – DELETE DEPLOYMENT

```
c:> kubectl delete deployment [nome do deployment]
```

Remove o deployment parametrizado.

KUBERNETES – PRINCIPAIS COMANDOS – SET IMAGE

```
c:> kubectl set image [tipo do serviço]/[nome do serviço] [nome do contêiner]=[nome da imagem]
```

Atualiza a imagem de contêiner de um deployment.

KUBERNETES – PRINCIPAIS COMANDOS – APPLY

```
c:> kubectl apply -f [arquivo yaml]
```

Aplica a configuração de um recurso a partir de um arquivo *.yaml.

KUBERNETES – PRINCIPAIS COMANDOS – PATCH

```
c:> kubectl patch deployment [nome do deploy] --patch $(Get-Content [yaml file name] -Raw)
c:> kubectl patch deployment [nome do deploy] --patch "$(cat [yaml file name])"
```

Atualização parcial de um objeto Kubernetes.

KUBERNETES

LABORATÓRIO 12

KUBERNETES

LABORATÓRIO 13