

DEVOPS, Container e Kubernetes

Kubernetes é a plataforma mais indicada se você quer hospedar aplicações nativas em nuvem que exigem escalabilidade rápida. Com transmissão de dados em tempo real por meio do Apache Kafka, esses clusters podem incluir hosts em nuvem pública, privada ou híbrida.

Criado e desenvolvido por engenheiros do Google, o Kubernetes trouxe o pioneirismo da empresa no desenvolvimento da tecnologia de containers Linux. Além disso, a organização já revelou publicamente que, no Google, tudo é executado em containers e que é, inclusive, a tecnologia por trás dos serviços em cloud da empresa.

O Kubernetes é uma tecnologia open source. Por isso, não possui uma estrutura formal de suporte na qual as empresas podem confiar totalmente. Problemas com a implantação do Kubernetes quando da execução no ambiente de produção podem representar grande dor de cabeça para o negócio e os clientes.

Para que isso não aconteça, existem soluções corporativas que oferecem todos os elementos extras que tornam o Kubernetes potente e viável para as



Isso significa que os desenvolvedores da sua empresa podem criar novas aplicações em containers, hospedá-las e implantá-las na nuvem. Tudo com escalabilidade, controle e orquestração capazes de transformar boas ideias em valiosos negócios de modo rápido e fácil.



Por que o Kubernetes é essencial?

Aplicações de produção abrangem múltiplos containers, que precisam ser implementados em vários hosts do servidor. A segurança dos containers conta com diversas camadas e pode vir a ser complexa. Assim sendo, é o Kubernetes que fornece os recursos de orquestração e gerenciamento necessários para implantar containers em escala para essas cargas de trabalho.

Com a orquestração do Kubernetes, é possível criar serviços de aplicações que abrangem múltiplos containers, programar seu no cluster, escalá-los e, com o passar do tempo, gerenciar sua integridade. Com o Kubernetes, a empresa adota medidas reais para aprimorar a segurança da TI.

É preciso, ainda, integrar o Kubernetes com os serviços de rede, armazenamento, segurança, telemetria, entre outros, para oferecer uma infraestrutura de containers global.



Claramente, isso depende do uso que cada empresa faz dos containers em seus próprios ambientes. Quando escalado para um ambiente de produção e diversas aplicações, nota-se que é necessário dispor de vários containers alocados funcionando em conjunto para disponibilizar serviços individuais. Isso multiplica substancialmente o número de containers no ambiente e, quanto mais eles se acumulam, mais a complexidade aumenta.

O Kubernetes corrige diversos problemas comuns que decorrem da proliferação de containers, organizando-os em “pods”. Os pods adicionam uma camada de abstração aos containers agrupados, de forma que se torna mais fácil programar as cargas de trabalho e oferecer os serviços necessários, como rede e armazenamento, a esses containers.

Já outros componentes do Kubernetes são úteis no balanceamento de carga entre os pods. Assim, é possível garantir que o número de containers em execução seja suficiente para fornecer suporte às cargas de trabalho.

Com a implementação correta do Kubernetes e a ajuda de outros projetos open source, como por exemplo, Atomic Registry, Open vSwitch, Heapster, OAuth e SELinux, as empresas podem orquestrar a totalidade da infraestrutura de containers.

Como o Kubernetes funciona e para o que ele serve?

A principal vantagem que as empresas possuem ao usar o Kubernetes, em especial se estiverem otimizando o desenvolvimento de aplicações para a nuvem, é que elas terão uma plataforma para programar e executar containers em clusters de máquinas físicas ou virtuais.

Em outras palavras, com o Kubernetes, fica mais fácil implementar uma infraestrutura baseada em containers para ambientes de produção e confiar totalmente nela. O Kubernetes permite que os containers realizem muitas das atividades possibilitadas por outros sistemas de gerenciamento ou plataformas de aplicações, visto que seu objetivo é automatizar completamente as tarefas operacionais.



-
- Orquestrar containers em vários hosts;
 - Controlar e automatizar as implantações e atualizações de aplicações;
 - Um melhor aproveitamento do hardware para maximizar os recursos necessários na execução das aplicações corporativas;
 - Montar e adicionar armazenamento para executar aplicações com monitoração de estado;
 - Escalar com rapidez as aplicações em containers e os recursos relacionados;
 - Gerenciar serviços de forma declarativa, propiciando que as aplicações sejam executadas sempre da mesma forma como foram implantadas;
 - Verificar a integridade e auto recuperação das aplicações com posicionamento, reinício, replicação e escalonamento automáticos.

Contudo, para oferecer plenamente esses serviços orquestrados o Kubernetes depende de outros projetos. Com a inclusão de outros projetos open source, o Kubernetes pode atingir sua capacidade total. Estão entre os projetos necessários:

- Registro, como Atomic Registry e Docker Registry;
- Rede, como OpenvSwitch e roteamento de borda inteligente;
- Telemetria, como Heapster, Kibana, Hawkular e Elastic;
- Segurança, como LDAP, SELinux, RBAC e OAUTH com camadas de multilocalização;
- Automação, com adição de playbooks do Ansible para instalação e gerenciamento do ciclo de vida do cluster;
- Serviços, ofertados em variado catálogo de conteúdos previamente criados a partir de padrões de aplicações populares;
- Faça uma avaliação gratuita do Red Hat OpenShift.

A linguagem do Kubernetes



Kubernetes, é necessário ter conhecimento de alguns termos:

- **Master:** é a máquina que controla os nós do Kubernetes. É onde todas as atribuições de tarefas têm origem.
- **Nó:** são máquinas que realizam as tarefas solicitadas e atribuídas. Os nós são controlados pela máquina mestre do Kubernetes.
- **Pod:** grupo de um ou mais containers implementados em um só nó. Todos os containers em um pod compartilham o mesmo endereço IP, IPC, nome do host e outros recursos. Os pods separam a rede e o armazenamento do container subjacente, o que facilita a movimentação dos containers pelo cluster.
- **Controlador de replicações:** controla quantas cópias idênticas de um pod devem ser executadas em determinado local do cluster.
- **Serviço:** desacopla as definições de trabalho dos pods. Os proxies de serviço do Kubernetes levam as solicitações de serviço para o pod correto automaticamente e independentemente do local do pod no cluster ou mesmo se ele foi substituído.
- **Kubelet:** serviço executado nos nós que lê os manifestos do container e assegura que os containers definidos foram iniciados e estão sendo executados.
- **Kubectl:** ferramenta de configuração da linha de comando do Kubernetes.

Uso do Kubernetes em produção

O Kubernetes é uma tecnologia open source. Por isso, não possui uma estrutura formal de suporte na qual as empresas podem confiar totalmente. Problemas com a implantação do Kubernetes quando da execução no ambiente de produção podem representar grande dor de cabeça para o negócio e os clientes.

Para evitar isso, existe o Red Hat OpenShift, uma solução corporativa que oferece o Kubernetes e muito mais. O OpenShift conta com elementos extras que tornam o Kubernetes potente e viável para as empresas, incluindo componentes de registro, rede, telemetria, segurança, automação e serviços.



com escalabilidade, controle e orquestração capazes de transformar boas ideias em valiosos negócios de modo rápido e fácil.

Além disso, a maior vantagem dessa solução é que a plataforma foi desenvolvida pela Red Hat, empresa líder global em tecnologia open source, e conta com seu suporte.

Veja como o Kubernetes se encaixa em sua infraestrutura

O Kubernetes é executado em um sistema operacional interagindo com pods de containers executados em nós. A máquina mestre do Kubernetes acata os comandos de um administrador ou equipe de DevOps e retransmite as instruções aos nós subservientes.

Essa retransmissão é realizada em conjunto com vários outros serviços para que, automaticamente, seja decidido qual é o nó mais adequado para a tarefa. Após isso, os recursos são alocados e os pods do nó são designados para cumprir a tarefa solicitada.

Do ponto de vista da infraestrutura, são poucas as mudanças que serão efetuadas em comparação ao modo como sua empresa faz o gerenciamento dos containers. O controle sobre os containers acontece em nível superior, sem a necessidade de microgerenciar separadamente cada container ou nó. Será necessário realizar algum trabalho, entretanto, em grande parte, trata-se apenas de uma questão de atribuição de um master do Kubernetes e da definição dos nós e pods.



WEBINAR
Desafios para implantações do Kubernetes em um ou vários clusters

 **SUSE**

 **memora**
PROCESSOS INOVADORES

Tecnologia do docker



iniciar containers especificados. O kubelet, então, regular e continuamente, coleta do docker os status desses containers e agrega as informações no master.

O docker insere os containers nesse nó, iniciando-os e interrompendo-os normalmente. A diferença é que é um sistema automatizado que agora solicita que o docker realize essas tarefas em todos os nós de todos os containers, ao invés de o administrador realizá-las manualmente.

Quer conhecer nossas soluções?

ENTRE EM CONTATO



A Memora é uma empresa dedicada a fornecer soluções em Gestão de Processos para clientes públicos e privados. Os resultados alcançados demonstram na prática a visão da empresa.



[Home](#)[Empresa](#)[Clientes](#)[Soluções](#)[Histórias Memoráveis](#)[Alianças](#)[Tra](#)

Matriz

Brasília

SIG Quadra 04 Lote 625

Parte A 70.610-440

(61) 3963-0030

contato@memora.com.br

Filiais

Goiânia

Av. Castelo Branco, 371

Edifício Lourenço Office, sala 2101

74.140-150 – St. Oeste

(62) 99421-7576

contato@memora.com.br

Lisboa

Rua Ivens N° 42 – 1° andar

1200-227 – Lisboa

+351 910807167 | +351 211218022

contato@memora.com.br

Contatos



contato@memora.com.br



(61) 3963-0030





Home

Empresa

Clientes

Soluções

Histórias Memoráveis

Alianças

Tra



© 2021 Memora | Desenvolvido por **Fusão Online**

