Artigo
Danusa Calixto · Dez. 22, 2022 6min de leitura

Entrega contínua de sua solução InterSystems usando GitLab – Parte V: por que contêineres?

Nesta série de artigos, quero apresentar e discutir várias abordagens possíveis para o desenvolvimento de software com tecnologias da InterSystems e do GitLab. Vou cobrir tópicos como:

- Git básico
- Fluxo Git (processo de desenvolvimento)
- Instalação do GitLab
- Fluxo de trabalho do GitLab
- Entrega contínua
- Instalação e configuração do GitLab
- CI/CD do GitLab
- **Por que contêineres?**
- CI/CD do GitLab usando contêineres

No primeiro artigo, abordamos os fundamentos do Git, por que um entendimento de alto nível dos conceitos do Git é importante para o desenvolvimento de software moderno e como o Git pode ser usado para desenvolver software.

No segundo artigo, abordamos o fluxo de trabalho do GitLab: um processo inteiro do ciclo de vida do software e a entrega contínua.

No terceiro artigo, abordamos a instalação e configuração do GitLab e a conexão dos seus ambientes a ele

No quarto artigo, escrevemos uma configuração de CD.

Neste artigo, falaremos sobre os contêineres e como (e por que) podem ser usados.

Este artigo presume a familiaridade com os conceitos de docker e contêiner. Confira estes artigos do @Luca Ravazzolo se quiser ler sobre contêineres e imagens.

Vantagens

Há muitas vantagens em usar contêineres:

- Portabilidade
- Eficiência
- Isolamento
- Leveza
- Imutabilidade

Vamos falar sobre cada uma em detalhes.

Portabilidade

Um contêiner embrulha um aplicativo com tudo o que ele precisa para ser executado, como arquivos de

configuração e dependências. Isso permite que você execute aplicativos de maneira fácil e confiável em diferentes ambientes, como seu desktop local, servidores físicos, servidores virtuais, testes, staging, ambientes de produção e nuvens públicas ou privadas.

Outro ponto da portabilidade é que, depois de criar sua imagem do Docker e verificar que ela é executada corretamente, ela pode ser executada em qualquer outro lugar que execute o docker, que atualmente são os servidores Windows, Linux e MacOS.

Eficiência

Você só precisa que o processo do aplicativo seja executado, e não todo o sistema, etc. E os contêineres oferecem exatamente isso: eles executam apenas os processos de que você precisa explicitamente e nada mais. Como os contêineres não exigem um sistema operacional separado, eles consomem menos recursos. Enquanto uma VM costuma ter vários gigabytes de tamanho, um contêiner geralmente tem apenas algumas centenas de megabytes, tornando possível executar muito mais contêineres do que VMs em um único servidor. Como os contêineres têm um nível de uso mais alto em relação ao hardware subjacente, você precisa de menos hardware, resultando em uma redução nos custos dos servidores bare metal, bem como dos centros de processamento de dados.

Isolamento

Os contêineres isolam seu aplicativo de todo o resto e, embora vários contêineres possam ser executados no mesmo servidor, eles podem ser completamente independentes uns dos outros. Qualquer interação entre contêineres deve ser explicitamente declarada como tal. Se um contêiner falhar, ele não afetará os outros e poderá ser reiniciado rapidamente. A segurança também se beneficia desse isolamento. Por exemplo, explorar a vulnerabilidade do servidor web em um servidor bare metal pode dar a um invasor acesso a todo o servidor, mas, no caso dos contêineres, o invasor só teria acesso ao contêiner do servidor web.

Leveza

Como os contêineres não exigem um sistema operacional separado, eles podem ser iniciados, interrompidos ou reinicializados em questão de segundos, o que acelera todos os pipelines de desenvolvimento relacionados e o tempo de produção. Você pode começar a trabalhar antes e não gastar nenhum tempo na configuração.

Imutabilidade

A infraestrutura imutável é composta por componentes imutáveis que são substituídos a cada implantação, em vez de serem atualizados no local. Esses componentes são inicializados a partir de uma imagem comum que é criada uma vez por implantação e pode ser testada e validada. A imutabilidade reduz a inconsistência e permite a replicação e a movimentação entre diferentes estados do seu aplicativo com facilidade. Mais sobre a imutabilidade .

Novas possibilidades

Todas essas vantagens nos permitem gerenciar a infraestrutura e o fluxo de trabalho de maneiras totalmente novas.

Orquestração

Há um problema com ambientes bare metal ou VM, eles ganham *individualidade*, o que traz várias surpresas depois, geralmente desagradáveis. A resposta para isso é a **infraestrutura como código**, o gerenciamento da infraestrutura em um modelo descritivo, usando as mesmas versões que a equipe de DevOps usa para o código-fonte.

Com a infraestrutura como código, um comando de implantação sempre coloca o ambiente de destino na mesma

configuração, não importa o estado inicial do ambiente. Isso é alcançado ao configurar automaticamente um destino existente ou descartar o destino existente e recriar um novo ambiente.

Assim, com a infraestrutura como código, as equipes fazem alterações na descrição do ambiente e na versão do modelo de configuração, que normalmente está em formatos de código bem documentados, como JSON. O pipeline de lançamento executa o modelo para configurar ambientes de destino. Se a equipe precisar fazer alterações, ela editará a fonte, e não o destino.

Tudo isso é possível e muito mais fácil de fazer com contêineres. Leva poucos segundos para desativar um contêiner e iniciar outro, enquanto o provisionamento de uma nova VM leva alguns minutos. E nem estou falando em reverter um servidor para um estado limpo.

Escalonamento

Do ponto anterior, você pode ter uma ideia de que a infraestrutura como código é estática por si só. Não é, pois as ferramentas de orquestração também podem fornecer escalonamento horizontal (provisionando mais do mesmo) com base na carga de trabalho atual. Você só deve executar o que é necessário no momento e escalonar seu aplicativo de acordo. Isso também pode reduzir custos.

Conclusão

Os contêineres podem otimizar seu pipeline de desenvolvimento. A eliminação de inconsistências entre ambientes permite testes e depurações mais fáceis. A orquestração permite que você crie aplicativos escalonáveis. A implantação ou reversão para qualquer ponto do histórico imutável é possível e fácil.

As organizações querem trabalhar em um nível mais alto, onde todos os problemas listados acima já estejam resolvidos e onde encontramos agendadores e orquestradores lidando com mais coisas de maneira automatizada.

O que vem a seguir

No próximo artigo, vamos falar sobre o provisionamento com contêineres e a criação da configuração de CD que usa o contêiner Docker do InterSystems IRIS.

#Conteinerização #Docker #Gestão da Mudança #Integração Contínua #Caché

URL de origem:https://pt.community.intersystems.com/post/entrega-cont%C3%ADnua-de-sua-solu%C3%A7%C3%A3o-intersystems-usando-gitlab-%E2%80%93-parte-v-por-que-cont%C3%AAineres