

Blog do Fernando Quadro

Geotecnologias livres e afins...

Clusterização do GeoServer com Docker – Parte 1



No [FOSS4G Argentina](#) desse ano (2019), o Gabriel Roldan ministrou um treinamento de GeoServer Avançado. Neste treinamento, ele ensinou como escalar o GeoServer

com o Docker. Eu achei o material muito interessante, originalmente escrito em espanhol, e resolvi traduzir e postar aqui no blog.

1. Clustering no GeoServer

Clustering é um sistema de processamento paralelo ou distribuído que envolve computadores independentes (ou máquinas virtuais, contêineres docker, etc.).

No GeoServer, ele pode ser aplicado para obter uma configuração de alta disponibilidade ou para obter maior escalabilidade. Independentemente do motivo, existem algumas limitações que devem ser levadas em consideração e, quando possível, devem ser resolvidas.

Na parte restante desta seção, serão fornecidas informações para obter um entendimento básico de clustering e, em seguida, analisar as várias possibilidades de agrupamento do GeoServer, usando o docker e o docker-compose para definir as características do cluster.

2. Introdução ao Clustering e Alta Disponibilidade

Técnicas de clustering são usadas para melhorar o desempenho e a disponibilidade de um sistema complexo. Em termos gerais, um cluster é considerado como um conjunto redundante de serviços que fornece o mesmo conjunto de funcionalidades.

A qualidade do cluster pode ser medida por:

Confiabilidade: a capacidade de fornecer respostas bem-sucedidas em todas as solicitações recebidas. **Disponibilidade** do tempo de atividade do servidor (geralmente medido como % do tempo de atividade anual). **Desempenho** medido pelo tempo médio gasto pelo serviço para fornecer respostas ou desempenho. **Escalabilidade** é a capacidade de lidar com uma quantidade crescente de trabalho de um maneira eficiente, sem degradação na qualidade do serviço.

2.1 Alta disponibilidade

Clusters de alta disponibilidade (HA) são grupos de serviços que podem ser usados de forma confiável com tempo de inatividade mínimo (ou nulo). Sem o armazenamento em cluster, se um serviço falhar ou estiver muito ocupado, o usuário que solicitar esse serviço nunca receberá uma resposta rápida. O pool de alta disponibilidade deve ser projetado para remediar essa situação, detectando um serviço inativo (por meio de um *watchdog*) e reiniciando-o imediatamente. Durante

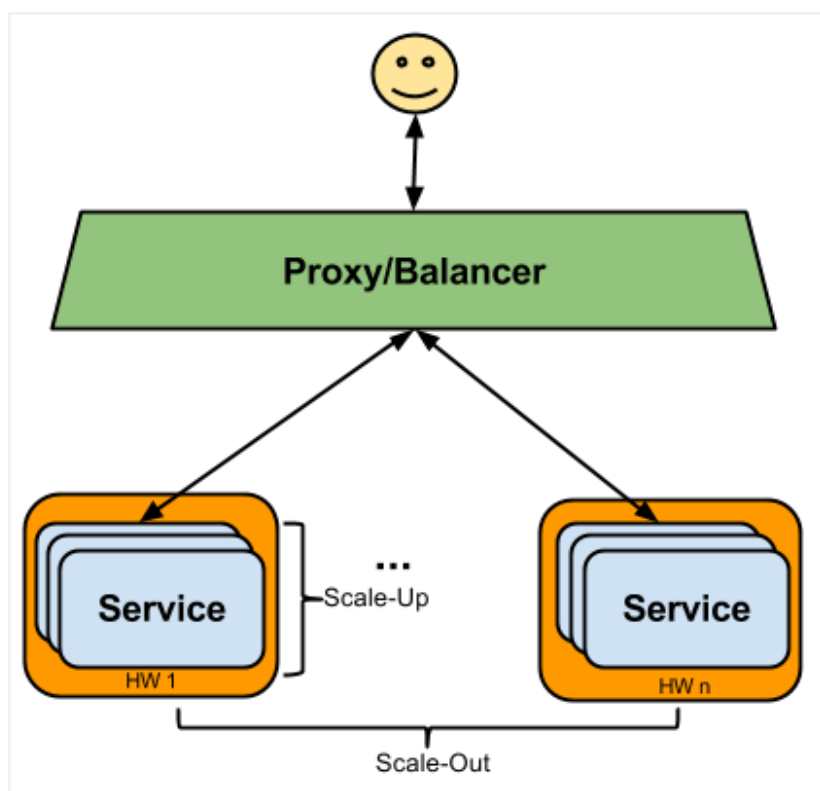
esta operação, o serviço será fornecido por uma instância de *failover* do mesmo serviço.

2.2 Escalabilidade

“Escalar” um sistema geralmente significa adicionar mais instâncias do produto. Podemos distinguir duas maneiras diferentes de escalar:

A escalabilidade horizontal também é conhecida como *scale-out*, que pode ser feito adicionando recursos de hardware (HW) extras ao grupo existente. Em nosso contexto, isso significa adicionar mais máquinas físicas ou virtuais com o GeoServer instalado.

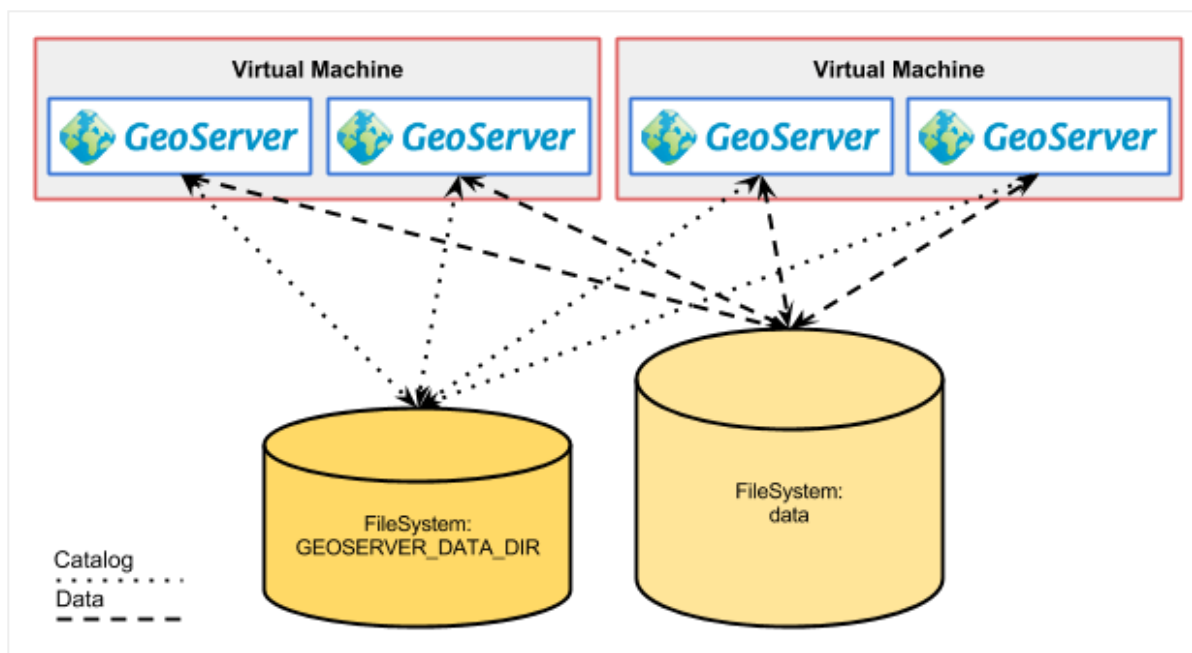
A escalabilidade vertical pode ser atingida com a obtenção de hardware mais potente (mais CPU/memória) e normalmente deve ser combinada adicionando mais instâncias de software também no mesmo servidor (na prática, não é um software 100% escalável linearmente). Em nosso contexto, isso significa instalar mais instâncias do GeoServer no HW existente para utilizar totalmente os recursos disponíveis, particularmente as CPUs adicionais.



3. Configuração de alta disponibilidade para o GeoServer

Isso requer o agrupamento de instâncias do GeoServer para implementar uma configuração de alta disponibilidade, bem como para obter uma maior

escalabilidade.



Esse tipo de configuração envolve uma série de problemas e só é viável para um serviço estático. Isto é, uma vez configurado, assume somente leitura em termos da configuração do catálogo. Do contrário a falta de comunicação entre as instâncias resulta em um desastre.

4. Backoffice – Configuração de produção (Diretório de Dados separado)

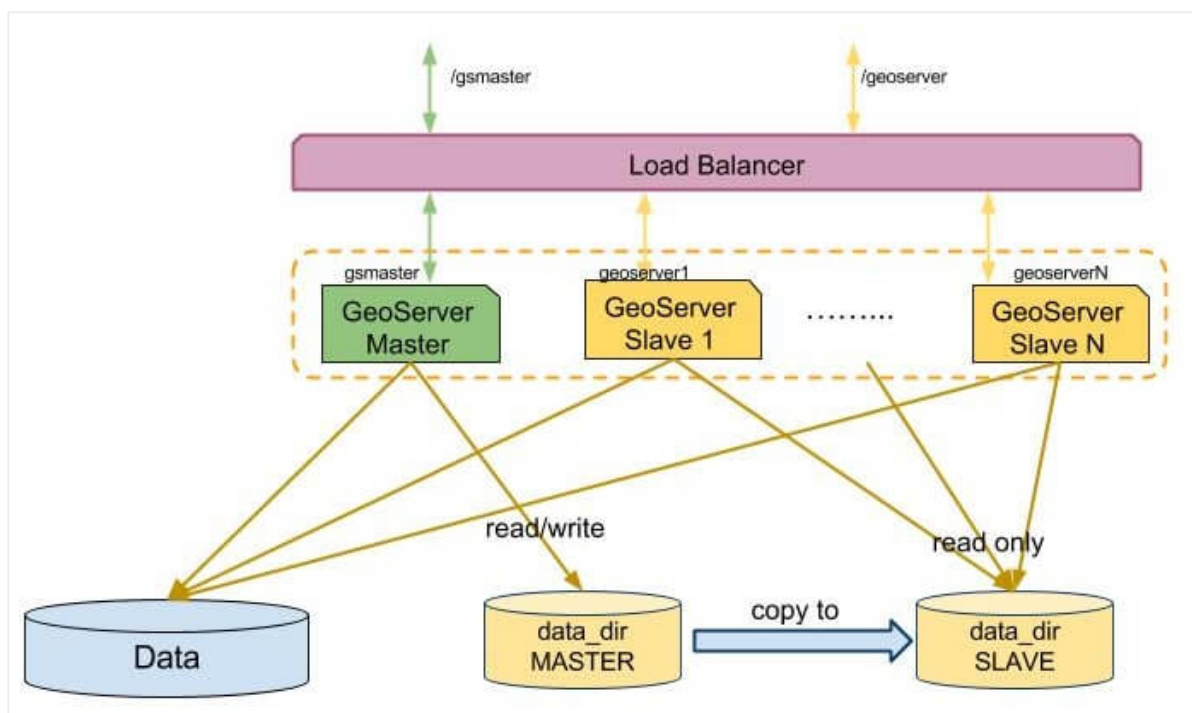
Ao configurar um cluster do GeoServer com um diretório de dados compartilhado, é recomendável configurar um GeoServer off-line que trabalhe com uma cópia do diretório de dados do cluster (área de preparação), faça alterações até que um novo design de configuração satisfatório seja alcançado e, em seguida, propague as alterações para o cluster de produção com as seguintes etapas:

- Copie o diretório de dados (configuração) da área de preparação para a área de produção.
- Execute o **comando de configuração REST “reload”** em cada nó do cluster para que a nova configuração seja carregada no disco.

Essa abordagem começa a mostrar suas limitações quando qualquer uma das seguintes situações se aplica:

- Alterações freqüentes de configuração que devem ser aplicadas imediatamente a todo o cluster.

- A configuração contém muitas camadas de dados (centenas ou mais), o que implica um longo tempo de recarga. Durante a fase de recarga, o servidor pode não responder corretamente às solicitações OGC porque a configuração está em um estado inconsistente e incompleto.



5. Clustering ativo do GeoServer

Como foi visto nas seções anteriores, existem várias abordagens para implementar um cluster do GeoServer com base em diferentes opções para compartilhar/sincronizar diretórios de dados e recarregar a configuração. No entanto, essas técnicas têm limitações intrínsecas em termos de escalabilidade em relação ao número de camadas, então decidimos criar uma Extensão de Cluster no GeoServer específica para superá-las.

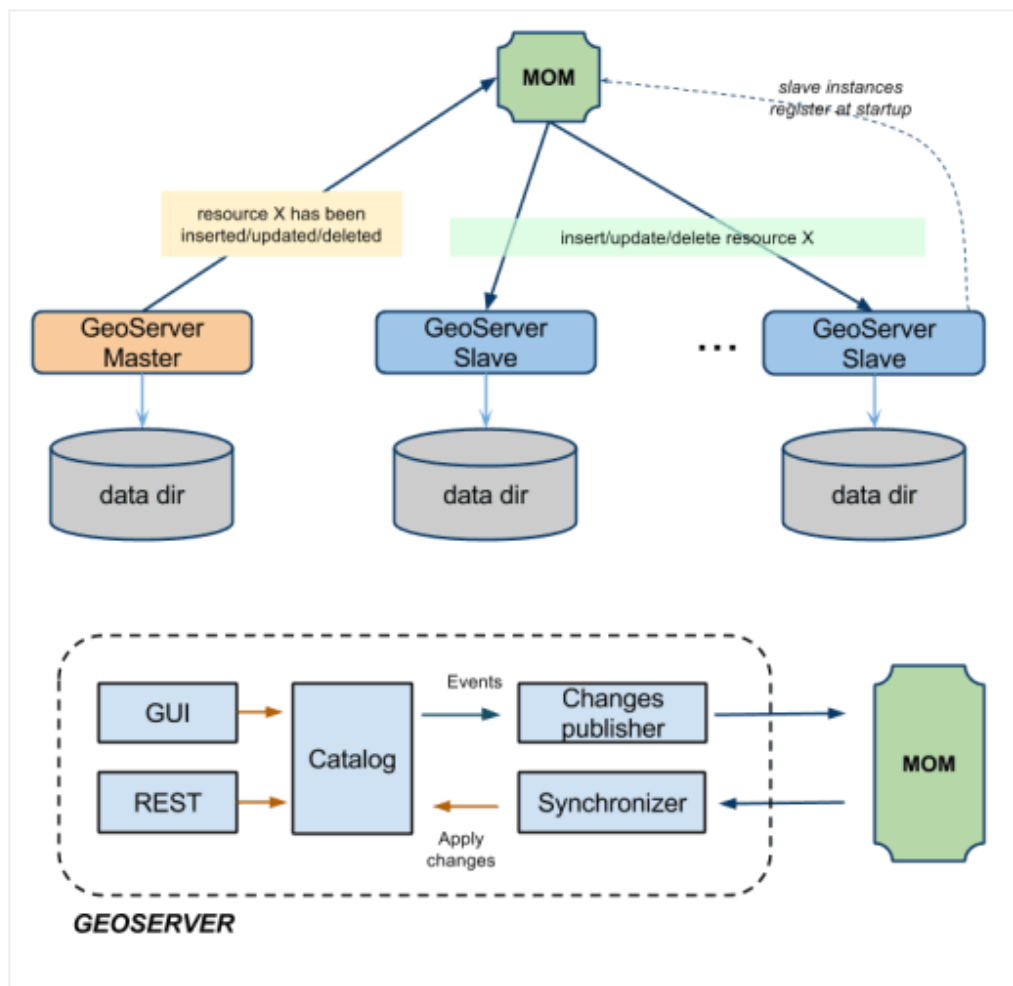
Implementamos uma extensão **Multi-Master** para o GeoServer que aproveita o **Message Oriented Middleware (MOM)** para manter todos os nós do cluster sincronizados em relação às suas configurações.

- As instâncias *Master* aceitam alterações na configuração interna, mantêm-nas em seu próprio diretório de dados e as enviam para os *Slaves* através do MOM (com entrega garantida)
- As instâncias *Slave* não devem ser usadas para alterar a configuração do REST ou da interface do usuário, pois elas são configuradas para receber e injetar

(diretamente na memória) as alterações de configuração transmitidas pelo(s) *Master(s)* pelo MOM

- O MOM é usado nos bastidores para que os *Masters* e *Slaves* troquem mensagens contendo alterações de configuração. Em alguns casos, usaremos o termo *Broker* como sinônimo de MOM

Cada *slave* pode ser configurado para ter seu próprio diretório de dados (privado), neste caso ele será configurado para usar uma assinatura durável no MOM para manter seu diretório de dados sincronizado com o do *Master*, para que quando seja feita uma alteração de configuração do master, através do MOM seja recebida e também persistida. Com esta configuração, no caso dos dados de um slave ser apagado quando subir novamente, ele receberá uma série de mudanças de configuração para alinhar seu diretório de dados com o do mestre. Na ilustração a seguir, esta configuração é mostrada (com um único *master*: foco de vários *slaves*).

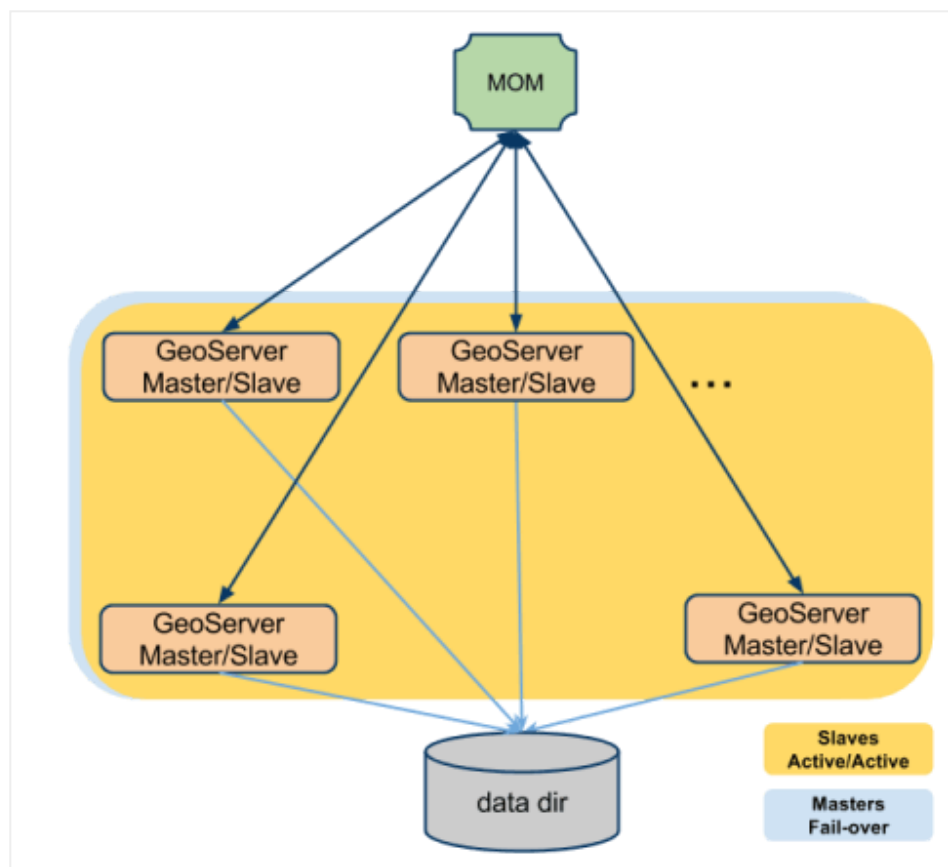


Uma configuração semelhante é útil nos casos em que é possível compartilhar o diretório de dados em várias instâncias (por exemplo, dimensionar ou dimensionar

verticalmente em uma única instância grande, mas até dimensionar enquanto um sistema de arquivos compartilhado em cluster ainda estiver disponível). Vale ressaltar que, embora as alterações na configuração através da interface do usuário ou da interface REST devam ser endereçadas ao máximo possível ao mesmo mestre (portanto, recomendamos configurá-las em “failover” ou Ativo/Passivo). No entanto, é possível configurar todas as instâncias em Ativo/Ativo ao tentar responder a consultas OGC.

Vale ressaltar que com a extensão de clustering ativo do GeoServer, um nó pode ser *master* e *slave* ao mesmo tempo, o que permite uma configuração “peer to peer”. Em termos gerais, todos os nós podem ser *master* e *slave* ao mesmo tempo, entretanto, como mencionado acima, seria bom configurar o balanceador de carga para usar um *switch* no modo “failover” para alterações de configuração (GUI ou REST).

Na ilustração a seguir, é mostrada uma configuração com um design P2P e um diretório de dados compartilhado.



Até agora vimos apenas a teoria, e todo o crédito pela [documentação](#) e implementação deve ser dado a [GeoSolutions](#). No próximo post, iremos ver como implementar tudo o que foi explicado acima. Então não perca o [post de amanhã](#)!

Fonte: [GitHub do Gabriel Roldan](#)



Fernando Quadro

Fernando Silveira de Quadro é analista de sistemas e consultor em geotecnologias livres, formado em Ciência da Computação e pós-graduado (MBA) em Gerenciamento de Projetos pela Universidade do Vale do Itajaí (UNIVALI).

Possui vasta experiência em SIG (Sistema de Informação Geográfica) com ênfase em software livre.

July 22, 2019

GIS

Cluster, Docker, GeoServer

[Previous post](#)

[Next post](#)

TAMBÉM EM FERNANDOQUADRO

2 anos atrás • 2 comentários

**Introdução ao
Leaflet –
Sobreposição de**

2 anos atrás • 9 comentários

**Como utilizar o
Cluster Marker
no Leaflet**

3 anos atrás

O milé

[Comentários](#) [Comunidade](#) [🔒 Política de Privacidade](#)

1 [Entrar](#) ▼

♥ [Recomendar](#)

🐦 [Tweet](#)

f [Compartilhar](#)

[Ordenar por Mais votados](#) ▼

Iniciar a discussão...

FAZER LOGIN COM

OU REGISTRE-SE NO DISQUS [?](#)

Nome

Seja o primeiro a comentar.

✉ [Inscreva-se](#)

🔗 [Adicione o Disqus no seu site](#) [Adicionar Disqus](#) [Adicionar](#)