

IaC: Infraestrutura como um código

Apresentação:

Administrar um ambiente de infraestrutura é uma tarefa para um administrador ou uma equipe de administradores dedicados às tarefas de implantar e manter um (ou mais) ambiente.

Mesmo uma equipe especializada, a maneira de administrar o ambiente de infraestrutura era executada de forma manual.

São chamadas de tarefas manuais, tarefas que são executadas localmente ou remotamente no host. Depende de um analista para iniciar a execução dos procedimentos e processos para atingir um resultado.

Como por exemplo:

Para instalar um serviço (programa); Configura-lo; Gerenciá-lo.

Era uma tarefa relativamente fácil, porém conforme cresce o ambiente e o número de serviços e servidores as demandas que devem ser executadas para administração, quando realizado de forma manual, consomem muito tempo e esforço de uma equipe de profissionais. Torna-se o problema para uma entrega Ágil e com qualidade.

São problemas comuns durante administração manual: (Que podem ou “devem” ser automatizadas)

- Falta de padrões para execução das demandas;
- Pular etapas do procedimento;
- Fadiga por executar tarefas repetitivas;
- Sem garantia de execução do procedimento (e/ou) funcionamento;
- Dificuldade para documentar o ambiente e os processos executados;
- Dificuldade para encontrar a causa raiz do problema.

“A solução é gerenciar ambientes com ferramentas de automação”.

Como parte filosofia **DevOps** propõe **integração, agilidade e escalabilidade** para administração de ambientes de infraestrutura.

Tratando de forma mais dinâmica as tarefas e rotinas do ambiente, simplificando a administração e etapas como homologação, análise e testes carga e qualidade, e aplicabilidade em ambiente de produção.

Prover um ambiente **ágil, com integração e escalabilidade**, necessita de soluções (ferramentas) para “*gerenciar configurações*”, “*automação de tarefas*” e “orquestração de ambientes” entre outros.

O uso das soluções de automação envolve descrever a necessidade do ambiente como uma receita, que contém as necessidades como tarefas, rotinas, programas, arquivos de configuração e estado do ambiente como um código para ser interpretado pela ferramenta de automação.

Em vez do analista de se conectar à máquina, e executar o comando para instalar um programa, ele descreve com uma receita; Que é interpretada pelas ferramentas de automação.

Está necessidade de criar código (receita) para automação; É a referência de “**laC Infrastructure As a Code**”.

Melhorias com automação e uso de Infraestrutura como código (laC).

Infrastructure as a Code (laC) significa Infraestrutura como Código. Termo que foi adotado devido às ferramentas de automação, terem como parte fundamental de seu funcionamento, gerenciar o ambiente de infraestrutura como um código que deve ser descrito para ser aplicado.

Algumas aplicabilidades da **laC** é poder **gerenciar, administrar, analisar e provisionar automaticamente**, por meio de código, ao invés de usar um processo manual.

Este código pode ser utilizado para ações sobre um item do ambiente (um único servidor), ou para todo o ambiente (Diversos servidores e sistemas operacionais diferentes).

Infraestrutura como Código é algumas vezes referida como infraestrutura programável.

A abordagem **laC** é do que a entrega e orquestração de um ambiente de forma ágil e segura. Mantendo controle sobre o ambiente.

Utiliza-se de codificação simples e objetiva, que vai ser interpretada nas ferramentas de automação e orquestração.

Considerando que para automatizar, deve-se descrever o código (ações, processos, e procedimentos); Chegamos ao efeito que todo o ambiente foi analisado, e temos documentado (quais são programas, processos, procedimentos executados).

São alguns benefícios de tratar a infraestrutura como um código:

- O código pode ser executado simultaneamente em diversos ambientes.

(A mesma ação, pode ser aplicada em diversos servidores simultaneamente).

- O código pode ser reaproveitado.

(Pode ser feito com parâmetros específicos de um cliente. Mas é facilmente ajustado para se adequar ao ambiente de outro cliente. Facilmente parametrizado).

- O código pode ter mais de uma versão.

(A versão do código de um ambiente de homologação, pode ser copiada em uma nova versão contendo parâmetros para ser utilizado no ambiente de produção).

- O código trata diferenças entre Sistemas Operacionais.

(A partir da gerência de infraestrutura por código, Aplica-se para Windows, Linux, Unix, plataformas 32 ou 64 bits.)

A necessidade para se ter a **IaC** envolve o uso de um solução (ferramenta) de “Automação e orquestração”. São exemplos:

- Ansible
- CFEngine
- Chef
- Puppet
- Rundeck

Ferramentas e Soluções.

Ansible

Ansible é uma ferramenta de código aberto, para automação de tarefas. Similar às ferramentas Puppet e Chef, CFEngine, porém muito mais poderosa para ambientes com sistemas Linux e Unix.

Com ela é possível fazer o deploy de aplicações, provisionando de servidores, realizar diversas configurações de host, automatizar tarefas como:

- Instalação e atualização de pacotes;
- Configuração e deploy de aplicações;
- Gerenciamento de usuários;
- E muitas outras tarefas administrativas com ganho enorme de desempenho, velocidade e produtividade.

O Ansible trabalha com uma linguagem em texto puro, gerando arquivos YAML, o principal arquivo de configuração é chamado de PLAYBOOK onde é definido todas as tarefas que serão executadas no host (Comandos que deveriam ser executados manualmente "yum", "mkdir", "useradd").

No arquivo inventário você adiciona os endereços dos servidores onde o Ansible irá executar esse PLAYBOOK uma receita. Cada Playbook possui roles (grupo de configurações, modulares e variáveis) que são as informações de provisionamento, após definir as roles você definirá em quais hosts essas roles serão executadas.

O Ansible é uma ferramenta muito mais simples que seus concorrentes citados anteriormente, já que ele não precisa que seja instalado agentes nos servidores na qual ele irá executar as tarefas. Ele aplica as configurações via SSH (Secure Shell), que é uma ferramenta padrão para acesso remoto em servidores Linux e Unix.

É possível integrar e utilizar Ansible em conjunto com Docker para resolver duas situações: Implantação de Containers (Hosts com sistemas Linux) e criação de imagens e aplicar suas configurações.

Ganho na produtividade: Open Source, facilmente configurável, utiliza como comunicação SSH, nativo para acesso remoto em hosts Linux. Não precisa de agentes instalados nos clientes.

CFEngine

CFEngine é um software proprietário (Enterprise Edition) e possui uma versão Gratuita (Community) que realiza gerência de configuração e automação.

Através do código utilizado no CFEngine, podemos automatizar tarefas como provisionamento de servidores e também manter o “estado” de configurações, assegurando que o parque de equipamentos esteja sempre no “estado” desejado.

Escrito em C, roda em múltiplas plataformas e arquiteturas, a partir de pequenos dispositivos embarcados, servidores na nuvem, mainframes, até dezenas e centenas de milhares de hosts.

CFEngine pode ser utilizado de várias maneiras, desde provisionamento de servidores, implantação e aplicação de patches de software bem como para tarefas de gerenciamento de configuração, programação, gestão de usuário local, gestão de processos, implementação de segurança, inventário e gerenciamento de conformidade.

Chef

Chef é uma ferramenta de código aberto para gerenciamento de configuração e automação.

Ele opera como um modelo mestre-cliente, com uma estação de trabalho separada necessária para controlar o mestre (servidor).

O projeto Chef é transparente e baseada em seguir as instruções que é dado, o que significa que você terá que certificar-se de suas instruções são claras.

Chef é ideal para as equipes e ambientes com foco no desenvolvimento.

Recomendável para empresas que procuram uma solução mais madura para um ambiente heterogêneo (Windows e Linux).

Possui uma rica coleção de módulos e receitas de configuração. Centralização com git e usa a ferramenta 'Knife' para inserir os agentes nos hosts via ssh, melhorando um pouco a sua abordagem, porém tomando mais tempo para instalação, não é uma ferramenta simples e pode trazer grandes bases de código e complicações ao ambiente.

Puppet

O Puppet é uma ferramenta que funciona de forma autônoma, ou em rede cliente servidor.

Permite administrar máquinas físicas ou virtuais, e um Data Center inteiro com centenas de hosts e sistemas de virtualização como Vmware.

Administra aplicando modernos conceitos de gerência de configuração com o objetivo de garantir a integridade e o pleno funcionamento de ambientes, sistemas e serviços de TI, fazendo isso de forma automatizada, centralizada e ágil.

Há lista de empresas de grande nome no mercado que se beneficiam com a configuração e orquestração utilizando o Puppet;

- EBC (+/- 350 hosts gerenciados pelo Puppet)
- Detran/DF (+/- 100 hosts gerenciados pelo Puppet)
- Caixa/Infraestrutura do Correio (2 Data Centers gerenciados pelo Puppet)
- Twitter, eBay, Google, Disney, Oracle, Github, Nokia, Motorola, Netflix, Vmware, Cisco, Nasa, PayPal, Salesforce, Uol, Locaweb, Walmart, Amil...

O Puppet é a ferramenta desenvolvida e distribuída ou comercializada pela PuppetLabs.

Sua versão comercial é chamado de "Puppet Enterprise" e sua versão Open Source é "Puppet".

Rundeck

O Rundeck é uma aplicação java de código aberto que automatiza processos e rotinas nos mais variados ambientes, gerenciado via interface gráfica fica extremamente simples de verificar status de execuções, saídas de erro, etc.

Basicamente funciona como um "Job Scheduler" simples e poderoso que permite execução de tarefas para um host ou um grupo de hosts.

Muito utilizado quando se trata de ambientes de infraestrutura ágil, principalmente em uma abordagem de "Entrega Contínua", onde em pequenos ciclos novas versões de software são construídas, testadas e liberadas de forma confiável e em curtos períodos de tempo.

Conta com diversos recursos para "Entrega Contínua" como:

- API da Web
- Execução de comando distribuído
- Sistema de execução plugável (SSH por padrão)
- Fluxos de trabalho de várias etapas
- Execução de trabalho com demanda ou corridas programadas
- Console web gráfico para execução de comando e trabalho
- Política de controle de acesso baseada em função com suporte para LDAP / ActiveDirectory.
- Histórico e registros de auditoria

- Abrir integração com ferramentas de inventário de host externo
- Ferramentas de interface de linha de comando

O fluxo de trabalho é tratado como comandos separados em “Jobs”. Entenda como uma job um comando, um script, uma URL (com um script) ou até mesmo chamada para uma outra job. O fluxo de trabalho dele em relação à execução de comandos pode usar três estratégias:

- **Node First**, que executa todos os steps em cada node e só depois vai para o próximo.
- **Parallel**, que executa todos os steps em todos os nodes simultaneamente.
- **Sequential**, executa o primeiro step em todos os nós antes de prosseguir para o próximo step.

Independente da estratégia utilizada, é possível especificar uma ação de tratativa de erros, que se resume em “Parar no step que falhar” ou “Execute os steps faltantes”.

Com o Rundeck existe a possibilidade de executarmos a ação localmente, caso haja alguma outra ferramenta responsável pelo deploy (ansible, puppet, chef, etc) ou permite executar em outros nodes (via SSH).

Se a execução for em hosts remotos, ele permite que seja criada uma orquestração de acordo com a necessidade, por exemplo:

- Executar nos nodes por ordem alfabética ascendente e caso algo falhe continue executando em todos os nós restantes. Outro fator interessante é a possibilidade de enviar notificações por e-mail ou API.

Melhorias do Ambiente Automatizado

Agilidade e Produtividade

As tarefas são executadas e entregues dentro do padrão, com menor tempo para realizar e com garantia de conclusão.

Imagine uma situação de rotina para administrar um ambiente Linux:

- Foi contratado um novo administrador. E será necessário acessar quarenta servidores para criar a conta de usuário, definir a senha inicial e suas permissões.

Para esta tarefa ser feita por um analista, serão executadas as ações;

- 1) Acessar o Servidor via SSH
- 2) Alternar para o usuário administrador do sistema
- 3) Executar o comando para adicionar o usuário
- 4) Executar o comando para incluir o usuário ao grupo de administradores
- 5) Executar o comando para definir uma senha inicial.

Esta ação não é uma tarefa complexa, a mesma tarefa deve ser repetida nos 40 servidores. Em cada servidor a tarefa leva cerca de 4 Minutos sendo feita de forma manual.

(Exemplo pois é relativo dependendo da velocidade de conexão entre cliente e servidor).

Esta mesma tarefa deve ser repetida em 40 servidores, Então para se estimar quanto tempo vai levar para execução da tarefa façamos o cálculo.

(4 Minutos x 40 Servidores = 160 Minutos ou 2H:40M)

A Tarefa será realizada manualmente em 2 Horas e 40 Minutos.

Com base neste mesmo exemplo. Vamos avaliar está mesma tarefa sendo orquestrada via automação.

- 1) As etapas de criação (comandos para adicionar usuário, definir grupo e senha) será descrito em código salvo em arquivo de configuração na ferramenta de automação. (5 Minutos para digitação)
- 2) Após ser criado o arquivo de instrução, devemos solicitar ao servidor de Automação que aplique esta instrução nos 40 servidores. (1 Minuto - A ação é executada de forma simultânea em todas as máquinas).
- 3) Vamos levar em consideração que a Automação terá que se comunicar com o agente de cada cliente e isso depende da velocidade de rede; Mas por exemplo levou 1 Minuto para passar as novas informações para o agentes das 40 máquinas.
- 4) Executar os comandos (1 Minuto).

Como as ações da Ferramenta de Automação são passados para todos os agentes simultaneamente, podemos considerar o tempo para executar a tarefa fazendo o seguinte cálculo:

(5M - Edição + 1M - Passando a informação para o Agente + 1M Execução de comandos)

A Tarefa será realizada automaticamente nos 40 Servidores, em 7 minutos.

Confiabilidade

Uma vez definido qual e o estado padrão, a Ferramenta de Automação garante que o procedimento e ações para manter o estado padrão sejam sempre aplicado aos clientes.

Mesmo que o ambiente tenha sofrido alterações manuais indevidas, A automação garante que o host sempre volte ao estado padrão.

Além do estado padrão, como arquivos de configuração, permissões, programas instalados. Também garante que o serviço esteja sempre em execução mesmo após reinício do sistema, falha do sistema, ou intervenção manual.

Escalabilidade

Para que o ambiente está sendo orquestrado. É necessário um conhecimento avançado sobre cada item do ambiente. Este conhecimento será descrito na configuração para que o orquestrador possa administrar.

Após o ambiente orquestrado. Tudo está registrado e controlado. Este controle permite que este estado possa ser aplicado ou replicado em um novo ambiente.

Este exemplo tanto se aplica para reconfiguração ou replicação de ambiente:

- Houve uma falha e perda total do sistema operacional.
- Foi instalado um novo host com um sistema operacional sem configurações.
- O administrador define que o orquestrador aplique um estado desejado ao novo host.
- O Orquestrador se encarrega de aplicar todas as ações e configurações para retornar o estado desejado; Configura as aplicações, realizando as tarefas de instalar os programas, configurar os arquivos, gerenciar usuários, iniciar aplicações, etc.

Melhorias do Ambiente IaC Automatizado

Instalar e configurar o servidor de Automação para orquestrar e automatizar os serviços e ambiente de infraestrutura dos servidores.

O Servidor será o orquestrador que irá gerenciar as de configurações dos sistemas operacionais e aplicações.

Administrando os serviços de infraestrutura para que sejam utilizados com máxima eficiência e eficácia evitando o DownTime.

Garantindo a confiabilidade dos processos e serviços executados nos servidores.

Estabelecer e aplicar um estado padrão para todo o ambiente, o que irá documentar a infraestrutura; Envolvendo configurações de sistema operacional, configuração de aplicações, e segurança.

Com este controle sobre o ambiente será possível prever as necessidades para estabelecer processos; Ou replicar a configuração de forma padrão para todos os clientes. Beneficiando a disponibilidade e escalabilidade.

Este controle permite remanejar ou reconstruir todo o ambiente de infraestrutura facilmente. Poderá recriar toda a infraestrutura atual, em um novo Data Center como

solução em caso desastre. Com todas as configurações, e ajustes personalizados sem a intervenção humana; Não depende de um analista ou um grupo, iniciar o processo de configuração e instalação de item por item.

Evitando falhas na execução dos procedimentos. Beneficiando a gerência de configuração.

O que é a solução Ágil (IaC)

O Puppet é uma solução distribuída pela empresa Puppet Labs, e é distribuída em duas versões a Open Source e a Enterprise.

A diferença entre a versão Open Source e Enterprise é que a versão Enterprise conta com suporte da equipe Puppet Labs e tem mais integração o gerenciamento de serviços do Vmware; E servidores e serviços Windows.

É uma das melhores soluções de mercado para gerenciamento de configurações e automação de ambiente de infraestrutura de TI.

O Puppet funciona de forma autônoma, em rede cliente/servidor. Gerenciando e automatizando o ambiente de infraestrutura.

Tudo é feito a partir de sua definição de estado; Deve se declarar no estado o que deve ser gerenciado pelo Puppet (Arquivos de configuração, usuários, programas, execução de serviço)

Cadastrar todos os servidores para ser administrados pelo Puppet, e associar o estado para o host. Em seguida aplica automaticamente o estado definido em todo seu ambiente.

Se você está gerenciando apenas alguns servidores ou milhares de máquinas em um Data Center, o Puppet automatiza tarefas que os administradores de sistemas fariam manualmente; Liberando tempo e espaço mental para que os administradores de sistemas possam trabalhar nos projetos que oferecem maior valor de negócio.

E evitando que procedimentos possam ter sido executados fora do padrão ou pulando etapas.

Além da gerência de configuração na implantação de aplicações fornecidos por uma equipe de desenvolvedores de software o Puppet automatiza todas as etapas do processo de entrega de software, como versionamento e aplicação de deploy.

A partir de provisionamento de máquinas físicas e virtuais, programas e arquivos de configuração para orquestração.

E gera e relatórios; do estágio inicial de desenvolvimento de código por meio de testes, produção e liberação de atualizações.

São inúmeros benefícios de um ambiente administrado com Puppet, como Gerenciamento de Configurações; Administração de segurança de serviços e sistemas; Escalabilidade e Alta Disponibilidade.

Compatibilidade

O Puppet é compatível e homologado para diversos sistemas Operacionais como:

- Linux (diversas distribuições)
- IBM AIX
- Solaris
- HP-UX
- Mac OS X
- FreeBSD, OpenBSD
- Windows WorkStation e Servers

Pré-Requisitos

Requisitos mínimos de hardware:

- Processador: 1GHz ou superior com múltiplos cores.
- Memória: 6G de ou superior.
- Disco: 100G ou superior (para o local de instalação do Puppet a parte do Banco de Dados)

Sobre a instalação dos componentes é recomendável a separação dos itens por servidor; Mas não é obrigatório.

1 Servidor para executar o Puppet.

1 Servidor Web para executar o Dashboard Manager. (Serviços Apache, Ruby).

1 Servidor de banco de dados para o PuppetDB (Serviços de SGBD compatíveis: MySql, PostgreSQL).

Sistema Operacional:

- GNU/Linux Debian

- GNU/Linux Ubuntu
- GNU/Linux CentOS
- Linux Red Hat
- Linux Suse
- IBM AIX

Benefícios

Agregam mais valor ao ambiente com maior confiabilidade, garantia, e controle providos pelo gerenciamento via Puppet:

- Analisar e planejar a implantação de gerência de configurações;
- Automatizar serviços em ambientes com sistemas Linux;
- Automatizar serviços em ambientes com sistemas Windows Server;
- Orquestração de Ambiente;
- Gerência de Configuração;
- Automação;
- Agilidade;
- Relatórios; Permite realização de inventário de hardware e software;
- Versionamento;
- Escalabilidade.

Fonte:

Puppet Labs

<http://docs.puppetlabs.com/>

Linkedin

<https://www.linkedin.com/pulse/gerenciamento-com-automacao-de-ambientes-washington-luis-araujo>