

# Aula 18 – Introdução ao Ansible e sua Arquitetura

No cenário atual da tecnologia, a infraestrutura de TI se tornou um ecossistema complexo e dinâmico, onde servidores, redes e aplicações precisam ser configurados e gerenciados de forma eficiente e consistente. A era da configuração manual, propensa a erros e demorada, está rapidamente se tornando uma relíquia do passado. Profissionais de TI buscam incessantemente ferramentas que possam automatizar tarefas repetitivas, garantir a padronização e escalar operações sem aumentar exponencialmente o esforço humano.

É nesse contexto que a Infraestrutura como Código (IaC) emerge como uma metodologia revolucionária, transformando a maneira como interagimos com nossos ambientes digitais. Em vez de configurar manualmente cada componente, nós descrevemos a infraestrutura em arquivos de código, que podem ser versionados, revisados e aplicados de forma automatizada. Essa abordagem não só acelera a implantação e a manutenção, mas também eleva a confiabilidade e a segurança dos sistemas.

Nesta aula, embarcaremos em uma jornada para desvendar o Ansible, uma das ferramentas mais poderosas e populares no universo da IaC e do gerenciamento de configuração. Você compreenderá o que o diferencia de outras soluções, explorará sua arquitetura elegante e sem agentes, e aprenderá os primeiros passos para configurar seu ambiente e executar comandos essenciais. Ao final, você estará apto a descrever os fundamentos do Ansible e a aplicar seus conceitos iniciais para automatizar tarefas em múltiplos servidores, preparando-se para um futuro onde a automação é a chave para a eficiência operacional.

# Gerenciamento de Configuração: Orquestrando a Complexidade

💡 **Analogia do Maestro:** Imagine que você é o maestro de uma orquestra sinfônica, mas em vez de músicos, você tem dezenas ou centenas de servidores, cada um com suas próprias configurações, softwares e dependências.

Se você tivesse que ajustar manualmente cada instrumento antes de cada concerto, o caos seria inevitável e a performance, inconsistente. O gerenciamento de configuração surge como a partitura e a batuta desse maestro, permitindo que você defina o estado desejado de cada "instrumento" (servidor) e garanta que todos toquem em perfeita harmonia, sem a necessidade de intervenção manual constante.

Essa disciplina é fundamental para manter a consistência, a segurança e a eficiência em ambientes de TI cada vez mais distribuídos e complexos. Ela não se trata apenas de instalar um software, mas de garantir que ele esteja na versão correta, com as permissões adequadas, os serviços iniciados e todas as dependências satisfeitas. É a garantia de que, não importa quantos servidores você tenha, todos eles se comportarão exatamente como esperado, reduzindo a "deriva de configuração" – quando as configurações de um servidor se desviam do padrão desejado.

## Provisionamento vs. Gerenciamento de Configuração

### **Provisionamento**

#### Construir a casa

- Cria recursos básicos
- Máquinas virtuais
- Redes e armazenamento
- Estabelece a base

### **Gerenciamento de Configuração**

#### Mobiliário e decorar

- Instala softwares
- Define configurações
- Mantém estado operacional
- Refina e mantém

O provisionamento é como construir a casa: ele cria os recursos básicos, como máquinas virtuais, redes e armazenamento. Já o gerenciamento de configuração é como mobiliário e decorar essa casa, garantindo que cada cômodo (servidor) tenha os móveis (softwares) certos, na posição correta (configurações), e que tudo esteja pronto para uso. O provisionamento estabelece a base; o gerenciamento de configuração refina e mantém o estado operacional.

# Arquitetura do Ansible: Simplicidade e Poder Sem Agentes



## Agentless

Sem necessidade de instalar software cliente em cada servidor



## SSH Nativo

Comunicação via protocolo seguro e universal



## Implantação Rápida

Menos software, menos complexidade, mais velocidade

Uma das características mais notáveis do Ansible, e que o diferencia de muitas outras ferramentas de gerenciamento de configuração, é sua arquitetura "**agentless**" – ou seja, sem a necessidade de instalar um software cliente (agente) em cada servidor que você deseja gerenciar. Essa abordagem simplifica drasticamente a implantação e a manutenção, eliminando a sobrecarga de gerenciar e atualizar agentes em centenas ou milhares de máquinas.

- 📄 **Conceito-chave:** Pense no Ansible como um gerente de projetos que se comunica com sua equipe (os servidores) usando uma linguagem universal e já existente: o SSH (Secure Shell). Em vez de ter um "supervisor" (agente) em cada local de trabalho, o gerente (Control Node) envia instruções diretamente para os trabalhadores (Managed Nodes) via SSH.

## Componentes Principais



### Control Node

A máquina onde o Ansible está instalado e de onde você executa todos os seus comandos e playbooks. É o cérebro da operação, o ponto central de controle.



### Managed Nodes

Os servidores ou dispositivos de rede que você deseja configurar ou automatizar. Não precisam de software especial do Ansible; apenas SSH e Python.

A arquitetura do Ansible é composta por dois componentes principais: o **Control Node** e os **Managed Nodes**. O **Control Node** é a máquina onde o Ansible está instalado e de onde você executa todos os seus comandos e playbooks. É o cérebro da operação, o ponto central de controle. Os **Managed Nodes**, por sua vez, são os servidores ou dispositivos de rede que você deseja configurar ou automatizar. Eles não precisam de nenhum software especial do Ansible; apenas um servidor SSH em execução e, idealmente, Python instalado (que geralmente já vem por padrão na maioria dos sistemas Linux). Essa simplicidade é um dos grandes trunfos do Ansible, permitindo que você comece a automatizar rapidamente, sem complexidades adicionais.

# Vantagens da Arquitetura Agentless



## Curva de Aprendizado Reduzida

Não há necessidade de aprender a instalar e configurar um agente específico para cada sistema operacional ou versão. A barreira de entrada é significativamente menor.



## Segurança Aprimorada

Não há processos adicionais rodando nos servidores gerenciados que possam ser explorados. A comunicação é criptografada via SSH, um protocolo amplamente testado e seguro.



## Manutenção Simplificada

Você não precisa se preocupar em atualizar agentes em todos os seus servidores quando uma nova versão do Ansible é lançada; basta atualizar o Control Node.



## Controle Direto (Push-Based)

Abordagem onde o Control Node "empurra" as configurações para os Managed Nodes, oferecendo flexibilidade e controle direto sobre as operações.

A ausência de agentes nos Managed Nodes traz consigo uma série de vantagens práticas. Primeiramente, a curva de aprendizado e a barreira de entrada são significativamente menores. Não há necessidade de aprender a instalar e configurar um agente específico para cada sistema operacional ou versão. Em segundo lugar, a segurança é aprimorada, pois não há processos adicionais rodando nos servidores gerenciados que possam ser explorados. A comunicação é criptografada via SSH, um protocolo amplamente testado e seguro.

Além disso, a manutenção é simplificada. Você não precisa se preocupar em atualizar agentes em todos os seus servidores quando uma nova versão do Ansible é lançada; basta atualizar o Control Node. Essa abordagem "push-based" (onde o Control Node "empurra" as configurações para os Managed Nodes) contrasta com modelos "pull-based" (onde os agentes nos Managed Nodes "puxam" as configurações de um servidor central), oferecendo flexibilidade e controle direto sobre as operações.



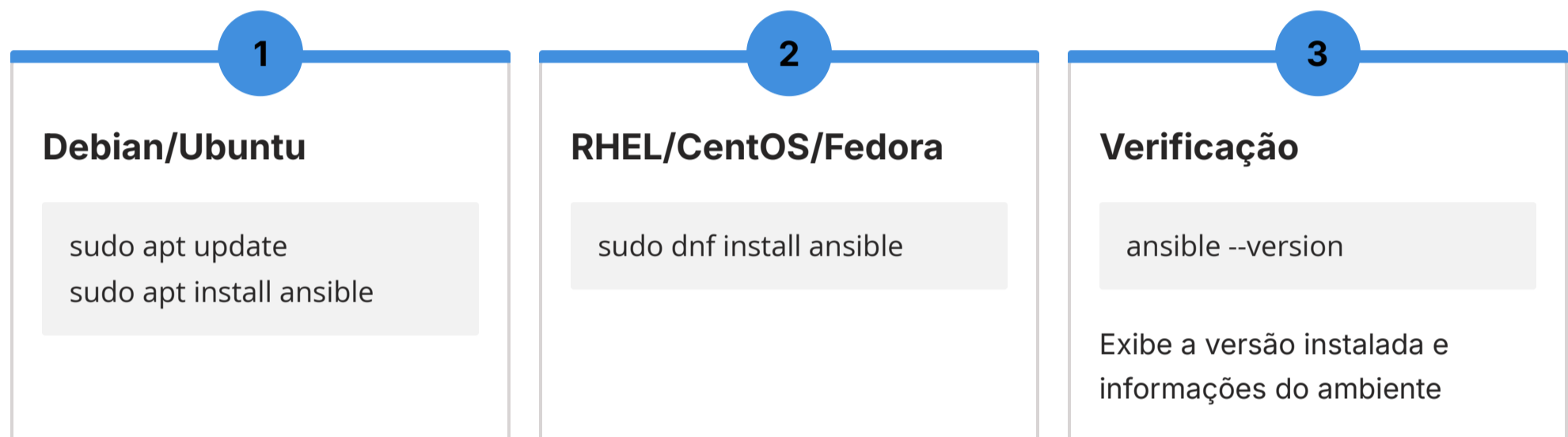
## Exemplo Prático

Imagine que você precisa garantir que um pacote específico, como o nginx, esteja instalado em 50 servidores web. Com uma ferramenta baseada em agentes, você precisaria garantir que o agente estivesse rodando e configurado em cada um desses 50 servidores. Com o Ansible, você simplesmente executa um comando ou playbook do seu Control Node, e ele se conecta via SSH a cada um dos 50 servidores, verifica a presença do nginx e o instala se necessário, tudo de forma transparente e eficiente. Essa é a beleza da simplicidade e do poder do Ansible.

# Configuração do Ambiente: Os Primeiros Passos com Ansible

## Instalação do Control Node

Para começar a usar o Ansible, o primeiro passo é configurar seu Control Node. Lembre-se, esta é a máquina de onde você orquestrará todas as suas tarefas de automação. A instalação do Ansible é surpreendentemente simples e pode ser feita em sistemas operacionais baseados em Linux ou macOS. Geralmente, é recomendado usar o gerenciador de pacotes da sua distribuição ou o pip (gerenciador de pacotes Python) para garantir uma instalação limpa e fácil de manter.



Por exemplo, em sistemas baseados em Debian/Ubuntu, você pode instalar o Ansible com os seguintes comandos:

```
sudo apt update
sudo apt install ansible
```

Já em sistemas baseados em RHEL/CentOS/Fedora, o comando seria:

```
sudo dnf install ansible
```

Após a instalação, você pode verificar se o Ansible está funcionando corretamente digitando `ansible --version` no terminal. Isso exibirá a versão instalada e outras informações úteis sobre seu ambiente. Este é o ponto de partida para toda a sua jornada de automação.

## Configuração de Chaves SSH

A conexão SSH é o coração da comunicação do Ansible com seus Managed Nodes. Para que o Ansible possa se conectar e executar comandos nos servidores remotos, é essencial que o SSH esteja configurado corretamente e que o Control Node tenha acesso sem senha aos Managed Nodes. A forma mais segura e prática de fazer isso é utilizando chaves SSH. Você gera um par de chaves (pública e privada) no seu Control Node e copia a chave pública para o arquivo `~/.ssh/authorized_keys` em cada Managed Node.

# Configuração SSH: Segurança e Automação

01

## Gerar Par de Chaves

Crie chaves pública e privada no Control Node

02

## Copiar Chave Pública

Transfira para `~/.ssh/authorized_keys` nos Managed Nodes

03

## Configurar Permissões

`chmod 600` para chave privada, `chmod 644` para pública

04

## Testar Conexão


Conecte via SSH sem senha para validar

Essa configuração de chave SSH é um pilar de segurança e automação. Ao invés de digitar senhas repetidamente para cada conexão, o Ansible utiliza a chave privada para autenticar-se automaticamente nos servidores remotos. Isso não só agiliza o processo, mas também é crucial para scripts e automações que precisam rodar sem intervenção humana. Certifique-se de que as permissões dos arquivos de chave SSH estejam corretas (geralmente `chmod 600 ~/.ssh/id_rsa` para a chave privada e `chmod 644 ~/.ssh/id_rsa.pub` para a pública).

### **Teste de Conexão**


Para testar a conexão SSH, você pode tentar um simples `ssh usuario@ip_do_servidor` do seu Control Node para um Managed Node. Se você conseguir se conectar sem precisar digitar uma senha, parabéns! Seu ambiente está pronto para o Ansible. Caso contrário, revise a configuração das chaves SSH e as permissões.

Lembre-se que o usuário SSH que o Ansible utilizará nos Managed Nodes deve ter privilégios suficientes para executar as tarefas desejadas, muitas vezes exigindo `sudo` para instalações ou modificações de sistema.

 **Ponto Crítico:** A preparação do ambiente é um passo crítico que garante a fluidez das operações futuras. Uma conexão SSH robusta e sem senhas é a base para a eficiência que o Ansible promete. Sem essa etapa bem-sucedida, qualquer tentativa de automação falhará, pois o Ansible não conseguirá "conversar" com os servidores remotos. É como garantir que o telefone esteja conectado antes de tentar fazer uma ligação importante.

# Comandos Ad-Hoc: Executando Tarefas Rápidas em Múltiplos Servidores

Com o Ansible instalado e a conexão SSH configurada, você está pronto para dar os primeiros passos na automação. Os comandos Ad-Hoc são a maneira mais rápida e direta de executar tarefas simples e pontuais em um ou mais servidores gerenciados, sem a necessidade de escrever playbooks complexos. Eles são ideais para verificações rápidas, reinicialização de serviços, instalação de pacotes ou qualquer outra tarefa que você faria manualmente em um terminal SSH, mas agora em escala.

 **Analogia:** Pense nos comandos Ad-Hoc como um megafone que você usa para dar uma instrução imediata a um grupo de pessoas. Em vez de ir até cada pessoa individualmente, você fala uma vez, e todos ouvem e executam a tarefa.

## Sintaxe Básica

```
ansible <grupo_de_hosts> -m <modulo> -a "<argumentos_do_modulo>"
```

### <grupo\_de\_hosts>

Define em quais servidores o comando será executado (pode ser um IP, um nome de host ou um grupo definido no inventário)

### -m <modulo>

Especifica qual módulo do Ansible será utilizado. Módulos são como ferramentas especializadas (ex: ping, apt, yum, service, command, shell)

### -a "<argumentos>"

Passa os argumentos necessários para o módulo executar a tarefa desejada

# Exemplos Práticos de Comandos Ad-Hoc

## Verificar Conectividade

```
ansible all -m ping
```

Tenta "pingar" todos os servidores e retorna pong se a conexão for bem-sucedida, indicando que o Ansible pode se comunicar via SSH.

## Instalar Pacote

```
ansible webservers -m apt -a "name=nginx state=present" --become
```

**webservers** seria um grupo de servidores no seu inventário. `state=present` garante que o pacote `nginx` esteja presente. `--become` usa privilégios de `sudo`.

## Reiniciar Serviço

```
ansible databases -m service -a "name=postgresql state=restarted" --become
```

Reinicia o serviço PostgreSQL em todos os servidores do grupo `databases`.

## Verificar Uso de Disco

```
ansible all -m shell -a "df -h"
```

Executa o comando `df -h` em todos os servidores para verificar o espaço em disco disponível.

Vamos a alguns exemplos práticos para solidificar o entendimento. Para verificar a conectividade com todos os seus servidores definidos no inventário (que veremos na próxima aula, mas por enquanto, imagine uma lista de IPs), você pode usar o módulo `ping`:

```
ansible all -m ping
```

Este comando tentará "pingar" todos os servidores e retornará `pong` se a conexão for bem-sucedida, indicando que o Ansible pode se comunicar via SSH.

Se você precisar instalar um pacote, como o `nginx`, em todos os seus servidores web, você usaria o módulo `apt` (para Debian/Ubuntu) ou `yum/dnf` (para RHEL/CentOS):

```
ansible webservers -m apt -a "name=nginx state=present" --become
```

Neste exemplo, `webservers` seria um grupo de servidores no seu inventário. `-a "name=nginx state=present"` instrui o módulo `apt` a garantir que o pacote `nginx` esteja presente. `--become` é crucial, pois ele instrui o Ansible a usar privilégios de `sudo` para executar a tarefa, já que a instalação de pacotes geralmente requer permissões de `root`.

Os comandos Ad-Hoc são excelentes para tarefas de manutenção rápida, como reiniciar um serviço:

```
ansible databases -m service -a "name=postgresql state=restarted" --become
```

Ou para verificar o uso de disco:

```
ansible all -m shell -a "df -h"
```

Eles são a porta de entrada para a automação com Ansible, permitindo que você experimente e veja os resultados imediatamente, construindo confiança antes de mergulhar nos playbooks mais complexos.

# A Evolução da IaC: GitOps, DevSecOps e AIOps com Ansible

O universo da Infraestrutura como Código está em constante evolução, e o Ansible, como uma ferramenta central, se adapta e integra-se às tendências mais recentes. Não basta apenas automatizar; é preciso garantir que essa automação seja segura, rastreável e inteligente. As metodologias de GitOps, DevSecOps e AIOps representam o futuro da gestão de infraestrutura, e o Ansible desempenha um papel fundamental em cada uma delas.



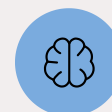
## GitOps

Git como única fonte da verdade



## DevSecOps

Segurança integrada desde o início



## AIOps

Automação inteligente com IA

## GitOps como Padrão

A metodologia GitOps eleva a IaC a um novo patamar, estabelecendo o Git como a única fonte da verdade para toda a infraestrutura. Isso significa que qualquer alteração no ambiente – seja uma nova configuração, uma atualização de software ou a implantação de um novo serviço – deve ser primeiramente declarada em um repositório Git. O Ansible se encaixa perfeitamente nesse modelo: seus playbooks e arquivos de configuração são armazenados no Git, e um sistema de CI/CD (Continuous Integration/Continuous Delivery) aciona o Ansible para aplicar essas mudanças automaticamente sempre que um novo commit é aprovado. Isso garante auditabilidade, versionamento e um fluxo de trabalho colaborativo e seguro.

## Segurança Integrada (DevSecOps)

A segurança não pode ser uma reflexão tardia; ela deve ser incorporada desde o início do ciclo de vida da infraestrutura. O conceito de DevSecOps enfatiza a "segurança à esquerda", ou seja, a integração de práticas de segurança em todas as etapas do desenvolvimento e operação. Com o Ansible, isso se traduz em várias frentes:

### Varredura de Código IaC

Ferramentas podem analisar seus playbooks Ansible no Git para identificar vulnerabilidades ou configurações inseguras antes mesmo de serem aplicadas.

### Gerenciamento de Segredos

O Ansible Vault permite criptografar informações sensíveis (senhas, chaves de API) dentro dos playbooks, garantindo que elas não sejam expostas em texto claro.

### Conformidade e Auditoria

Playbooks podem ser usados para verificar a conformidade dos servidores com políticas de segurança e gerar relatórios de auditoria.

# AIOps e Automação Inteligente

A integração de Inteligência Artificial (IA) e Machine Learning (ML) nas operações de TI, conhecida como AIOps, visa otimizar a gestão de infraestrutura, prever falhas e automatizar a remediação. Embora o Ansible não seja uma ferramenta de IA por si só, ele é o braço executor perfeito para as decisões tomadas por sistemas de AIOps.

## Cenário de AIOps com Ansible

Imagine um cenário onde um sistema de AIOps detecta um padrão de alto uso de CPU em um grupo de servidores e prevê uma falha iminente. Em vez de um operador humano intervir, o sistema de AIOps pode acionar um playbook Ansible para executar ações automatizadas.



### Provisionar Novos Servidores

Automaticamente criar novos servidores para balancear a carga



### Reiniciar Serviços

Reiniciar serviços específicos nos servidores afetados



### Aplicar Patches

Aplicar patches de segurança urgentes



### Coletar Logs

Coletar logs adicionais para análise posterior

O Ansible atua como a ponte entre a inteligência da IA e a ação no mundo real da infraestrutura, transformando insights em automação prática. Essa sinergia entre GitOps, DevSecOps e AIOps, com o Ansible como um pilar de execução, define o futuro da gestão de infraestrutura, tornando-a mais resiliente, segura e autônoma. É a evolução natural da IaC, onde o código não apenas descreve, mas também governa e otimiza a infraestrutura.



**O Futuro da IaC:** A convergência de GitOps, DevSecOps e AIOps com o Ansible representa a evolução natural da Infraestrutura como Código, onde o código não apenas descreve, mas também governa e otimiza a infraestrutura de forma inteligente e segura.

# Quadro Comparativo: Gerenciamento de Configuração vs. Provisionamento

Para consolidar a distinção entre esses dois conceitos fundamentais da IaC, apresentamos um quadro comparativo:

<b>Provisionamento</b>	Criação e configuração inicial de recursos	Infraestrutura bruta (VMs, redes, discos)	Terraform, CloudFormation (criação de VMs na AWS)
<b>Gerenciamento de Configuração</b>	Configuração e manutenção do estado de software	Software, serviços, arquivos de configuração	Ansible, Chef, Puppet (instalação de Nginx, MySQL)

# Quadro Comparativo: Ansible (Agentless) vs. Ferramentas com Agentes

Entender a diferença arquitetural é crucial para escolher a ferramenta certa.

<b>Instalação no Managed Node</b>	Não requer instalação de software cliente	Requer a instalação e execução de um agente
<b>Comunicação</b>	SSH (padrão), WinRM (Windows)	Protocolo proprietário do agente
<b>Manutenção</b>	Apenas Control Node precisa de atualização	Agentes em todos os Managed Nodes precisam de atualização
<b>Segurança</b>	Menor superfície de ataque, usa SSH seguro	Agente rodando pode ser um ponto de falha
<b>Complexidade</b>	Mais simples para começar, menos dependências	Maior complexidade inicial de setup e gestão

# Conectando com a Realidade Profissional



## Valorização no Mercado

Profissionais que dominam o gerenciamento de configuração e a automação com ferramentas como o Ansible são essenciais em equipes de DevOps, engenharia de confiabilidade (SRE), administradores de sistemas e arquitetos de nuvem.



## Diferencial Competitivo

A capacidade de automatizar tarefas repetitivas, garantir a consistência de ambientes e integrar segurança desde o início é um diferencial competitivo que impulsiona a eficiência e a inovação nas empresas.



## Impacto Organizacional

Ao dominar o Ansible, você não apenas otimiza seu próprio trabalho, mas também contribui para a resiliência e a escalabilidade da infraestrutura de TI, um pilar fundamental para qualquer organização moderna.

A compreensão do Ansible e de sua arquitetura não é apenas um conhecimento teórico; é uma habilidade prática altamente valorizada no mercado de trabalho. Profissionais que dominam o gerenciamento de configuração e a automação com ferramentas como o Ansible são essenciais em equipes de DevOps, engenharia de confiabilidade (SRE), administradores de sistemas e arquitetos de nuvem. A capacidade de automatizar tarefas repetitivas, garantir a consistência de ambientes e integrar segurança desde o início é um diferencial competitivo que impulsiona a eficiência e a inovação nas empresas.

Ao dominar o Ansible, você não apenas otimiza seu próprio trabalho, mas também contribui para a resiliência e a escalabilidade da infraestrutura de TI, um pilar fundamental para qualquer organização moderna. As tendências de GitOps, DevSecOps e AIOps mostram que a automação inteligente e segura é o caminho, e o Ansible é uma das ferramentas mais eficazes para trilhá-lo.

# Consolidação do Conhecimento

Nesta aula, desvendamos o mundo do Ansible, uma ferramenta poderosa para o gerenciamento de configuração e automação de infraestrutura. Compreendemos a distinção crucial entre provisionamento e gerenciamento de configuração, posicionando o Ansible como o maestro que orquestra o estado desejado dos nossos sistemas. Exploramos sua arquitetura "agentless", que simplifica a implantação e a manutenção, baseada na comunicação segura via SSH entre o Control Node e os Managed Nodes. Demos os primeiros passos práticos na configuração do ambiente e na execução de comandos Ad-Hoc, que nos permitem realizar tarefas rápidas e eficientes em múltiplos servidores. Finalmente, conectamos o Ansible às tendências futuras da IaC, como GitOps, DevSecOps e AIOps, mostrando seu papel central na construção de infraestruturas mais inteligentes, seguras e automatizadas.

## Em Prática

Agora que você entende os fundamentos, comece a experimentar. Instale o Ansible em uma máquina virtual, configure a conexão SSH com outra VM e pratique os comandos Ad-Hoc para instalar pacotes ou verificar serviços. Essa experiência prática solidificará seu aprendizado e o preparará para desafios maiores.

## Próxima Aula

Na próxima aula, "Aula 19 – Inventário: Gerenciando seus Servidores", aprofundaremos um conceito fundamental para o Ansible: o inventário. Você aprenderá como organizar seus servidores em grupos lógicos, facilitando a automação em escala e a aplicação de configurações específicas para diferentes ambientes.

## Recursos Adicionais

- **Documentação Oficial do Ansible:** Para detalhes técnicos e exemplos aprofundados.
- **Livros e Cursos Online sobre Ansible:** Para aprofundar seus conhecimentos e habilidades práticas.
- **Comunidades de DevOps e IaC:** Para trocar experiências e resolver dúvidas com outros profissionais.

# Autoavaliação

## Questão 1

Qual das seguintes afirmações melhor descreve a principal diferença entre provisionamento e gerenciamento de configuração no contexto da Infraestrutura como Código (IaC)?

1

1. Provisionamento lida com a criação de código para aplicações, enquanto gerenciamento de configuração lida com a implantação desse código.
2. Provisionamento foca na criação de recursos de infraestrutura (VMs, redes), enquanto gerenciamento de configuração foca na instalação e manutenção de software e configurações nesses recursos.
3. Ambos os termos são sinônimos e podem ser usados de forma intercambiável na IaC.
4. Provisionamento é uma etapa manual, enquanto gerenciamento de configuração é sempre automatizado.

## Questão 2

A arquitetura "agentless" do Ansible significa que:

2

1. Ele não precisa de conexão de rede para gerenciar servidores remotos.
2. Um software cliente (agente) precisa ser instalado e mantido em cada Control Node.
3. Não é necessário instalar um software cliente (agente) nos Managed Nodes, utilizando protocolos padrão como SSH.
4. O Ansible só pode gerenciar servidores que rodam o mesmo sistema operacional que o Control Node.

## Questão 3

Para que o Ansible possa se comunicar com um Managed Node sem a necessidade de digitar senhas repetidamente, qual é a configuração mais recomendada?

3

1. Desativar completamente a segurança SSH no Managed Node.
2. Configurar um usuário com senha padrão em todos os Managed Nodes.
3. Utilizar chaves SSH, copiando a chave pública do Control Node para o Managed Node.
4. Instalar um agente de autenticação proprietário do Ansible em cada Managed Node.

## Questão 4

Qual das seguintes tendências da IaC é diretamente suportada pelo Ansible ao utilizar o Git como a única fonte da verdade para a infraestrutura, garantindo versionamento e auditabilidade?

4

1. AIOps
2. DevSecOps
3. GitOps
4. Serverless Computing

## Gabarito

1. b)
2. c)
3. c)
4. c)

## Questão Discursiva

Explique como a integração do Ansible com as práticas de DevSecOps pode contribuir para a segurança de uma infraestrutura de TI, citando pelo menos dois exemplos práticos de como essa integração pode ser implementada.