



9ª Semana de Ciência da Computação e Tecnologia da Informação

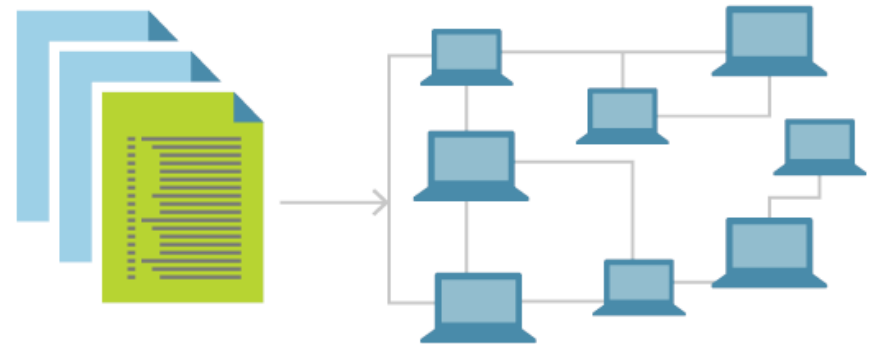


Infraestrutura como código (IaC) e Ansible

Prof. Vinicius Barcelos



A N S I B L E





DevOps e Infraestrutura ágil



“Meu time precisava de um ambiente para a homologação do nosso produto, que já estava com um atraso considerável. Entretanto, ouvi da nossa infraestrutura que a requisição do ambiente de homologação demoraria algumas semanas. A partir deste momento, toda uma cadeia de estresses se estabeleceu de lado a lado, com pressões diversas de pessoas do desenvolvimento e respostas áridas do time de operações. E no final das contas, o nosso time precisou esperar 3 semanas para ter um ambiente liberado. E isso somente tornou o nosso projeto, que já estava ruim, em um pesadelo vivo.”



DevOps e Infraestrutura ágil



- Em 2009 o termo DevOps é citado em uma palestra, onde o palestrante mostra como ele fazia de forma automatizada deploys diários com ajuda da infraestrutura.
- Agilidade em Infraestrutura ganhou grande importância depois da “criação” da filosofia DevOps.
- Com os avanços das últimas duas décadas nas tecnologias de virtualização e containerização, automatizar a infraestrutura de TI, provendo maior agilidade, passou a ser perfeitamente possível.
- Um dos principais pilares do conceito de infraestrutura ágil consiste em ter sua infraestrutura como código.
- **Infraestrutura como código:** a prática de tratar a infraestrutura de TI como se fosse código — exatamente como um software.
- Escreve-se um código (que pode ser feito usando um linguagem de alto nível ou qualquer linguagem descritiva) para gerenciar configurações e/ou automatizar o provisionamento da infraestrutura e implantações.



DevOps e Infraestrutura ágil



- Vantagens:
 - “Subir” um servidor deixa de levar dias/semanas para minutos.
 - Controle de versão do código.
 - Automatização na criação / recriação da infraestrutura.
 - Testes automatizados.
 - Refatoração e Otimização.
- A infraestrutura como código é parte essencial para a prática de **Entrega Contínua** em DevOps.
- Roadmap DevOps: <https://dev.to/joehobot/devops-roadmap-5d0>
- Níveis de maturidade de Infraestrutura como código:
 - <https://marco-mendes.com/2016/12/15/maturidade-devops-infraestrutura-como-codigo/>



Dos scripts até a Infraestrutura como Código (IaC)



- **Quantos servidores um SysAdmin consegue administrar?**
- **Quanto tempo leva para alterar uma linha de um arquivo em todos os servidores da empresa?**
- SysAdmins ao longo dos tempos sempre buscaram automatizar algumas tarefas, principalmente as repetitivas, em seu ambiente de infraestrutura de TI.
- Um pouco de história:
 - Uso de scripts usando **shell script** sempre foi muito forte na automatização de tarefas em ambientes Linux/Unix.
 - Em 2005 surge o Puppet e o Git.
 - Em 2006 a Amazon lança sua nuvem pública AWS.
 - Em 2009 o termo DevOps é citado em uma palestra.
 - Em 2012 surge o Ansible.
 - Em 2013 o Docker.



Infraestrutura ágil e Infraestrutura como Código (IaC)



- Partes importantes da Infraestrutura como Código dentro da ideia de agilidade em Infraestrutura (<https://4devops.com.br/e-book/>)

AGILE INFRASTRUCTURE	ETAPA 1.1 INFRAESTRUTURA PADRONIZADA	ETAPA 1.2 MUDANÇAS CONFIÁVEIS	ETAPA 1.3 INFRAESTRUTURA EM CONTÊINERES	ETAPA 1.4 MONITORAMENTO FOCO EM NEGÓCIO
	Provisionamento automatizado de VMs	Fluxo automatizado para aplicar mudanças	Implantação de Contêineres Docker	SLAs
	Gestão de configuração	Ambiente de Homologação idêntico a Produção	Red Hat OpenShift (Infra PaaS)	Planejamento de capacidade
	Infraestrutura versionada	Testes de infraestrutura automatizados	AWS Elastic Cluster Suite (ECS)	Indicador chave de desempenho (KPI)
	Monitoramento de SO	Rollbacks automatizados	Escalonamento automático	Engenharia de Confiabilidade do site (SER)
		Monitoramento de serviços	Monitoramento de Contêineres	Integração com Sistema de Tickets
		Monitoramento de Appliance		Integração com Sistema de Inventário
		Monitoramento de Aplicações		Dashboards visões Operacionais
				Dashboards visões de Negócio

Principais ferramentas de IaC

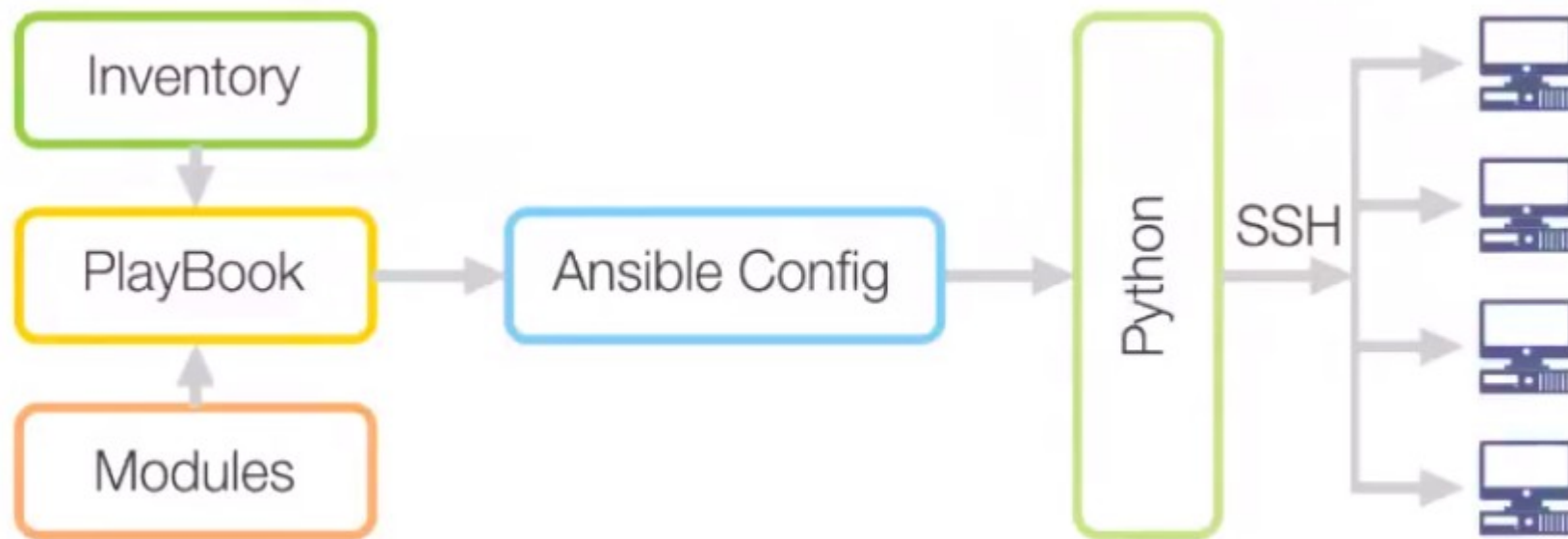


	Initial release	Current version
Puppet	2005	6.0.9
Chef	2009	12.19.31
CloudFormation	2011	2010-09-09
SaltStack	2011	2019.2.0
Ansible	2012	2.5.5
Heat	2012	12.0.0
Terraform	2014	0.12.0

Uma comparação do prazo de vencimento do IAC a partir de maio de 2019. Clique na imagem em tamanho normal.



- Mantido pela Red Hat + comunidade, é uma das principais ferramentas de automatização e gerenciamento de configuração.
- Sua principal característica (que diferencia dos demais) é que não existe a necessidade de instalação de um cliente do Ansible nos hosts clientes.
- O Ansible utiliza a conexão SSH para automatizar tarefas nos hosts.
- Obs.: É necessário que os hosts alvo tenham o Python instalado.





Vantagens do Ansible



- É possível automatizar qualquer distribuição que tenha o suporte ao Python e ao SSH.
- Não requer a instalação de um app cliente nos hosts alvo (nós).
- Seus Playbooks utilizam linguagem de fácil entendimento, escritos em YAML, sendo autoexplicativos.
- Diversos módulos disponíveis, que facilitam a configuração e garantem um padrão de qualidade.
- Possibilidade de adição de módulos.
- Também permite gerenciar máquinas Windows, através do PowerShell.



Instalação do Ansible



- No Debian 10 e no Ubuntu 18.04, basta instalar o pacote ansible.

```
# apt-get update
```

```
# apt-get install ansible
```
- Devemos gerar a chave SSH para o usuário atual, que acessará os hosts alvos sem precisar de senha.

```
# ssh-keygen
```
- Agora devemos copiar a chave pública para máquina do Ansible para o host alvo.
 - ```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@192.168.0.1
```
- Em caso de máquinas Debian/Ubuntu, utilize o ssh para garantir que o python está instalado no host alvo.  

```
ssh root@192.168.0.1 'apt-get update && apt-get install python -y'
```



# Estrutura padrão de um playbook



---

```
- hosts: alvos
 remote_user: root
 vars:
 variavel1: valor1
 Variavel2: valor2
 tasks:
 - name: Descrição da tarefa 1
 modulo:
 atributo1: valor1
 atributo2: valor2
 - name: Descrição da tarefa 2
 modulo:
 atributo1: valor1
 atributo2: valor2
```



# Principais Módulos do Ansible



- **TODOS OS MÓDULOS:**
  - [https://docs.ansible.com/ansible/latest/modules/modules\\_by\\_category.html](https://docs.ansible.com/ansible/latest/modules/modules_by_category.html)
- **APT:** [https://docs.ansible.com/ansible/latest/modules/apt\\_module.html](https://docs.ansible.com/ansible/latest/modules/apt_module.html)
- **DEBCONF:** [https://docs.ansible.com/ansible/latest/modules/debconf\\_module.html](https://docs.ansible.com/ansible/latest/modules/debconf_module.html)
- **FILE:** [https://docs.ansible.com/ansible/latest/modules/file\\_module.html](https://docs.ansible.com/ansible/latest/modules/file_module.html)
- **SHELL:** [https://docs.ansible.com/ansible/latest/modules/shell\\_module.html](https://docs.ansible.com/ansible/latest/modules/shell_module.html)
- **COPY:** [https://docs.ansible.com/ansible/latest/modules/copy\\_module.html](https://docs.ansible.com/ansible/latest/modules/copy_module.html)
- **SERVICE:** [https://docs.ansible.com/ansible/latest/modules/service\\_module.html](https://docs.ansible.com/ansible/latest/modules/service_module.html)
- **GIT:** [https://docs.ansible.com/ansible/latest/modules/git\\_module.html](https://docs.ansible.com/ansible/latest/modules/git_module.html)
- **TEMPLATE:** [https://docs.ansible.com/ansible/latest/modules/template\\_module.html](https://docs.ansible.com/ansible/latest/modules/template_module.html)