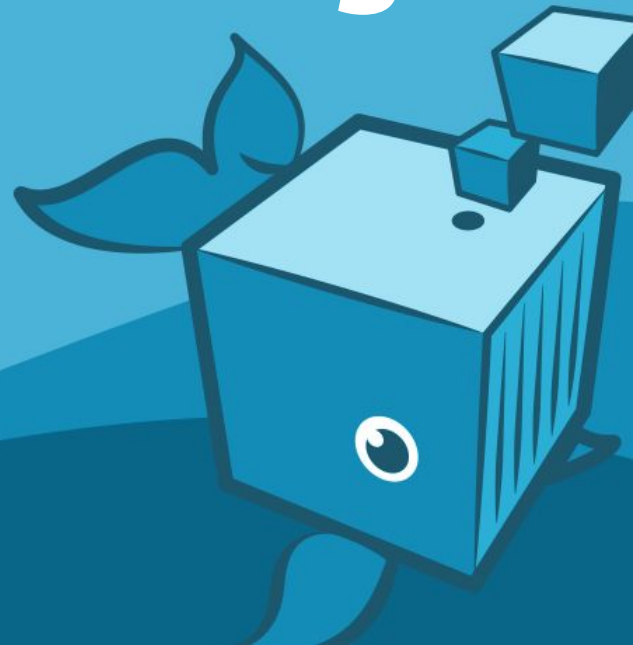


Tornando a infra mais ágil



Cristhian Bicca

- Infraestrutura 8 anos
- Co-Fundador MundoDocker
- Todas as soluções a um container de distância

Servidor

Servidor

- ISO (CD, DVD, Pen drive)

Servidor

- ISO (CD, DVD, Pen drive)
- Criar a máquina

Servidor

- ISO (CD, DVD, Pen drive)
- Criar a máquina
- Instalar S.O

Servidor

- ISO (CD, DVD, Pen drive)
- Criar a máquina
- Instalar S.O
- IP

Servidor

- ISO (CD, DVD, Pen drive)
- Criar a máquina
- Instalar S.O
- IP
- Instalar softwares

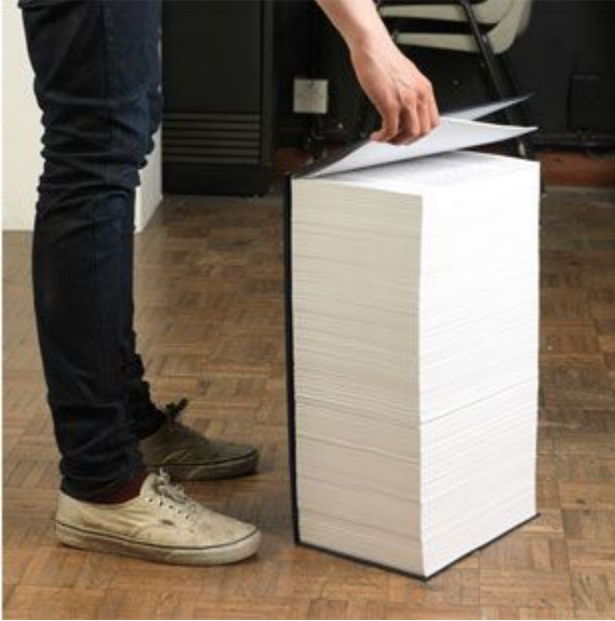
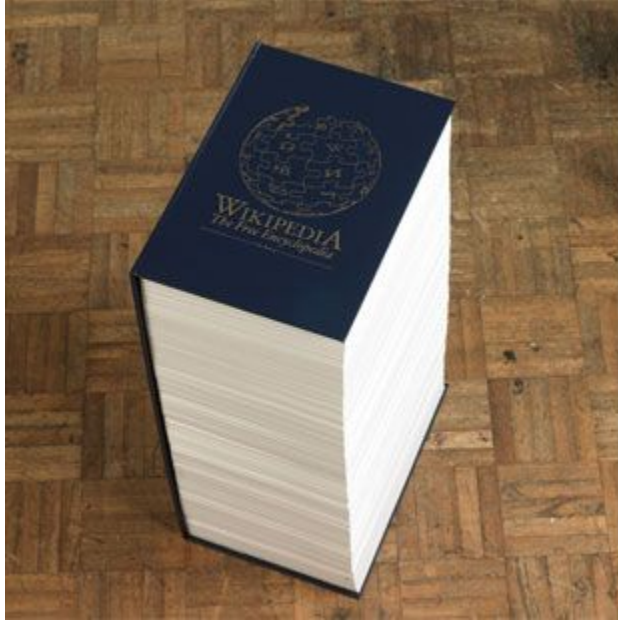
Servidor

- ISO (CD, DVD, Pen drive)
- Criar a máquina
- Instalar S.O
- IP
- Instalar softwares
- Entregar máquina

Intel(R) Boot Agent GE v1.2.22

Copyright (C) 1997-2004, Intel Corporation

CLIENT MAC ADDR: 90 E2 BA 79 26 CC GUID
DHCP. /





CHEF™



HashiCorp

Terraform



A N S I B L E



Microsoft®

System Center
Orchestrator



Fabric



CloudFormation



SALTSTACK



puppet

Versionável

Controlável

Rastreável

Testável

“Alinhável” (Dev ↔ Ops)



Ambiente

- Hyper-v
- VMs Linux
- VMs Windows
- 60 máquinas por dia.

System Center 2012 Orchestrator Runbook Designer

Actions: Edit Options View Help

Refresh Run Check In Check Out Undo Check Out Runbook Tester Orchestration Console

Connections: Operations Management | 4.2.1 Update Alert Operations Management | 4.2.2 Create Alert

Activities

- System
- Scheduling
- Monitoring
- File Management
- Email
- Notification
- Utilities
- Active Directory
- Social Media IP
- Text File Management
- SC 2012 Data Protection Manager
- SC 2012 Configuration Manager
- SC 2012 Operations Manager
- Create Alert
- Get Alert
- Get Monitor
- Monitor Alert
- Monitor State
- Start Maintenance Mode
- Stop Maintenance Mode
- Update Alert
- SC 2012 Service Manager
- SC 2012 Virtual Machine Manager
- Runbook Control

SCOR0013RC

Runbooks

- SC2012 Solutions
 - 0.0 Simple Example
 - 1.0 Cloud Management
 - 1.1.0 VM Checkpoint Management
 - 1.2.0 Virtual Disk Management
 - 1.3.0 Virtual Network Management
 - 1.4.0 VM Management
 - 2.0 Service Management
 - 2.1.0 Change Management
 - 2.2.0 CI Management
 - 2.3.0 Incident Management
 - 3.0 Data Protection
 - 3.1.0 Recovery Management
 - 3.2.0 Protection Management
 - 4.0 Operations Management
 - 4.1.0 Maintenance Mode Management
 - 4.2.0 Alert Management**
 - 5.0 Configuration Management
 - 5.1.0 Collection Management
 - 5.2.0 Deployment Management
 - 5.3.0 Client Management
 - 6.0 Administrative Tasks
- Computer Groups
- Runbook Servers
- Global Settings

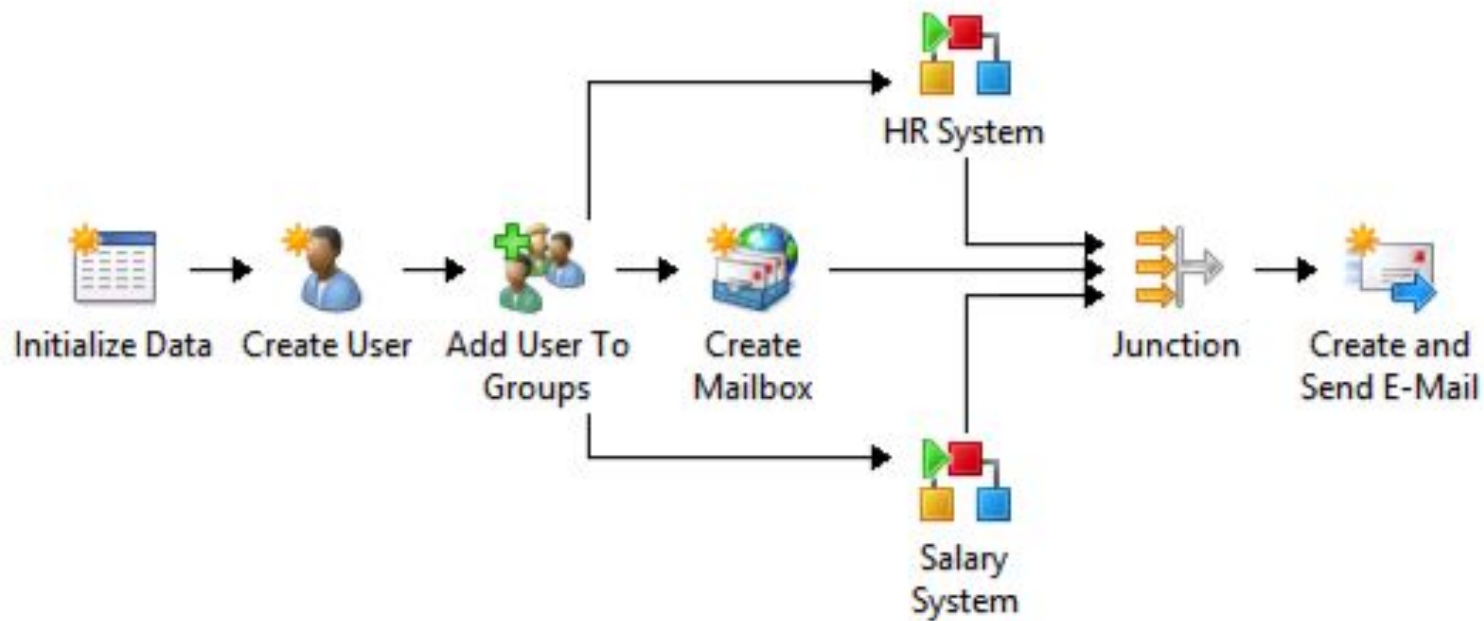
Diagram:

```

graph LR
    Start([Start]) --> InitAlertUpdate[Initiate Alert Update]
    InitAlertUpdate --> GetAlertName[Get Alert - Name]
    InitAlertUpdate --> GetAlertCategory[Get Alert - Category]
    InitAlertUpdate --> GetAlertOwner[Get Alert - Owner]
    InitAlertUpdate --> GetAlertWebSite[Get Alert - WebSite]
    GetAlertName --> PassThrough[PassThrough]
    GetAlertCategory --> PassThrough
    GetAlertOwner --> PassThrough
    GetAlertWebSite --> PassThrough
    PassThrough --> UpdateAlert[Update Alert]
    UpdateAlert --> AlertUpdated[Alert Updated]
  
```

Log

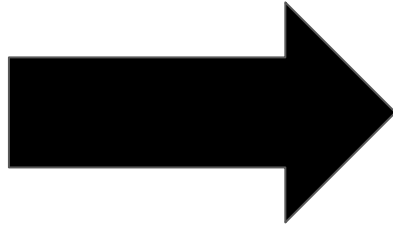
Log History Audit History Events





©2012 Peanut Worldwide

snoopy.example.com



web230.ex.com



web840.ex.com



web936.ex.com



web745.ex.com

Terraform

- Open Source
- Hashicorp (vagrant, consul, packer, vault)
- Iniciado em 2014
- GO
- Plugins

Providers

Terraform is used to create, manage, and update infrastructure resources such as physical machines, VMs, network switches, containers, and more. Almost any infrastructure type can be represented as a resource in Terraform.

A provider is responsible for understanding API interactions and exposing resources. Providers generally are an IaaS (e.g. AWS, GCP, Microsoft Azure, OpenStack), PaaS (e.g. Heroku), or SaaS services (e.g. Terraform Enterprise, DNSimple, CloudFlare).

Use the navigation to the left to find available providers by type or scroll down to see all providers.

Alicloud	Archive	Arukas
AWS	Azure	Azure Stack
Bitbucket	Brightbox	CenturyLinkCloud
Chef	Circonus	Cloudflare
CloudScale.ch	CloudStack	Cobbler
Consul	Datadog	DigitalOcean
DNS	DNSMadeEasy	DNSimple
Docker	Dyn	External
Fastly	FlexibleEngine	GitHub
Gitlab	Google Cloud	Grafana
Heroku	Hetzner Cloud	HTTP
HuaweiCloud	Icinga2	Ignition

TERRAFORM DEMO

Ansible

- Provisionador de ambiente
- Gerência de configuração
- Orquestração
- Automação de TI
- Desenvolvido em Python



Ansible

2.6

For previous versions, see the [documentation archive](#).

INSTALLATION, UPGRADE & CONFIGURATION

[Installation Guide](#)

[Configuring Ansible](#)

[Ansible Porting Guides](#)

USING ANSIBLE

User Guide

[Ansible Quickstart](#)

[Getting Started](#)

[Working with Command Line Tools](#)

[Introduction To Ad-Hoc Commands](#)

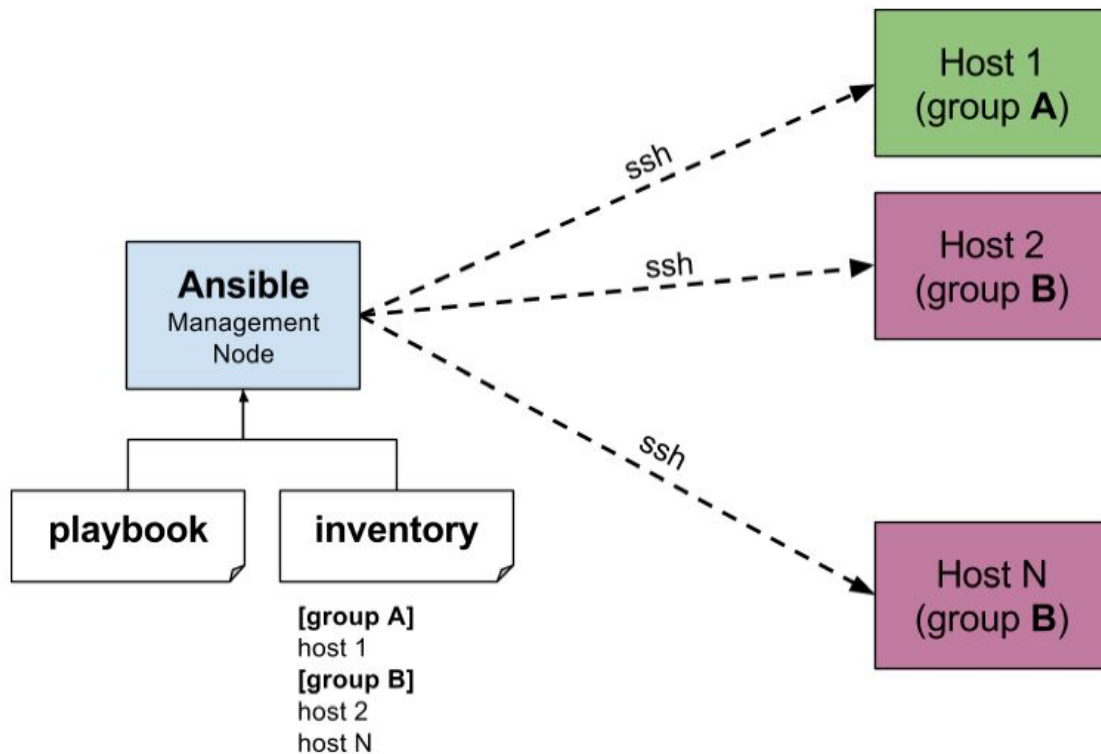
[Working with Inventory](#)

[Working With Dynamic Inventory](#)

[Working With Playbooks](#)

Module Index

- [All modules](#)
- [Cloud modules](#)
- [Clustering modules](#)
- [Commands modules](#)
- [Crypto modules](#)
- [Database modules](#)
- [Files modules](#)
- [Identity modules](#)
- [Inventory modules](#)
- [Messaging modules](#)
- [Monitoring modules](#)
- [Net Tools modules](#)
- [Network modules](#)
- [Notification modules](#)
- [Packaging modules](#)
- [Remote Management modules](#)
- [Source Control modules](#)
- [Storage modules](#)
- [System modules](#)
- [Utilities modules](#)
- [Web Infrastructure modules](#)
- [Windows modules](#)



Porque Ansible?

- Curva de aprendizado pequena
- Utiliza uma DSL
- Se comunica por SSH → SSH Keys
- Sem daemons ou outros softwares do lado dos nós gerenciados
- Linha de comando → um comando simples
- Integrações com quase tudo
- Customização de módulos

Inventory

```
[webservers]  
foo.example.com  
bar.example.com
```

```
[dbservers]  
one.example.com  
two.example.com  
three.example.com
```

Playbook

```
---
- hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
    remote_user: root
  tasks:
    - name: ensure apache is at the latest version
      yum:
        name: httpd
        state: latest
    - name: write the apache config file
      template:
        src: /srv/httpd.j2
        dest: /etc/httpd.conf
      notify:
        - restart apache
    - name: ensure apache is running (and enable it at boot)
      service:
        name: httpd
        state: started
        enabled: yes
  handlers:
    - name: restart apache
      service:
        name: httpd
        state: restarted
```



Seleção

Ordenação

Idempotência

HostVars x GroupVars

```
[atlanta]  
host1 http_port=80 maxRequestsPerChild=808  
host2 http_port=303 maxRequestsPerChild=909
```

```
[atlanta:vars]  
ntp_server=ntp.atlanta.example.com  
proxy=proxy.atlanta.example.com
```

Jinja2

```
{{ lookup('env', 'MY_USER') | default('admin', true) }}
```

vars:

```
redis_maxmemory: 2mb  
redis_port: 4096
```

tasks:

- name: Ensure Redis is configured
 template:
 src: redis.conf.j2
 dest: /path/to/redis.conf

```
## redis.conf.j2:
```

```
maxmemory {{ redis_maxmemory }}  
port {{ redis_port }}
```

Module

- name: add a single line
command: echo "Another comment" >> /test/test.conf

Abstração
por
módulo

- name: add a single line
lineinfile: dest=/test/test.conf
 regexp='^Another'
 insertafter='^#Another'
 line='Another comment'
 state=present

Convenção de diretório

```
production          # inventory file for production servers
staging             # inventory file for staging environment

group_vars/
  group1            # here we assign variables to particular groups
  group2            # ""
host_vars/
  hostname1         # if systems need specific variables, put them here
  hostname2         # ""

library/            # if any custom modules, put them here (optional)
module_utils/       # if any custom module_utils to support modules, put them
here (optional)
filter_plugins/     # if any custom filter plugins, put them here (optional)

site.yml            # master playbook
webservers.yml     # playbook for webserver tier
dbservers.yml       # playbook for dbserver tier
```

Convenção de diretório

```
roles/
  common/                # this hierarchy represents a "role"
    tasks/              #
      main.yml          # <-- tasks file can include smaller files if warranted
    handlers/           #
      main.yml          # <-- handlers file
    templates/          # <-- files for use with the template resource
      ntp.conf.j2       # <----- templates end in .j2
    files/              #
      bar.txt           # <-- files for use with the copy resource
      foo.sh            # <-- script files for use with the script resource
    vars/               #
      main.yml          # <-- variables associated with this role
    defaults/           #
      main.yml          # <-- default lower priority variables for this role
    meta/               #
      main.yml          # <-- role dependencies
    library/            # roles can also include custom modules
    module_utils/       # roles can also include custom module_utils
    lookup_plugins/    # or other types of plugins, like lookup in this case

webtier/                # same kind of structure as "common" was above, done for the webtier role
monitoring/            # ""
fooapp/                 # ""
```

Ansible - Ad-Hoc

- Coisas rápidas que não precisa ser criado playbooks
- `ansible web -m shell -a "date"`
- `ansible web -m yum -a "name=httpd state=present"`

Perguntas?



Stickers
Devs



Stickers
Devs



Stickers
Devs



Stickers
Devs

Contatos



cristhian.bicca@gmail.com



@mundodocker



facebook.com/mundodocker



<http://www.mundodocker.com.br>