

# *AUTOMATING INFRASTRUCTURE* *with* **ANSIBLE**

*Hands on presentation by*  
*matyas@justrocket.de*

*23. apr. 2018*

# In the beginning...

- Manual server setup
- Manual installation and configuration
- Heap of shell scripts

# ... there were problems...

- What is actually on the machine?
- Keeping track of installed versions & configurations
- Setting up new machines
- Everybody had his / her own way of doing things

... that needed to be solved...

- Structured Infrastructure management
- Documented versioning
- Replayability of configurations
- Testability of installations
- Small learning curve

... by



ANSIBLE

- Straightforward setup
- Easy to learn
- Scalable
- Large team features
- Enterprise backing & support (\$)
- Large & active community



484



734



7.05K



29.8K

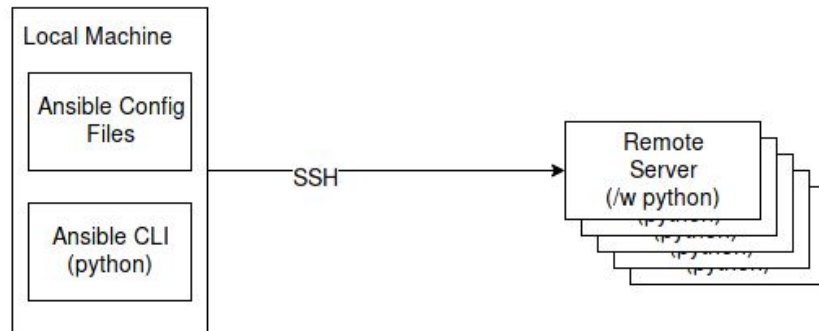


10.9K



1 day ago

# How does it work?



# Key concepts

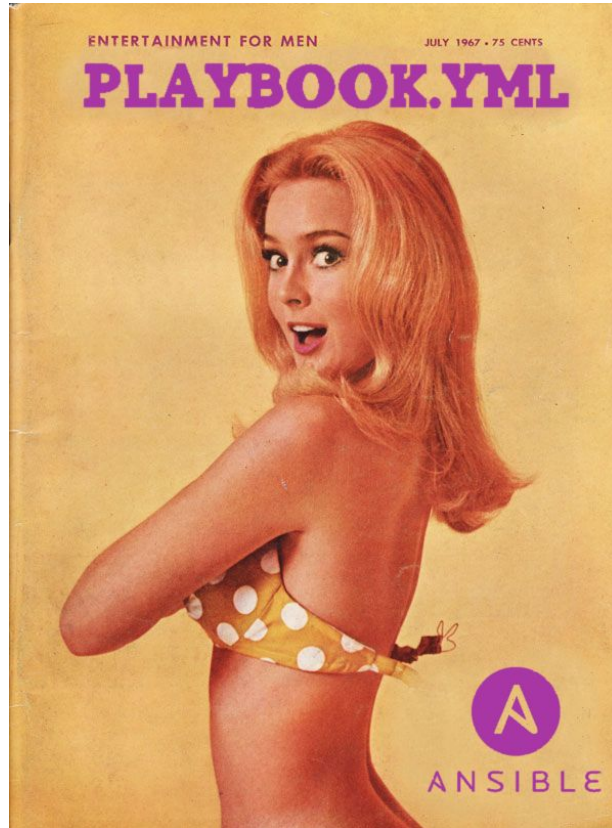
- **Inventory:** list of servers
- **Playbook:** describe how to set up servers
- **Modules:** the tools using which help you execute tasks

# Inventory

```
all:
  hosts:
    backup:
      ansible_host: 192.168.123.123
      ansible_user: root
  children:
    production_servers:
      hosts:
        mariadb:
          ansible_host: 192.168.123.123
          ansible_user: dbadmin
        nginx:
          ansible_host: 192.168.123.124
          ansible_user: appadmin
```



# Playbooks and modules



# Playbooks and modules

---

```
- name: User friendly description of play
  hosts: all
  tasks:
    - name: Install a file manager
      apt:
        package: mc
        state: present
    - name: Check out a project
      git:
        repo: git@gitlab.com:justrocket/dev
        accept_hostkey: yes
        dest: /opt/dev
```

# Core Modules

Cloud / Clustering / Commands / Crypto

Database / Files / Identity / Inventory

Messaging / Monitoring / Net Tools / Network

Notification / Packaging / Remote Management

Source Control / Storage / System / Utilities

Web Infrastructure / Windows

# Variables, basic loops

```
---  
  
- name: Backup data  
  hosts: mariadb_server  
  variables:  
    - label: "important_dump"  
  tasks:  
    - name: Create database dumps  
    mysql_db:  
      state: dump  
      name: {{ item }}  
      target: /tmp/{{ inventory_hostname }}_{{ item }}_{{ label }}.sql  
    with_items:  
      - app_db_X  
      - app_db_Y  
      - app_db_Z
```

# Shell module

---

```
- name: Init db cluster
  hosts: mariadb_server
  tasks:
    - name: Copy script to server
      file:
        src: files/cluster_init_script.sh
        dest: /opt/scripts/cluster_init_script.sh
    - name: Run init script
      shell: "/opt/scripts/cluster_init_script.sh"
```

# Organizing: Tags

---

```
- name: Backup data
  hosts: mariadb_server
  tasks:
    - name: Create database dumps
      tags: backup, weekly
      mysql_db:
        state: dump
        name: {{ item }}
        target: /tmp/{{ inventory_hostname }}_{{item}}.sql
      with_items:
        - app_db_X
        - app_db_Y
        - app_db_Z
```

# Organizing: Roles



## ROLES

common

file\_server

database\_server

app\_server

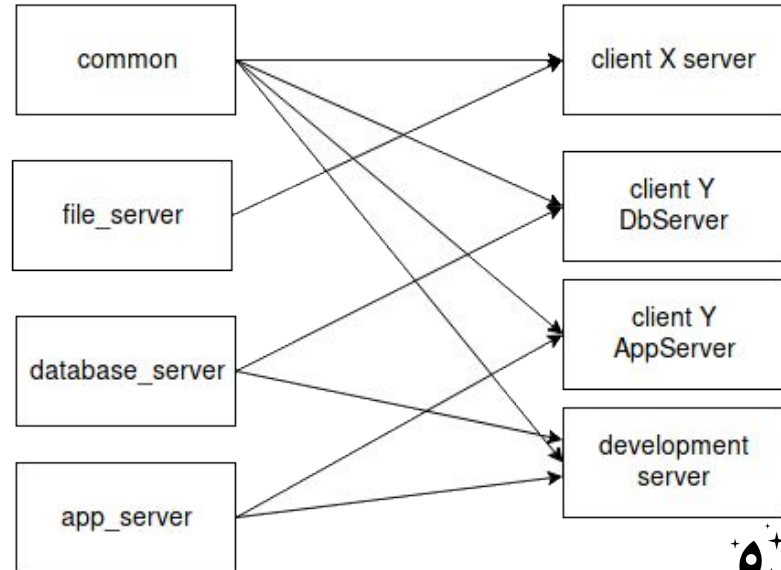
## SERVERS

client X server

client Y  
DbServer

client Y  
AppServer

development  
server

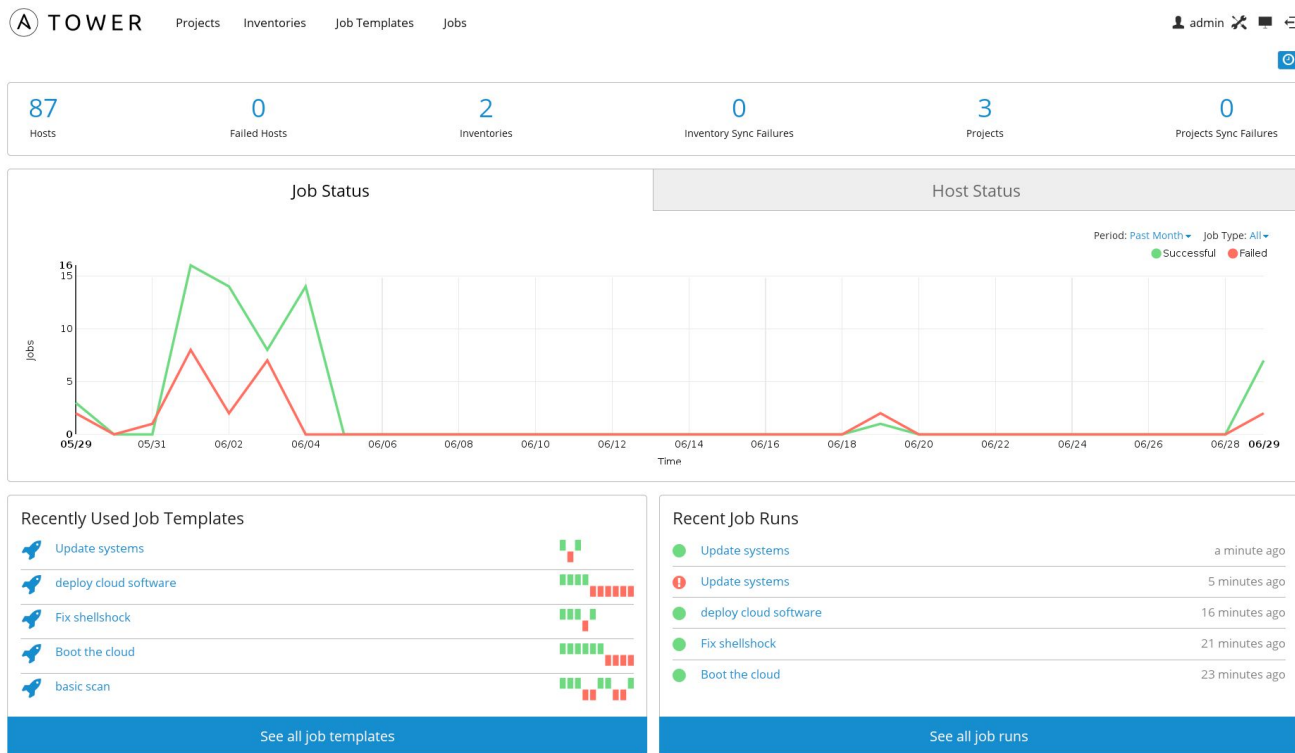


# DEMO

<https://github.com/justrocket/ansible-presentation>



# Going Enterprise with Tower or AWX



# Questions.

?

# Thank you



email: [matyas@justrocket.de](mailto:matyas@justrocket.de)

twitter: @albertmatyi

presentation:  
[git@github.com:justrocket/ansible-presentation.git](https://github.com/justrocket/ansible-presentation)

