

AUTOMATE EVERYTHING

Anton Kasperovich



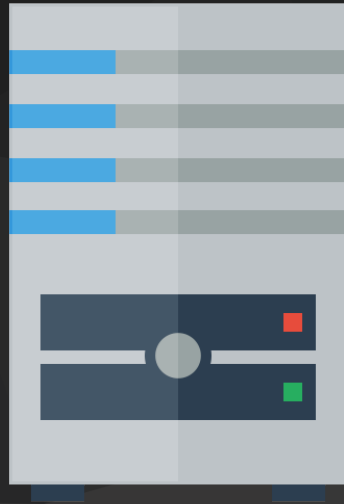
www.axiomadev.com

History

- Small company
- Mid-size projects
- No developers rotation between projects
- Default stack - LAMP

History

1 server



N projects

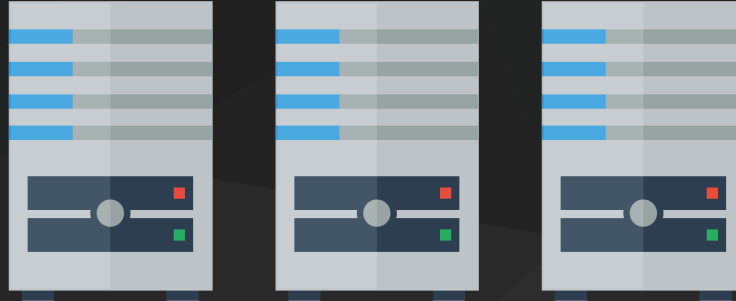
Demo

Present

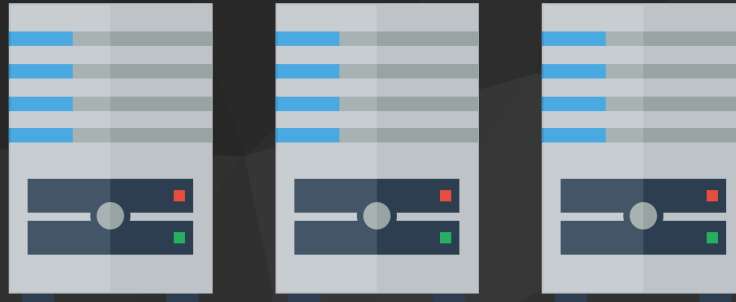
- Mid company
- Mid & Big projects
- Developers' rotation between projects
- Most projects has it's own stack

Present

N servers

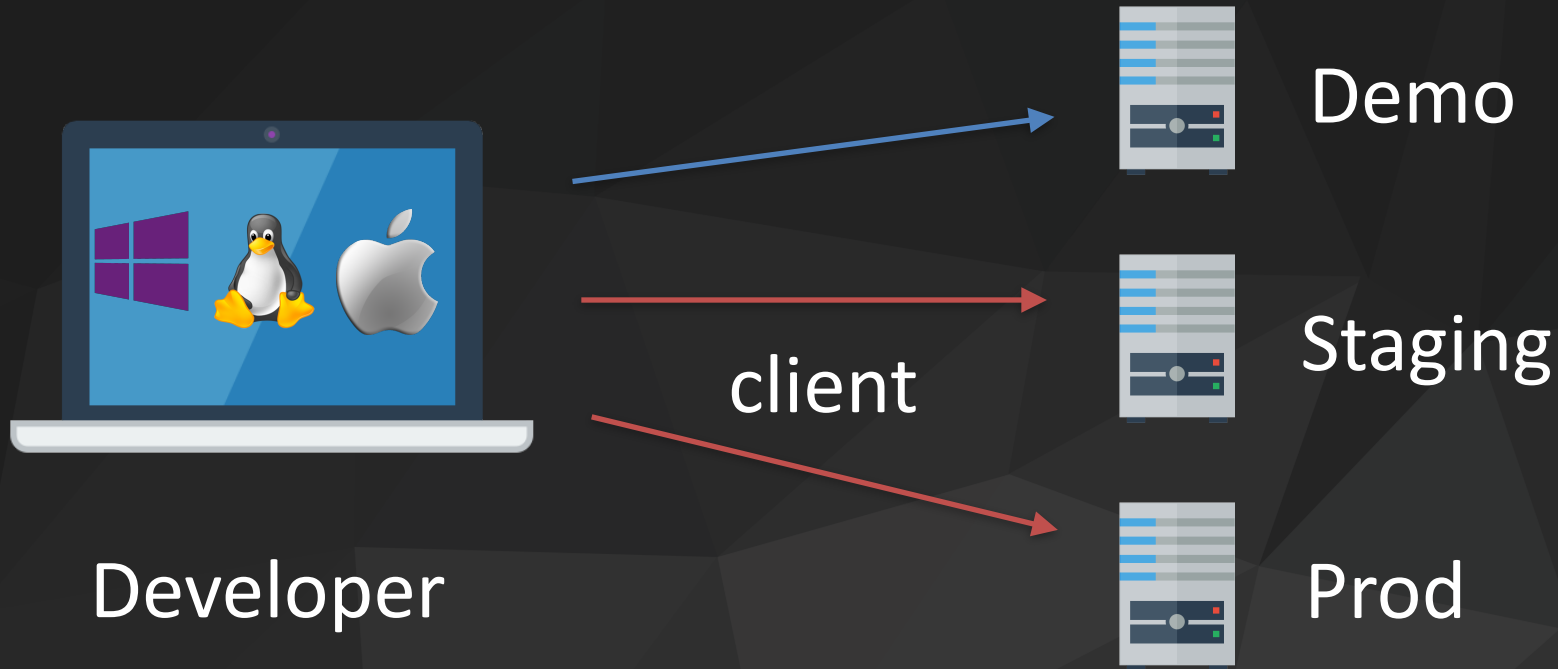


N projects



Demo

One project delivery cycle



Problem

1. Each developer has own workspace
 - OS / PHP v / MySQL or MariaDB and so on...
2. ~3 servers for 1 project
3. Deploy to multiple hosts, specific scenario

Solution?



VAGRANT

Each developer has own workspace

But

vagrant provision

we still should install somehow all software on
VirtualBox

BUT, again, plus "server"

- Demo server
- Staging server
- Production server
- Vagrant (developers PC)
- * server
- ...



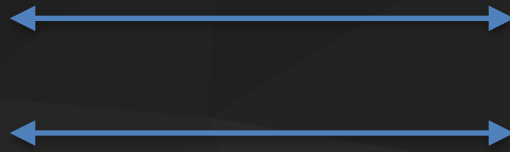
Ansible is the simplest way to automate.

Automate

- Vagrant
- Configuration management
- Application deployment
- Cloud provisioning

Symfony2

- Yaml
- Twig



Ansible

- Yaml
- Jinja2

AGENTLESS

Ansible uses SSH instead of agents.

Install

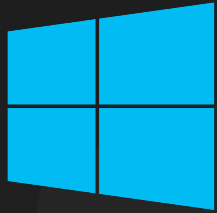


```
pip install -U ansible
```

Windows?

Windows isn't supported for the control machine

Use Docker



Windows



Docker and Microsoft partner to bring container applications across platforms

Ansible provisioner



```
Vagrant.configure("2") do |config|  
  config.vm.provision "ansible" do |ansible|  
    ansible.playbook = "playbook.yml"  
  end  
end
```

Ansible + Vagrant Windows



```
is_windows = (RbConfig::CONFIG['host_os'] =~ /mswin|mingw|
cygwin/)
if is_windows
  config.vm.provision "shell" do |sh|
    sh.path = "app/ansible/JJG-Ansible-Windows/windows.sh"
    sh.args = "ansible/site.yml"
  end
end
```

How it works?

Ansible works by connecting to your nodes and pushing out small programs

```
/usr/bin/python /home/vagrant/.ansible/tmp/ansible-tmp-1430988475.17-4862925373856/apt;
```

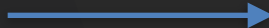
```
rm -rf /home/vagrant/.ansible/tmp/ansible-tmp-1430988475.17-4862925373856/
```

How it works?

```
$ ansible all -i '191.236.89.219,' -m ping
```

```
191.236.89.219 | success >> {  
  "changed": false,  
  "ping": "pong"  
}
```

Inventory, INI file

- Connection type
- Host Variables
- Group Variables 
- Nesting
- `ansible_connection=local`
- `timezone=Europe/Riga`
- `[dbs:vars]`
- `[*:children]`

Inventory

```
#azure-hosts
```

```
[US]
```

```
azure-node-us ansible_ssh_host=191.236.89.219 ansible_ssh_user=devclub
```

```
[EU]
```

```
azure-node-eu ansible_ssh_host=104.41.211.40 ansible_ssh_user=azureuser  
ansible_ssh_port=2233
```

Inventory

```
$ ansible -i inventory -m ping all
```

```
azure-node-us | success >> {  
  "changed": false,  
  "ping": "pong"  
}
```

```
azure-node-eu | success >> {  
  "changed": false,  
  "ping": "pong"  
}
```

```
$ ansible -i inventory -m ping EU
```

```
azure-node-eu | success >> {  
  "changed": false,  
  "ping": "pong"  
}
```


Playbooks

Contain required tasks to configure systems and deploy

- YAML
- Declaratively define your OS/App configuration
- Each group of tasks is a play
- Collection of tasks using modules

Playbook

```
# devclub.yml
```

```
---
```

```
- hosts: all
```

```
  sudo: yes
```

```
  tasks: ...
```

```
- hosts: EU
```

```
  tasks: ...
```

Facts

This module is automatically called by playbooks to gather useful variables about remote hosts that can be used in playbooks.

```
ansible all -i azure_hosts -m setup
```

Tasks

- name: Update apt

 - sudo: yes

 - apt: update_cache=yes

- name: Check or build files image

 - docker_image: path="/files-container" name="files" state=present

Handlers

tasks:

- Change NGINX config file

notify:

- restart nginx

handlers:

- name: restart nginx

service:

name: nginx

state: restarted

Task



Handler



Variables

vars:

```
npm_env: "{{ ansible_env.PATH }}:/var/www/node_modules/.bin"
```

```
ruby_gems:
```

- sass
- compass

Loops

tasks:

- name: Dump multiple databases

command: mongodump --db {{ item.name }} -u {{ item.user }}

with_items:

- { name: 'db1', user: 'test' }

- { name: 'db2', user: 'noo-root' }

← list of hashes

Conditionals

tasks:

- name: Dump database

command: mongodump --db {{ mongodb_name }}

when: dump_path.stat.exists != True

Playbooks hosts

```
# file: staging.yml
```

```
- hosts: staging
```

```
roles:
```

```
- build-tools
```

```
- init
```

```
# file: setup.yml
```

```
- include: staging.yml
```

```
- include: production.yml
```

```
ansible-playbook setup.yml --limit staging
```

Play!

```
TASK: [AbdoulNdiaye.Blackfire | Install the blackfire-php package] *****
```

```
skipping: [construct]
```

```
TASK: [AbdoulNdiaye.Blackfire | Create agent configuration] *****
```

```
ok: [construct]
```

```
TASK: [constructionHL | Install bower dependencies] *****
```

```
failed: [construct] => {"failed": true, "parsed": false}
```

```
PLAY RECAP *****
```

```
construct      : ok=64  changed=0  unreachable=0  failed=1
```

Playbooks

```
- hosts: all
  gather_facts: true

  vars:
    init_packages:
      - git
      - curl
      - rsync
    ruby_gems:
      - { name: 'sass', version: '3.4.9' }
      - { name: 'compass', version: '0.12.6' }
    npm_env: "{{ ansible_env.PATH }}:/var/www/node_modules/.bin"
    ruby_gems_env: "{{ ansible_env.PATH }}:/home/vagrant/.gem/ruby/1.9.1/bin"
    db_dump_path: /var/www/dump.sql

    db_name: axiomadev
    db_user: axiomadev_user
    db_pass: axiomadev_pass

  vars_files:
    - vars/php.yml
    - vars/nginx.yml
    - vars/mysql.yml
    - vars/nodejs.yml
    - vars/composer.yml
    - vars/blackfire.yml

  tasks:
    - include_vars: vault.yml

    - name: Check dump folder exists
      stat: path={{ mongodb_dump }}
      register: dump_path

    - name: Dump database
      command: mongodump --db {{ mongodb_name }} --port {{ mongodb_port }} -p {{ mongodb_pass }} -u {{ mongodb_user }} --out {{ mongodb_dump }}
      when: dump_path.stat.exists != True and dump_path.stat.isdir is not defined

    - name: Check dump archive exists
      stat: path="{{ mongodb_dump_path }}.tar.gz"
      register: dump_archive_path

    - name: Create dump archive
      command: tar -pczf {{ mongodb_dump }}.tar.gz {{ mongodb_dump }}/
```

Variable File Separation

vars_files:

- vars/php.yml
- vars/nginx.yml
- vars/mysql.yml
- vars/nodejs.yml
- vars/composer.yml
- vars/blackfire.yml

vars/php.yml

```
php_enable_php_fpm: true
php_enable_webserver: true
php_webserver_daemon: "nginx"
php_memory_limit: "512M"
php_date_timezone: "Europe/Riga"
php_packages:
  - php5-redis
  - php5-imagick
```

Playbook Roles

Roles are ways of automatically loading certain vars_files, tasks, and handlers based on a known file structure.

```
roles/  
  init/  
    files/  
    templates/  
    tasks/  
    handlers/  
  vars/  
  defaults/  
  meta/
```

Playbook Roles

```
---
```

```
- hosts: all
```

```
  roles:
```

```
    - init
```

```
    - symfony
```

Tags

```
---
```

```
- hosts: all
```

```
roles:
```

```
- init
```

```
- { role: mongo, v: 3.0, tags: [ 'mongo', 'dbs' ] }
```

```
ansible all -i hosts backup.yml --tags "mongo"
```


Check Mode ("Dry Run")

```
ansible-playbook -i hosts backup.yml -v --check --diff
```

```
{"msg": "check mode not supported for command"}
```

Start and Step

Start at task:

```
ansible-playbook playbook.yml --start-at-task="Install PHP"
```

Step:

```
ansible-playbook playbook.yml --step
```

Perform task: Check dump folder exists (y/n/c):

Ansible Vault

The vault feature can encrypt any structured data file used by Ansible.

```
ansible-playbook -i hosts backup.yml --ask-vault-pass
```

```
$ANSIBLE_VAULT;1.1;AES256
```

```
36306339666534646231633535363433393865373730386436626238663632393732393363373635
```

```
3436316132346532363136366537343765356262326330320a663065633061336636353164333861
```

```
37633939653331343861343666356534356532366365376331376537326436623333646139313065
```

```
6534376234613239630a6135643239346339663230
```

Ansible Modules

- Cloud Modules
- Commands Modules
- Database Modules
- Files Modules
- Inventory Modules
- Messaging Modules
- Monitoring Modules
- Network Modules
- Notification Modules
- Packaging Modules
- Source Control Modules
- System Modules
- Utilities Modules
- Web Infrastructure Modules
- Windows Modules

Templates, Jinja2

Customizable files destined for managed machines

```
- template: src=nginx/src.j2 dest=/etc/nginx.conf
tags:
  - nginx
```

File: nginx/src.j2

```
server {  
    listen {{ nginx_listen }};  
    server_name {{ inventory_hostname }};  
    root {{ nginx_root_path }};  
  
    location / {  
        try_files $uri @{{ nginx_backend_name }};  
    }  
}
```

Packaging Modules

- name: Install Sass and Compass ruby gems

```
gem: name={{ item.name }} version={{ item.version }}  
state=present
```

```
with_items:
```

```
- { name: 'sass', version: '3.4.9' }
```

```
- { name: 'compass', version: '0.12.6' }
```

Packaging Modules

```
---
```

```
tasks:
```

```
- name: Install bower dependencies
```

```
  sudo: no
```

```
  environment:
```

```
    PATH: "{{ ansible_env.PATH }}:{{ npm_env }}"
```

```
  bower: path=/var/www
```


Files Modules

```
---
```

```
tasks:
```

```
- name: Check dump archive exists
```

```
  stat: path="{{ mongodb_dump_path }}.tar.gz"
```

```
  register: dump_archive_path
```

```
- debug: msg="MongoDB Dump already exists"
```

```
  when: dump_archive_path.stat.exists
```

No module? Use command or shell

tasks:

- name: Dump database

- command: mongodump --db {{ mongodb_name }} --port {{ mongodb_port }} -p {{ mongodb_pass }} -u {{ mongodb_user }} --out {{ mongodb_dump }}

Use Dynamic Inventory With Clouds



Microsoft Azure



Galaxy

Ansible Galaxy is your hub for finding, reusing, and sharing the best Ansible content.

Browse Roles

Categories

web
system
packaging
networking
monitoring
development
database:sql
database:nosql
database
clustering
cloud:rax
cloud:gce
cloud:ec2
cloud

sphinx x

Role Name

 Reverse Order

Role

Categories

Description

sphinx

database

Ansible role for Sphinx search engine

4.8 Community Score

Created on: 8/27/14 11:46 AM Modified: 8/27/14 6:27 PM

NA Ansible Score

Author: [devjatkin](#)

sphinxdoc

system
web

Ansible playbook for Sphinxdoc dependencies

NA Community Score

Created on: 1/13/15 12:01 AM Modified: 1/15/15 9:33 AM

NA Ansible Score

Author: [deimosfr](#)

1

Galaxy: requirements.txt

```
ANXS.nodejs,v1.1.0  
jdauphant.nginx,v1.3.3  
geerlingguy.php,1.4.5  
geerlingguy.ruby,2.0.1  
geerlingguy.mysql,1.4.3  
kossi.composer,v1.3.0  
AbdoulNdiaye.Blackfire,v1.0.2
```

```
ansible-galaxy install -r requirements.txt
```

Galaxy: requirements.yml

```
# from github  
- src: https://github.com/bennojoy/nginx  
  version: master  
  name: bennojoy_nginx_role
```

```
ansible-galaxy install -r requirements.yml
```

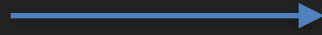
Deployment

- Capistrano (**RUBY**)
 - Capifony →
- Fabric (**PYTHON**)

```
`-- /var/www/app.com
|-- current → /var/www/app.com/releases/20100512131539
|-- releases
|  `-- 20100512131539
|  `-- 20100509150741
|  `-- 20100509145325
`-- shared
    |-- web
    |  `-- uploads
    |-- log
    `-- config
        `-- databases.yml
```


Deployment

- cap deploy:setup
- cap deploy



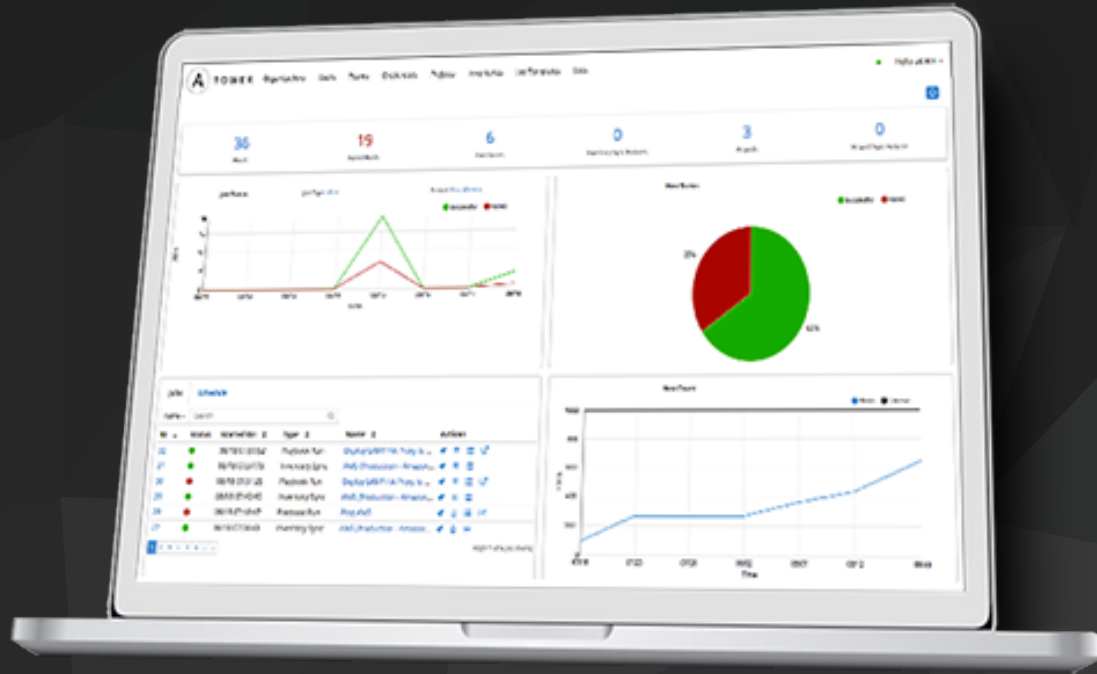
deploy.yml

- Something went wrong???
- cap deploy:rollback



rollback.yml

Tower - freemium, 10 hosts



Give a shot

- If you do anything, do it from Ansible. Don't SSH to a server. (Achieving Continuous Delivery: An Automation Story - PyCon 2015)
- Separate your setup and deploy playbooks
- Use roles from galaxy, or use tools like:
 - phansible.com (PHP)
 - github.com/ansible/ansible-examples

Questions?
Thank you!



www.axiomadev.com