



**pradedantiesiems**

Linus Gricius  
KaunasPHP v.12, 2014-03-12



# Linas Gricius

Senior Developer  
LLC Kupishoes / Lamoda.ru



[www.linkedin.com/in/linasgricius](http://www.linkedin.com/in/linasgricius)

# Kodėl Ansible?

- Užtenka SSH prieigos - Python Ansible įdiegs po prisijungimo jei sistema neturi
- Palaiko autentifikaciją su SSH raktais (+ Kerberos, LDAP ir kt.)
- Nereikia žinoti kitų programavimo kalbų
- YAML formato failai + komandos
- Veikia su Vagrant

# Žodynas

- Playbook - komandų rinkinys
- Inventory - mašinų sąrašas
- ad-hoc command - komanda, kurią norime paleisti, bet nenorime išsaugoti ateičiai

# Diegimas

Pagal jūsų naudojamą OS. Variantai:

- Source - iš GITHUB
- Paketai - iš OS distribucijos
- Brew - OS X

Valdymo mašina turi turėti Python 2.6 kad veikėtų Ansible.

Windows OS nepalaikoma :-)

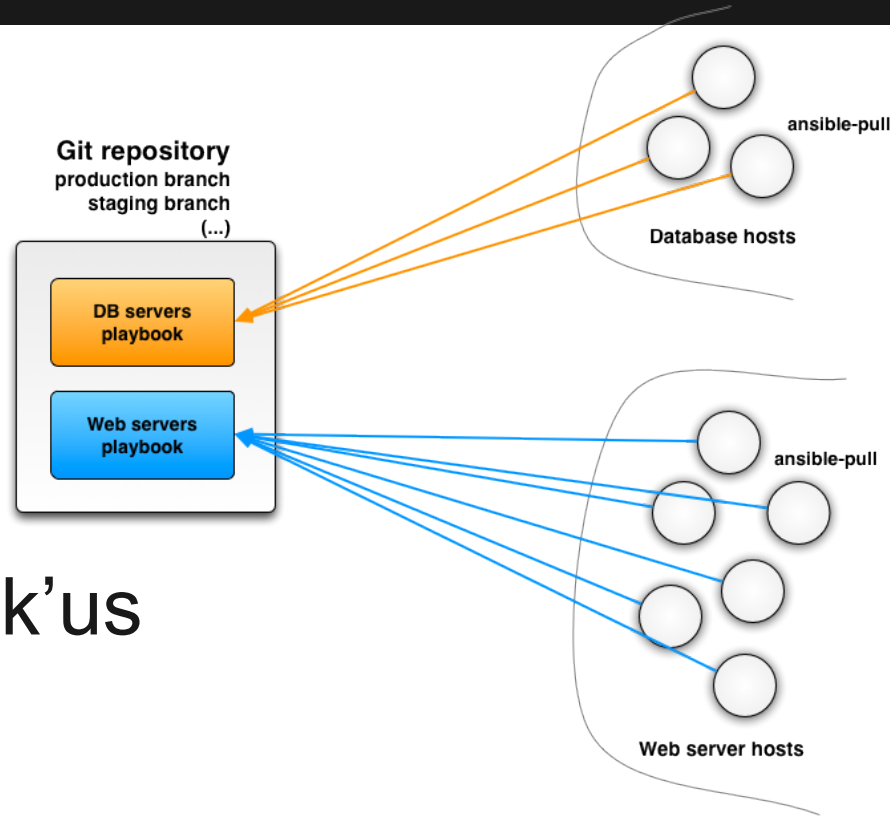
# ad-hoc tasks vs playbooks

Vienkartinėms užduotims galima naudoti ad-hoc komandas. Sudėtingesniems veiksmams - Ansible Playbook'us.

Ad-hoc pvz.: turim serverių grupę [WEB]:  
\$ ansible WEB -a "/usr/local/bin/clean-cache"

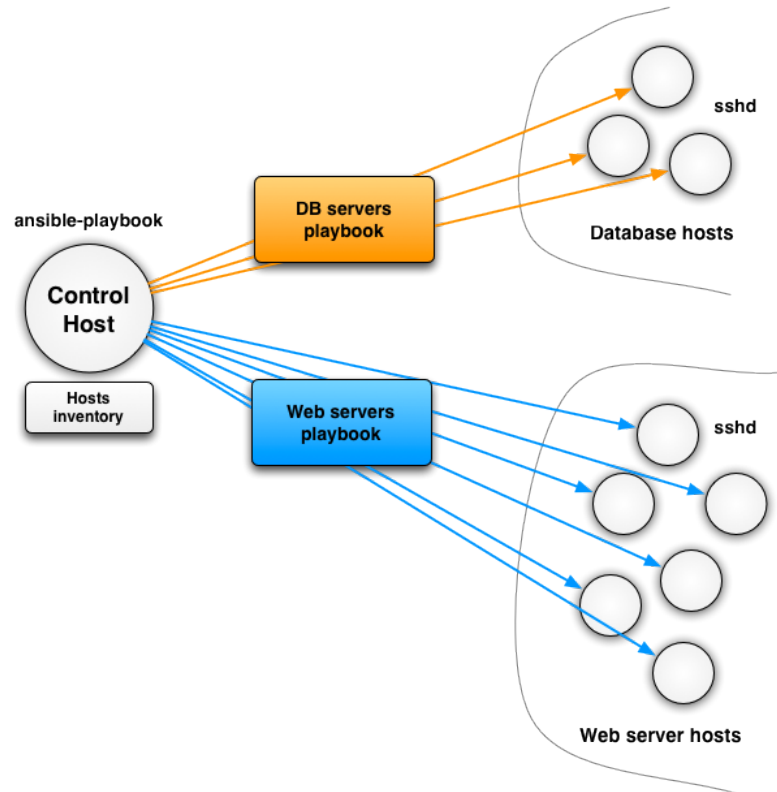
# Ansible - pull

- mašinose turi būti įdiegtas Ansible
- Pačios mašinos turi kreiptis į pvz. Git repozitoriją iš kur ima Ansible Playbook'us



# Ansible - push

- galima viską valdyti iš vienos mašinos
- vykdomas komandas galima sugrupuoti pagal valdomų mašinų tipą





# Pavyzdys

Dirbant su daug projektų, patogiu turėti įrankius, kurie leidžia greitai paruošti darbinę aplinką ir lengvai ją atnaujinti jei atsirado naujų pakeitimų.

Siūlyčiau įrankius:

- Virtualbox
- Vagrant

Ansible

# Įsivaizduokim...

... kad dirbam su projektu, kuris turi:

- Debian
- Nginx
- PHP5-FPM
- MariaDB

ir pvz Zend Framework 2...

# Vagrant (1)

Pradèkim nuo Vagrant :-)

```
$ mkdir zf2demo
```

```
$ cd zf2demo
```

```
$ vagrant init precise32 http://files.vagrantup.com/precise32.box
```

# Vagrant (2)

Vagrant sugeneravo standartinį failą, kurį pasiredaguojam pagal savo poreikius:

- nustatom tinklo tipą ir IP
- nustatom, kad naudosis Ansible

# Vagrant (3)

```
Vagrant.configure(2) do |config|  
  config.vm.box = "precise32"  
  config.vm.box_url = "http://files.vagrantup.com/precise32.box"  
  config.vm.network "private_network", ip: "123.123.123.123"  
  config.vm.provision "ansible" do |ansible|  
    ansible.verbose = 'vv'  
    ansible.limit = 'all'  
    ansible.playbook = "provisioning/playbook.yml"  
  end  
end
```

# Perspējimas (1)

Gali kilti problēmu dēļ Vagrant ir SSH...

Papildom failā ~/.ssh/config :

*Host 127.0.0.1*

*StrictHostKeyChecking no*

*UserKnownHostsFile=/dev/null*

# Perspējimas (2)

<http://files.vagrantup.com/precise32.box> jau yra  
īdiegtas **Python 2.7** palaikymas. Jei jūsu  
mašina neturi Python, tai vienu no Ansible  
užduočių turētu būti Python īdiegimas

# Vagrant (4)

Susikūriau tokią struktūrą:

provisioning	← Ansible failai
roles	← Rolės (common, nginx...)
vars	← Nustatymai
playbook.yml	← Ansible Playbook'as



# Pavyzdys (3)

hosts	# aprašom naudojamą mašiną
group_vars/group1	# "group1" skirti kintamieji
host_vars/hostname1	# "hostname1" mašinos kintamieji
site.yml	# pagrindinis Ansible Playbook'as
webservers.yml	# atskirai grupei skirtas Ansible Playbook'as
roles/common/	# Užduotys grupuojamos pagal "roles"
tasks/main.yml	# pagrindinis rolės užduočių failas
handlers/main.yml	# kaip užduotys, tik suaktyvinamos "notify" pagalba
templates/ntp.conf.j2	# failų šablonai
files/bar.txt	# jei reikės failus perkelti į mašiną
vars/main.yml	# rolės kintamieji

# Ansible (1)

Sugrupavau užduotis į tokias ROLES:

- common
- mariadb
- nginx
- php5-fpm
- zf2

# Ansible (2)

Kaip ir anksčiau minėjau, Ansible Playbook'as naudoja YML failo formatą. Pvz:

“---

- name: Common | define hostname

  - sudo: yes

  - action: template src=hosts.j2

  - dest=/etc/hosts"

# Ansible (3)

Toliau pateiksiu iš kiekvienos rolės komandų pavyzdžių. Ansible palaikomų komandų aprašymą su pavyzdžiais rasite dokumentacijoje arba Google :)

# Ansible (4)

provisioning/roles/common/tasks/main.yml:

- name: Common | install required packages

sudo: yes

action: apt name={{ item }} state=latest

with\_items:

- vim

- curl

# Ansible (5)

provisioning/roles/common/tasks/main.yml:

- name: Common | create web user

sudo: yes

user: name={{ web\_user }} password={{  
web\_password }} groups="sudo" comment="  
Comment" state=present

# Ansible (6)

- name: MariaDB | Check if repositories for mariadb exist

sudo: yes

apt\_repository: repo='deb http://mirrors.supportex.net/mariadb/repo/5.5/debian wheezy main' state=present update\_cache=yes

# Ansible (7)

provisioning/roles/mariadb/tasks/main.yml:

- name: MariaDB | Install mariadb

sudo: yes

apt: pkg=mariadb-server state=latest

notify: restart mysql



# Ansible (8)

roles/mariadb/handlers/main.yml:

---

- name: restart mysql
  - sudo: yes
  - service: name=mysql state=restarted

# Ansible (9)

provisioning/roles/nginx/tasks/main.yml:

- name: enable the default site

- sudo: yes

- file: path={{nginx\_dir}}/sites-enabled/default  
src={{nginx\_dir}}/sites-available/default  
state=link

- notify: restart nginx

# Ansible (10)

provisioning/roles/php5-fpm/tasks/main.yml:

- name: Install composer

- sudo: yes

- shell: curl -sS https://getcomposer.

- org/installer | /usr/bin/php && /bin/mv -f

- /home/vagrant/composer.phar

- /usr/local/bin/composer

- creates=/usr/local/bin/composer

# Pabaigai

Parodžiau tik pavyzdį kaip galima būtų naudojant nemokamus įrankius pasidaryti patogesnę gyvenimą :) Toliau viskas priklauso tik nuo jūsų ir jūsų fantazijos...

Kodą rasite: <https://github.com/lgricius/ansible-Inmp/>

# Nuorodos

- <http://ansible.com/ansible-resources>
- <http://docs.ansible.com/glossary.html>
- [http://docs.ansible.com/modules\\_by\\_category.html](http://docs.ansible.com/modules_by_category.html)
- <https://github.com/ansible/ansible-examples>
- <http://www.michaelrigart.be/en/blog/configuration-management-with-ansible-playbooks-execution.html>
- <http://julien.ponge.org/>
- <http://docs.vagrantup.com/v2/provisioning/ansible.html>
- <https://github.com/francisbesset/ansible-playbooks/>

Ačiū!  
Klausimai?