

● Быть лучше каждый день

Ansible: Тракторный режим

Практика в МТС ИТ по работе с ролями Ansible.

- *Что сподвигло нас*
- *Что такое «Тракторный режим»*
- *Какие сложности возникли при переходе на него*
- *Это идеал?*

НИФАНИН Константин
МТС ИТ



ksnifani@mts.ru

МТС

Ansible: Тракторный режим
Что сподвигло нас?



???



???



Новенький



???

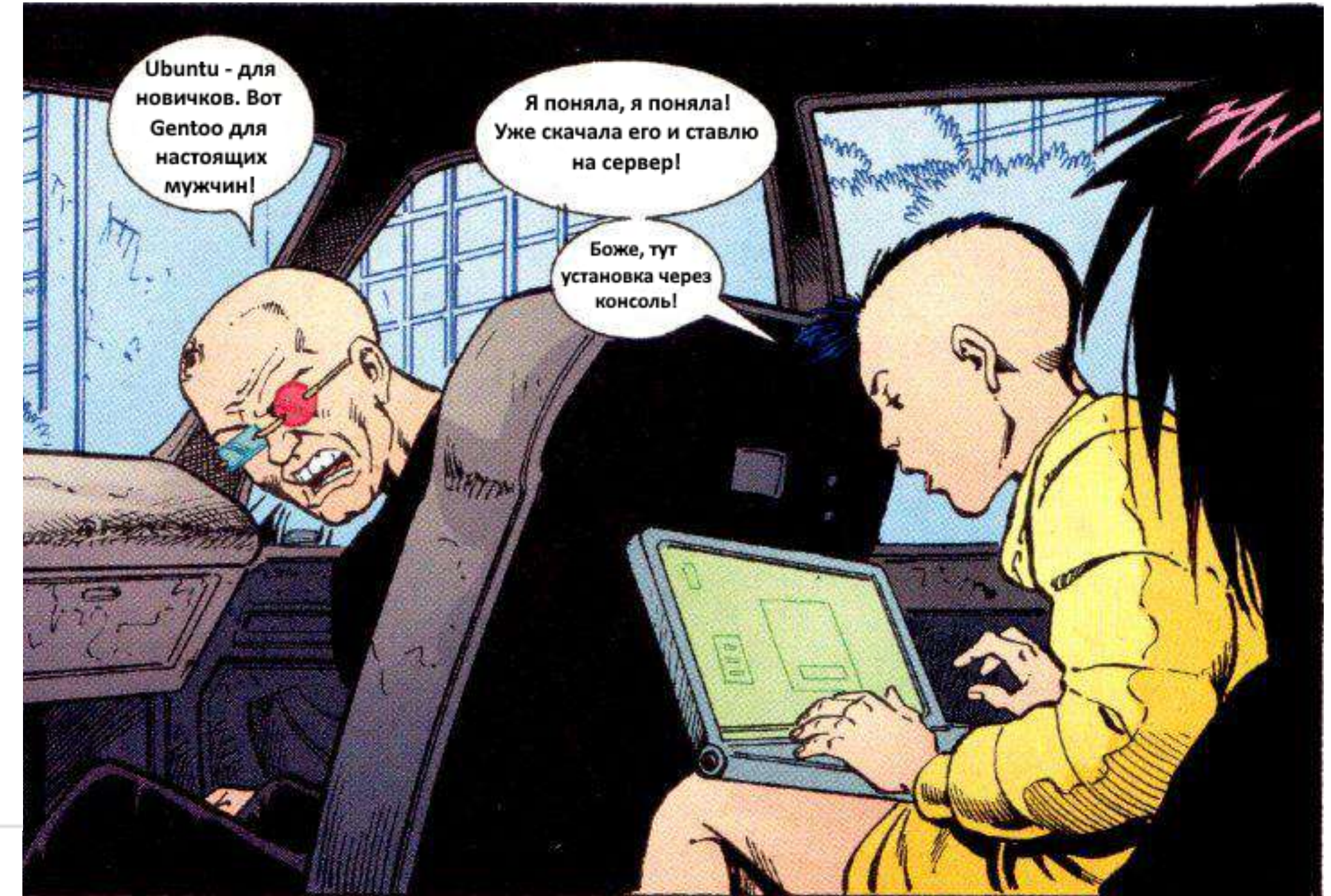
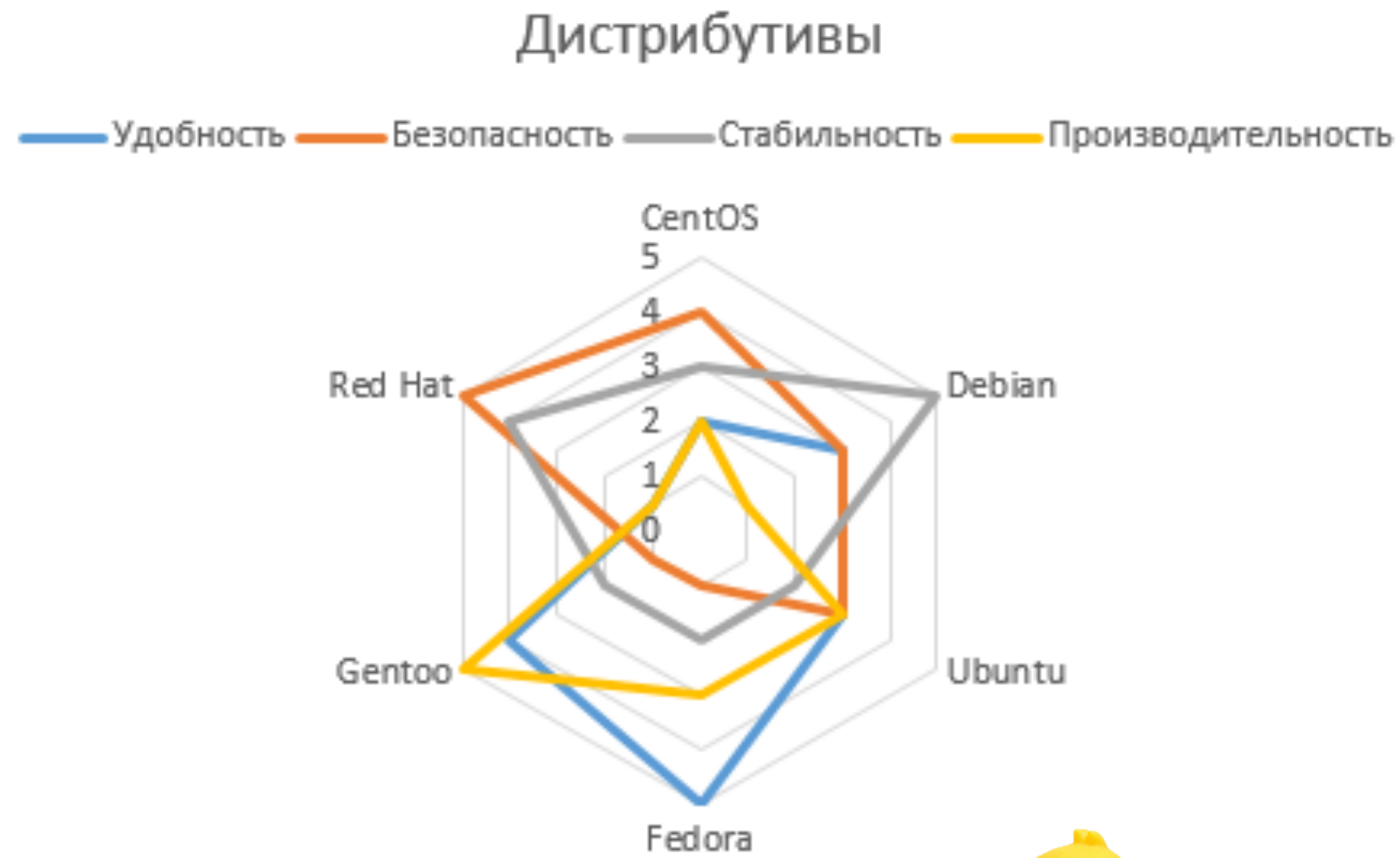


???



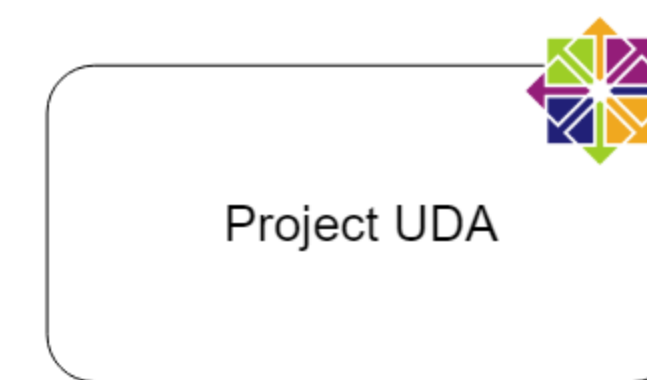
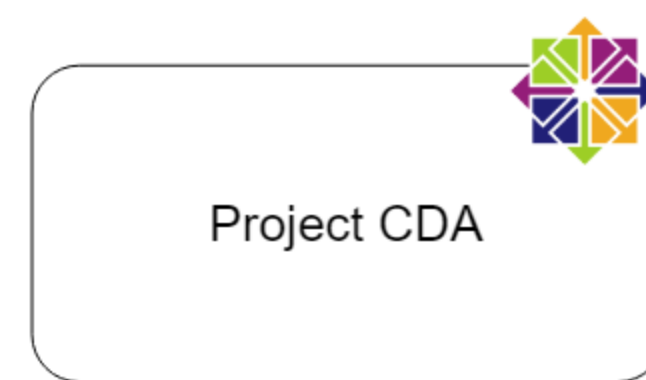
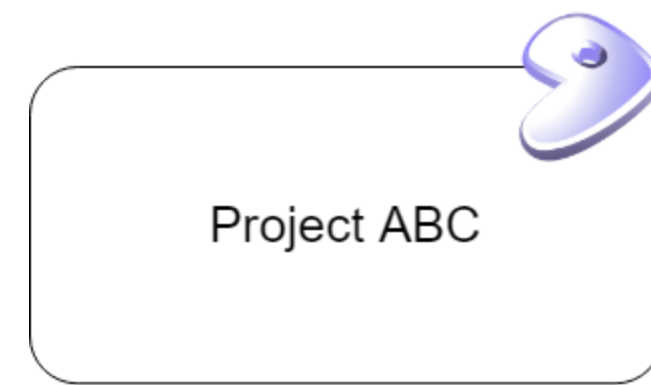
Ansible: Тракторный режим

Что сподвигло нас?



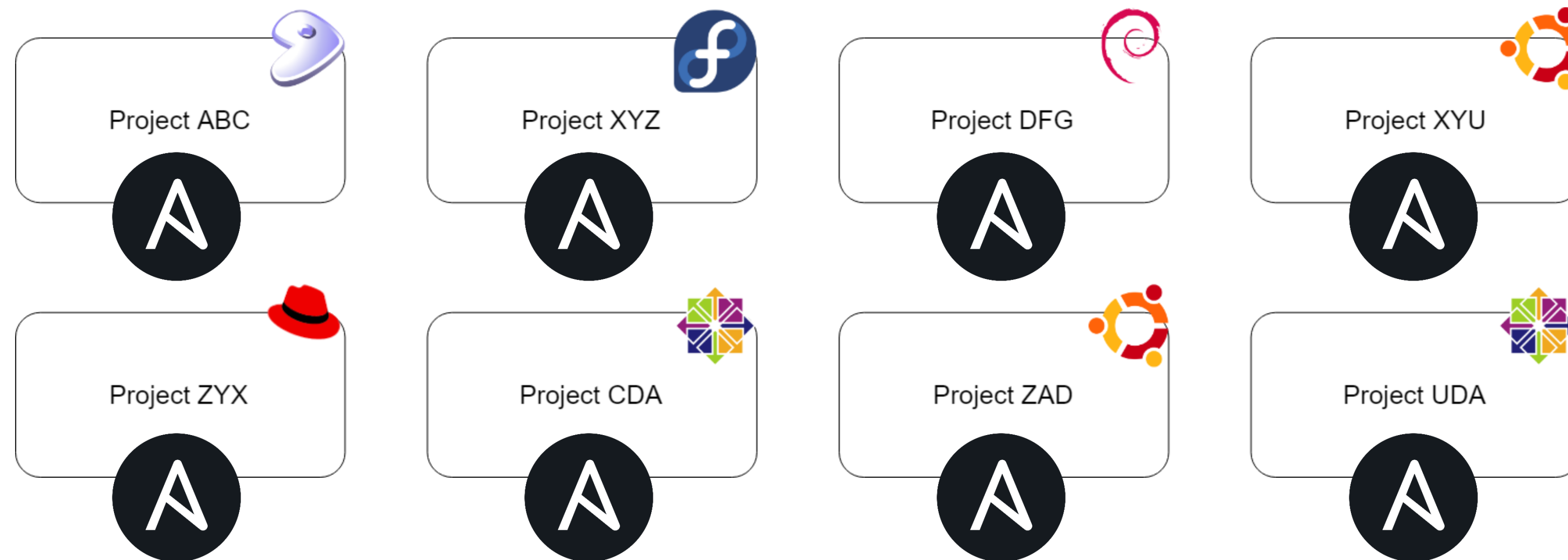
Ansible: Тракторный режим

Что сподвигло нас?



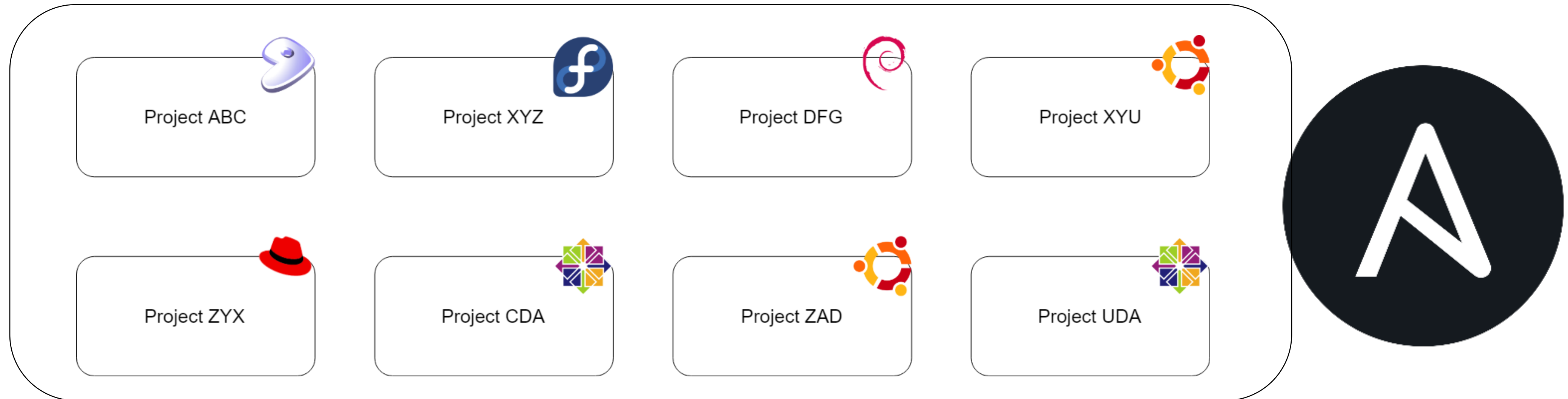
Ansible: Тракторный режим

Что сподвигло нас?



Ansible: Тракторный режим

Что сподвигло нас?



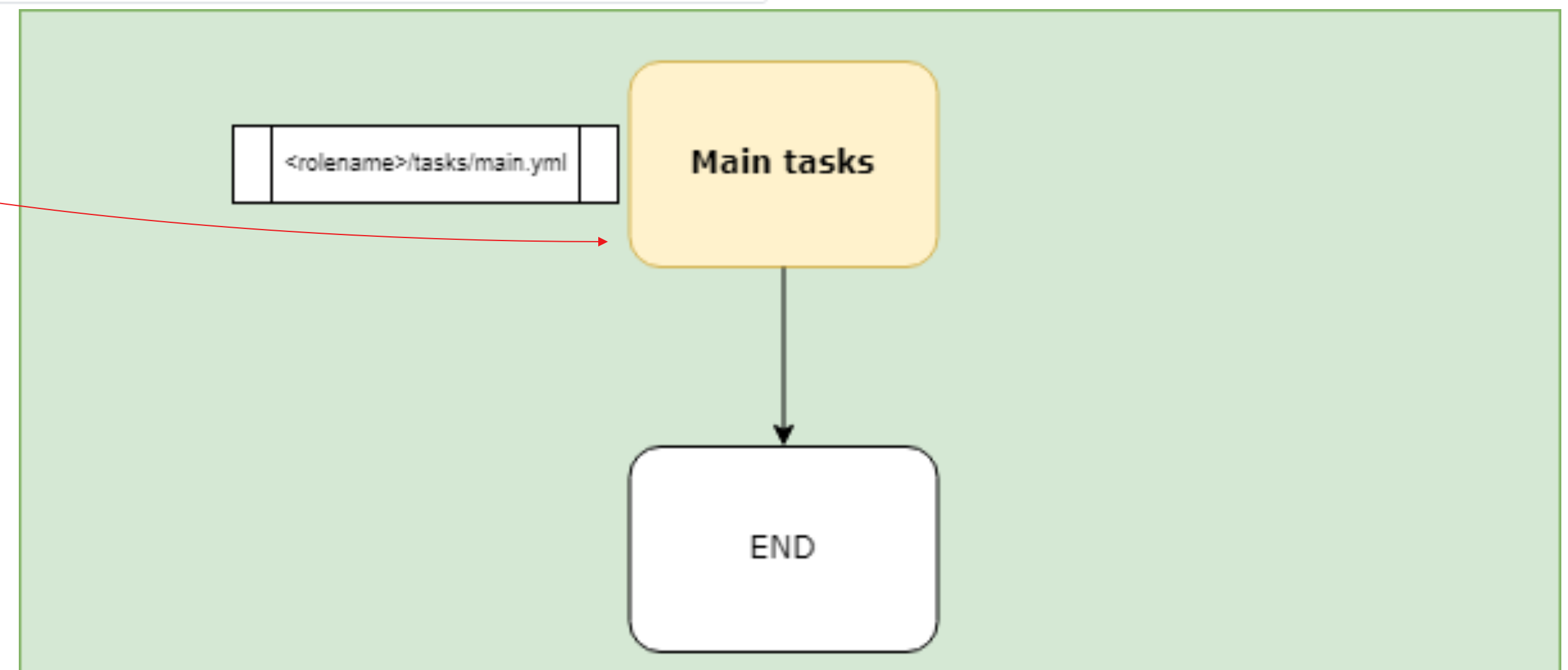
Ansible: Тракторный режим

Что сподвигло нас?

Плохие примеры:

<https://github.com/nickjj/ansible-docker>

The screenshot shows the GitHub interface for the repository 'nickjj/ansible-docker'. It includes statistics for Watch (20), Star (358), and Fork (121). Below the repository name, there are navigation tabs for Code, Issues (0), Pull requests (1), Projects (0), Security, and Insights. The current view is the 'tasks' directory on the 'master' branch. A commit by 'nickjj' titled 'Fix Ansible 2.8.x warnings' is shown, with the latest commit hash '81a0caF' on 'Jul 20'. A file named 'main.yml' is listed with the commit message 'Fix Ansible 2.8.x warnings' and a commit date of 'last month'. A red arrow points from the 'main.yml' file name to the diagram below.



Ansible: Тракторный режим

Что сподвигло нас?

Плохие примеры:

<https://github.com/nickjj/ansible-docker>

```
- name: Disable pinned Docker version
file:
  dest: "/etc/apt/preferences.d/docker.pref"
  state: "absent"
when: not docker__version | d()

- name: Enable pinned Docker version
template:
  src: "etc/apt/preferences.d/docker.pref.j2"
  dest: "/etc/apt/preferences.d/docker.pref"
  owner: "root"
  group: "root"
  mode: "0644"
when: docker__version | d()

- name: Install Docker's dependencies
apt:
  name: "{{ docker__package_dependencies + docker__pip_dependencies }}"

- name: Add Docker's public GPG key to the APT keyring
apt_key:
  id: "{{ docker__apt_key_id }}"
  url: "{{ docker__apt_key_url }}"

- name: Configure Docker's upstream APT repository
apt_repository:
  repo: "{{ docker__apt_repository }}"
  update_cache: true

- name: Install Docker
apt:
  name: "docker-{{ docker__edition }}"
  state: "{{ docker__state }}"

- name: Install Virtualenv
pip:
  name: "virtualenv"

- name: Configure Docker daemon options (json)
template:
  src: "etc/docker/daemon.json.j2"
  dest: "/etc/docker/daemon.json"
  owner: "root"
  group: "root"
  mode: "0644"
when: docker__default_daemon_json | d() or docker__daemon_json | d()
notify: ["Restart Docker"]

- name: Configure Docker daemon options (flags)
template:
  src: "etc/systemd/system/docker.service.d/options.conf.j2"
  dest: "/etc/systemd/system/docker.service.d/options.conf"
  owner: "root"
  group: "root"
  mode: "0644"
register: docker__register_daemon_flags
when: docker__daemon_flags | d()
notify: ["Restart Docker"]

- name: Configure Docker daemon environment variables
template:
  src: "etc/systemd/system/docker.service.d/environment.conf.j2"
  dest: "/etc/systemd/system/docker.service.d/environment.conf"
  owner: "root"
  group: "root"
  mode: "0644"
register: docker__register_daemon_environment
when: docker__daemon_environment | d()
notify: ["Restart Docker"]

- name: Configure custom systemd unit file override
template:
  src: "etc/systemd/system/docker.service.d/custom.conf.j2"
  dest: "/etc/systemd/system/docker.service.d/custom.conf"
  owner: "root"
```

Configuring the Docker daemon environment variables

```
docker__daemon_environment: []

# For example, here's how to set a couple of proxy environment variables.
docker__daemon_environment:
  - "HTTP_PROXY=http://proxy.example.com:80"
  - "HTTPS_PROXY=https://proxy.example.com:443"
```

```
register: docker__register_daemon_flags
when: docker__daemon_flags | d()
notify: ["Restart Docker"]
```

```
- name: Configure Docker daemon environment variables
template:
  src: "etc/systemd/system/docker.service.d/environment.conf.j2"
  dest: "/etc/systemd/system/docker.service.d/environment.conf"
  owner: "root"
  group: "root"
  mode: "0644"
register: docker__register_daemon_environment
when: docker__daemon_environment | d()
notify: ["Restart Docker"]
```

```
- name: Configure custom systemd unit file override
template:
  src: "etc/systemd/system/docker.service.d/custom.conf.j2"
  dest: "/etc/systemd/system/docker.service.d/custom.conf"
  owner: "root"
```


Ansible: Тракторный режим

Что сподвигло нас?

Плохие примеры:

<https://everythingshouldbevirtual.com/ansible-playbook-lvm/>

Whenисты...

```
---  
- name: installing lvm2  
  apt: name=lvm2 state=present  
  when: config_lvm and ansible_os_family == "Debian"  
  
- name: installing lvm2  
  yum: name=system-storage-manager state=present  
  when: config_lvm and ansible_os_family == "RedHat"  
  
- name: installing scsitools  
  apt: name=scsitools state=present  
  when: config_lvm and ansible_os_family == "Debian"  
  
- name: installing sg3_utils  
  yum: name=sg3_utils state=present  
  when: config_lvm and ansible_os_family == "RedHat"  
  
- name: rescanning for new disks  
  command: /sbin/rescan-scsi-bus  
  when: config_lvm and ansible_os_family == "Debian"  
  
- name: rescanning for new disks  
  command: /usr/bin/rescan-scsi-bus.sh  
  when: config_lvm and ansible_os_family == "RedHat"
```

Ansible: Тракторный режим

Что подвигнуло нас?

Плохие примеры:

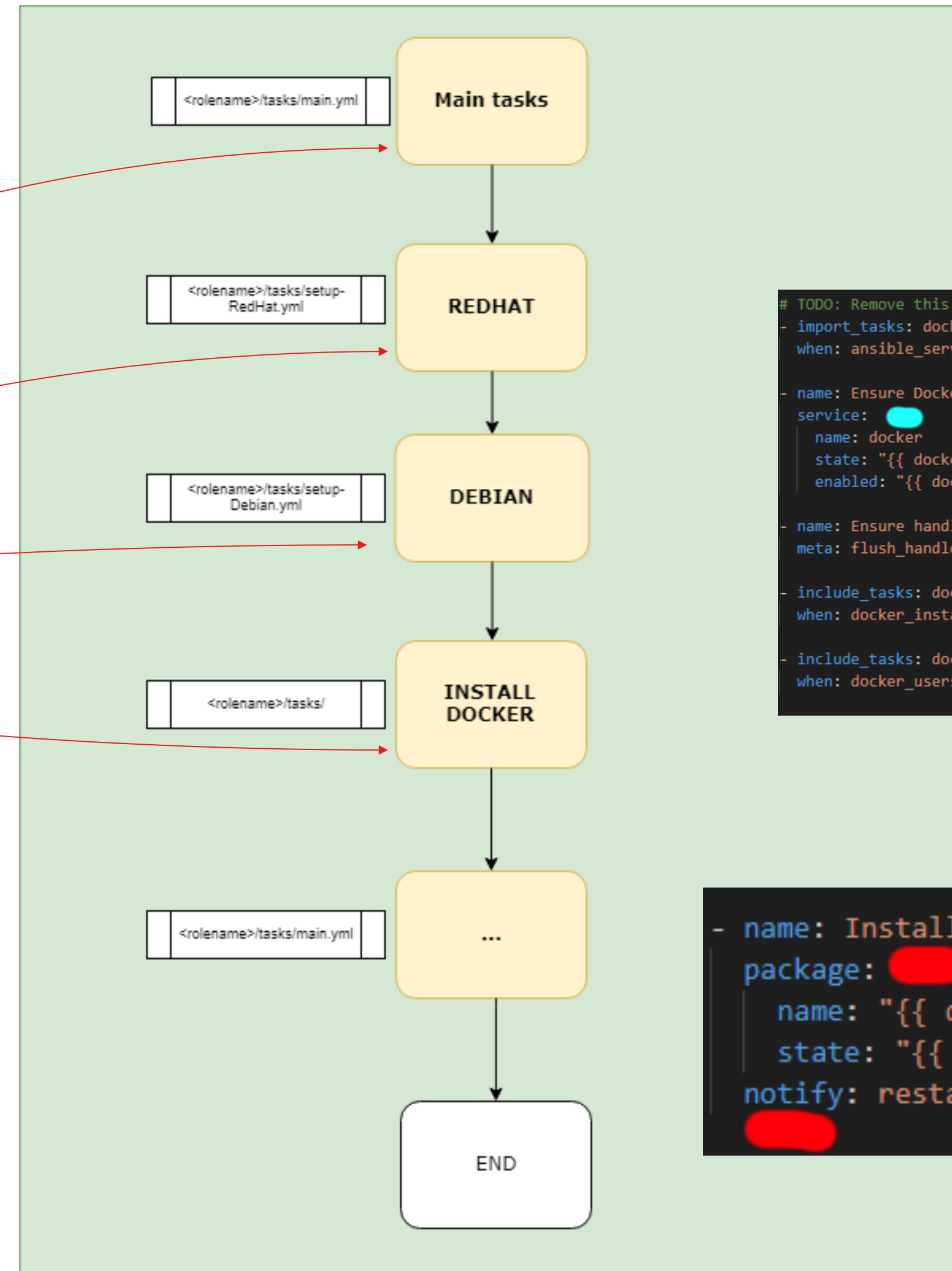
<https://github.com/geerlingguy/ansible-role-docker>

geerlingguy / ansible-role-docker

Code Issues 35 Pull requests 37 Projects 0 Security Insights

Branch: master ansible-role-docker / tasks /

| File | Description | Time |
|----------------------|---|---------------|
| docker-1809-shim.yml | Fixes #97, Mentions #98, Still makes me sad: Fix Docker 18.09.0 star... | 10 months ago |
| docker-compose.yml | Issue #2: Allow customization of the Docker Compose path. | 2 years ago |
| docker-users.yml | Switch tests to use Molecule. | 11 months ago |
| main.yml | PR #146: Check list length instead of casting to bool. | 3 days ago |
| setup-Debian.yml | Fixes ansible-lint failure on shell task without pipefail. | 5 months ago |
| setup-RedHat.yml | Fix broken Fedora 27 test, clean up uninstall task. | 10 months ago |



```

- include_tasks: setup-RedHat.yml
  when: ansible_os_family == 'RedHat'
- include_tasks: setup-Debian.yml
  when: ansible_os_family == 'Debian'
- name: Install Docker.
  package:
    name: "{{ docker_package }}"
    state: "{{ docker_package_state }}"
  notify: restart docker
  
```

INCLUDES!
NO INCLUDE???

```

# TODO: Remove this shim once 18.09.1 or later is released.
- import_tasks: docker-1809-shim.yml
  when: ansible_service_mgr == 'systemd'
- name: Ensure Docker is started and enabled at boot.
  service:
    name: docker
    state: "{{ docker_service_state }}"
    enabled: "{{ docker_service_enabled }}"
- name: Ensure handlers are notified now to avoid firewall conflicts.
  meta: flush_handlers
- include_tasks: docker-compose.yml
  when: docker_install_compose | bool
- include_tasks: docker-users.yml
  when: docker_users | length > 0
  
```

INCLUDES
NO INCLUDES
INCLUDES...

```

- name: Install Docker.
  package:
    name: "{{ docker_package }}"
    state: "{{ docker_package_state }}"
  notify: restart docker
  
```

NO PROXY
NO VERSIONS
PACKAGE

Ansible: Тракторный режим

Что сподвигло нас?

Плохие примеры:

<https://github.com/openstack/openstack-ansible>

Ansible playbooks for deploying OpenStack. <https://opendev.org>

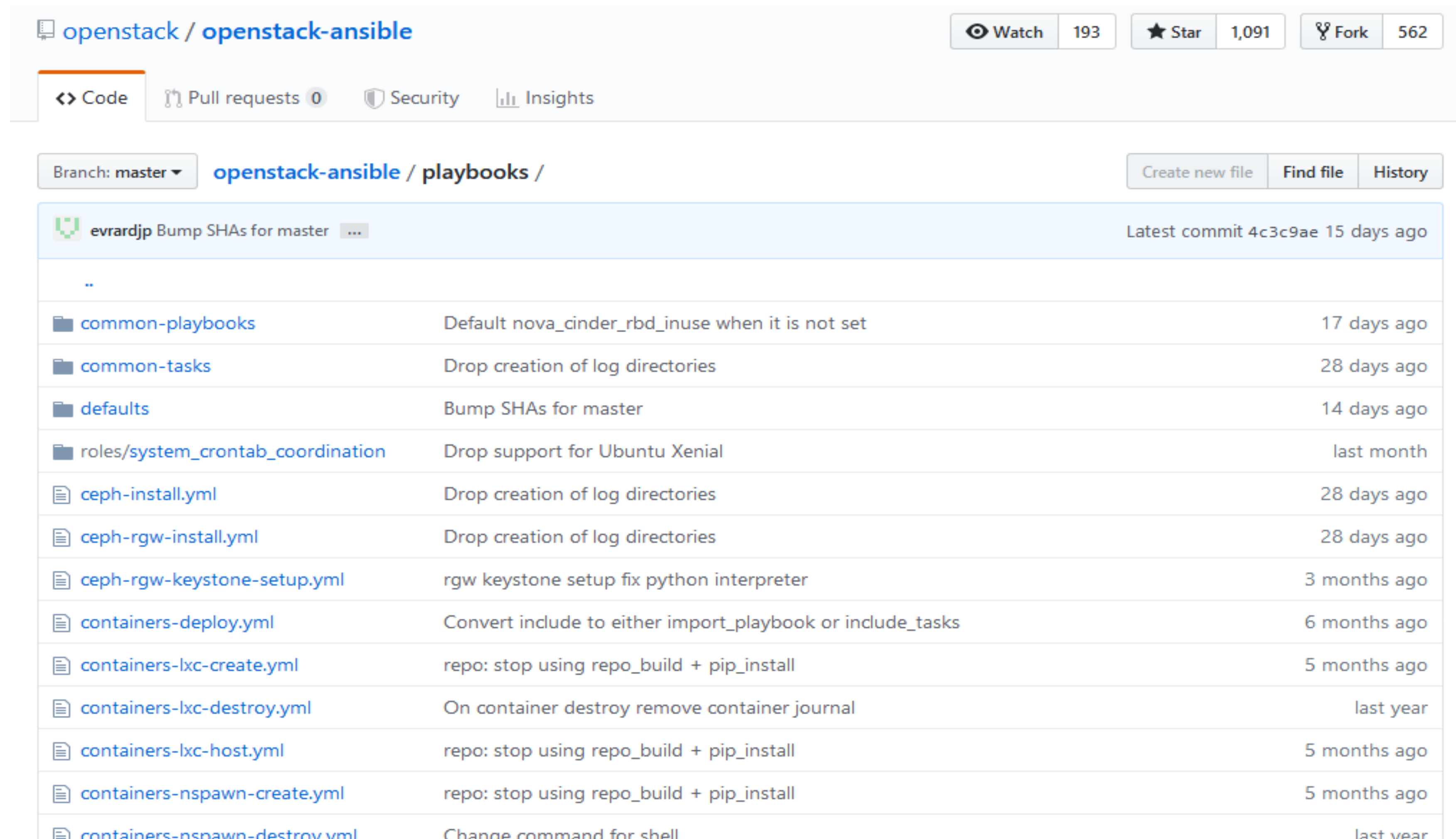
| | | | | |
|--|--|--------------|-------------------|---------------|
| 6,369 commits | 6 branches | 290 releases | 196 contributors | Apache-2.0 |
| Branch: master | New pull request | Find File | Clone or download | |
| logan2211 and noonedeapunk Generate static ARA report when necessary ... Latest commit e97e321 10 days ago | | | | |
| deploy-guide/source | Update links | | | 22 days ago |
| doc | Imported Translations from Zanata | | | 11 days ago |
| etc | lxd: remove support | | | 19 days ago |
| inventory | Temporary override of ceph openstack_keys | | | 5 days ago |
| osa_toolkit | Merge "Fix scripts/inventory-manage.py help invocation" | | | 13 days ago |
| playbooks | Bump SHAs for master | | | 14 days ago |
| releasenotes | Imported Translations from Zanata | | | 12 days ago |
| scripts | Generate static ARA report when necessary | | | 4 days ago |
| tests | powervm: drop un-used vars | | | 19 days ago |
| zuul.d | Generate static ARA report when necessary | | | 4 days ago |
| .gitignore | Remove non-working dynamic build of AIO scenario table | | | 11 months ago |
| .gitreview | OpenDev Migration Patch | | | 4 months ago |
| LICENSE.txt | Correct path to callback plugins in gate script | | | 4 years ago |
| README.rst | Replace git.openstack.org with opendev.org | | | 4 months ago |
| Vagrantfile | Remove xenial from Vagrantfile | | | 9 months ago |
| ansible-role-requirements.yml | Update Ansible to 2.8.2, Ceph-Ansible to 4.0, Ceph to Nautilus | | | last month |
| ansible-role-requirements.yml.example | Convert existing roles into galaxy roles | | | 5 years ago |
| bindep.txt | Replace git.openstack.org with opendev.org | | | 4 months ago |
| global-requirement-pins.txt | Bump requirement for master | | | 4 months ago |
| requirements.txt | Update ansible to latest stable 2.6.x | | | 10 months ago |

Ansible: Тракторный режим

Что сподвигло нас?

Плохие примеры:

<https://github.com/openstack/openstack-ansible>



The screenshot shows the GitHub repository page for `openstack / openstack-ansible`. The repository has 193 watchers, 1,091 stars, and 562 forks. The current view is the `playbooks` directory on the `master` branch. A commit by `evrardjp` titled "Bump SHAs for master" is shown at the top, with the latest commit hash `4c3c9ae` and a date of 15 days ago.

| File/Folder | Description | Last Commit |
|--|--|--------------|
| .. | | |
| <code>common-playbooks</code> | Default nova_cinder_rbd_inuse when it is not set | 17 days ago |
| <code>common-tasks</code> | Drop creation of log directories | 28 days ago |
| <code>defaults</code> | Bump SHAs for master | 14 days ago |
| <code>roles/system_crontab_coordination</code> | Drop support for Ubuntu Xenial | last month |
| <code>ceph-install.yml</code> | Drop creation of log directories | 28 days ago |
| <code>ceph-rgw-install.yml</code> | Drop creation of log directories | 28 days ago |
| <code>ceph-rgw-keystone-setup.yml</code> | rgw keystone setup fix python interpreter | 3 months ago |
| <code>containers-deploy.yml</code> | Convert include to either import_playbook or include_tasks | 6 months ago |
| <code>containers-lxc-create.yml</code> | repo: stop using repo_build + pip_install | 5 months ago |
| <code>containers-lxc-destroy.yml</code> | On container destroy remove container journal | last year |
| <code>containers-lxc-host.yml</code> | repo: stop using repo_build + pip_install | 5 months ago |
| <code>containers-nspawn-create.yml</code> | repo: stop using repo_build + pip_install | 5 months ago |
| <code>containers-nspawn-destroy.yml</code> | Change command for shell | last year |

Ansible: Тракторный режим

Что сподвигло нас?

Плохие примеры:

<https://github.com/openstack/openstack-ansible>

The image shows a GitHub repository for 'openstack-ansible' with the following structure:

- Branch: master
- Repository: openstack-ansible / playbooks / setup-openstack.yml
- Contributors: 17 contributors
- File size: 48 lines (45 sloc) | 1.85 KB

The main content displays a list of imported playbooks in 'setup-openstack.yml':1 ---
2 # Copyright 2014, Rackspace US, Inc.
3 #
4 # Licensed under the Apache License, Version 2.0 (the "License");
5 # you may not use this file except in compliance with the License.
6 # You may obtain a copy of the License at
7 #
8 # http://www.apache.org/licenses/LICENSE-2.0
9 #
10 # Unless required by applicable law or agreed to in writing, software
11 # distributed under the License is distributed on an "AS IS" BASIS,
12 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 # See the License for the specific language governing permissions and
14 # limitations under the License.
15
16 - import_playbook: os-keystone-install.yml
17 - import_playbook: os-barbican-install.yml
18 - import_playbook: os-placement-install.yml
19 - import_playbook: os-glance-install.yml
20 - import_playbook: os-cinder-install.yml
21 - import_playbook: os-nova-install.yml
22 - import_playbook: os-neutron-install.yml
23 - import_playbook: os-heat-install.yml
24 - import_playbook: os-horizon-install.yml
25 - import_playbook: os-designate-install.yml
26 - import_playbook: os-gnocchi-install.yml
27 - import_playbook: os-swift-install.yml
28 - import_playbook: os-ceilometer-install.yml
29 - import_playbook: os-aodh-install.yml
30 - import_playbook: os-panko-install.yml
31 - import_playbook: os-ironic-install.yml
32 - import_playbook: os-magnum-install.yml
33 - import_playbook: os-trove-install.yml

Two detailed views of playbooks are shown with red arrows indicating dependencies:

- os-keystone-install.yml** (lines 43-82):

```
43 - keystone
44 pre_tasks:
45
46 # In order to ensure that any container, software or
47 # config file changes which causes a container/service
48 # restart do not cause an unexpected outage, we drain
49 # the load balancer back end for this container.
50 - include_tasks: common-tasks/haproxy-endpoint-manage.yml
51 vars:
52   haproxy_backend: "keystone_service-back"
53   haproxy_state: disabled
54 when: "groups['keystone_all'] | length > 1"
55
56 - name: Configure container
57 include_tasks: "common-tasks/os-{{ container_tech | default('lxc') }}-container-setup.yml"
58 vars:
59   extra_container_config_no_restart:
60     - "lxc.start.order=19"
61 when: not is_metal
62
63 - include_tasks: common-tasks/unbound-clients.yml
64 when:
65   - hostvars['localhost']['resolvconf_enabled'] | bool
66
67 roles:
68   - role: "os_keystone"
69   - role: "system_crontab_coordination"
70 tags:
71   - crontab
72
73 post_tasks:
74 # Now that container changes are done, we can set
75 # the load balancer back end for this container
76 # to available again.
77 - include_tasks: common-tasks/haproxy-endpoint-manage.yml
78 vars:
79   haproxy_backend: "keystone_service-back"
80   haproxy_state: enabled
81 when: "groups['keystone_all'] | length > 1"
82
```
- common-tasks/unbound-clients.yml** (lines 30-65):

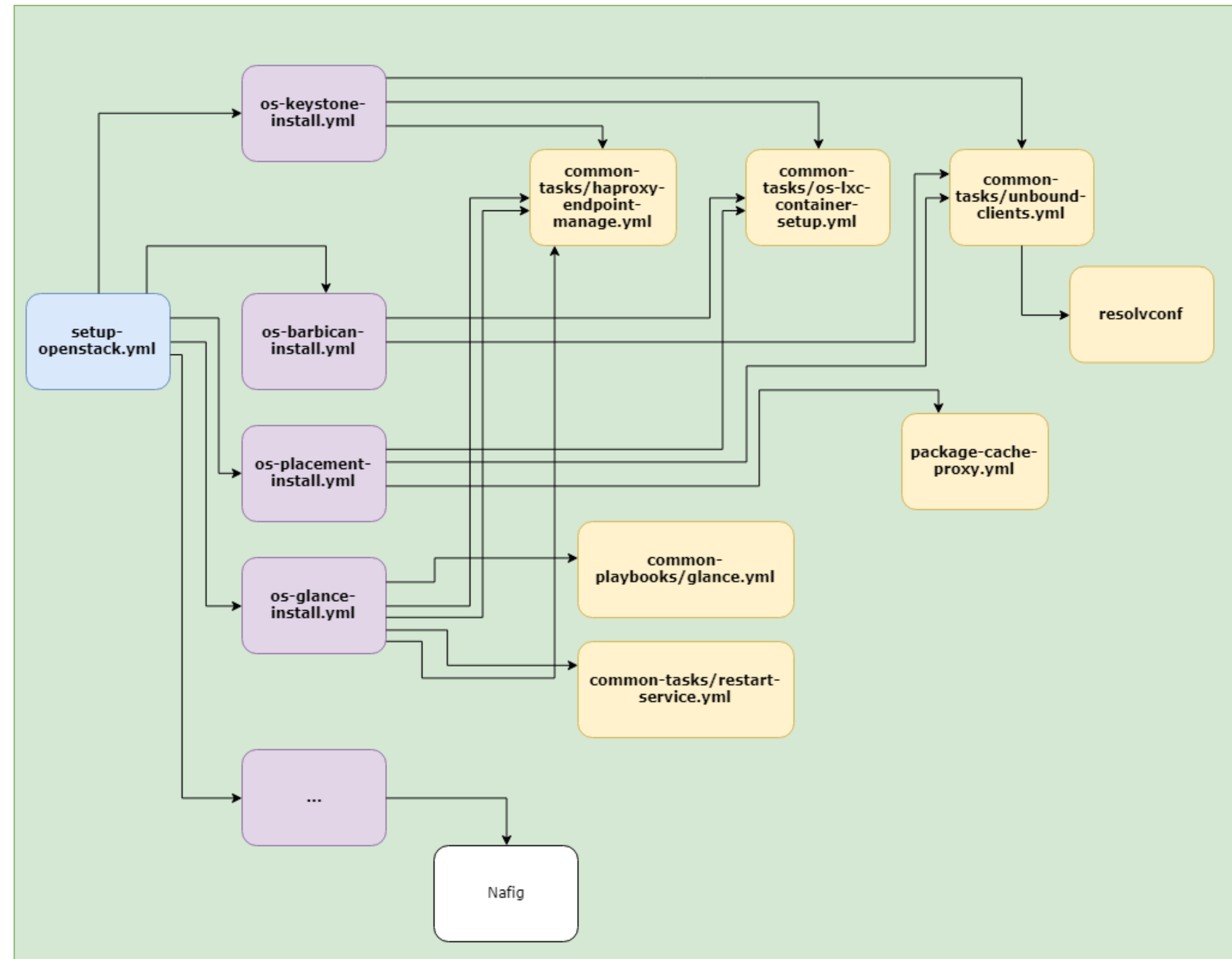
```
30 {% set _ = _var.append({
31   'host': hostvars[host]['ansible_host'],
32   'priority': _prio
33 })
34 %}
35 {% endfor %}
36 {{ _var |
37   sort(reverse=true, attribute='priority') |
38   map(attribute='host') |
39   list
40 }}
41 when:
42   - physical_host is defined
43   - physical_host in hostvars['localhost']['unbound_physical_hosts']
44
45 - name: Set resolver IP list fact
46 set_fact:
47   resolvconf_resolver_ips: "{{ resolver_list }}"
48   resolvconf_options:
49     - 'timeout:1'
50 when:
51   - resolver_list is defined
52   - resolver_list | length > 0
53
54 #rotate is only used when no physical_host affinity is defined
55 - name: Set resolver rotate when physical_host is not an unbound host
56 set_fact:
57   resolvconf_options:
58     - 'timeout:1'
59     - 'rotate'
60 when:
61   - resolver_list is not defined
62
63 - name: Include the resolvconf role
64 include_role:
65   name: resolvconf
```


Ansible: Тракторный режим

Что сподвигло нас?

Плохие примеры:

<https://github.com/openstack/openstack-ansible>



Ansible: Тракторный режим

Что такое «Тракторный режим»

Что лучше?



Ansible: Тракторный режим

Что такое «Тракторный режим»

Основные правила

1 Один сервис – одна роль

2 Роль должна уметь всё!

3 Всё в переменные.

4 Обязательное использование `default` и запрещение использование `vars`

5 Преобладание `include_tasks` над `when` и `result`

6 Обязательное заполнение README

7 Каждый пишет по мере своих знаний, но не ломай чужое

8 Роль должна запускаться как под `ansible-playbook`, так и через AWX

Ansible: Тракторный режим

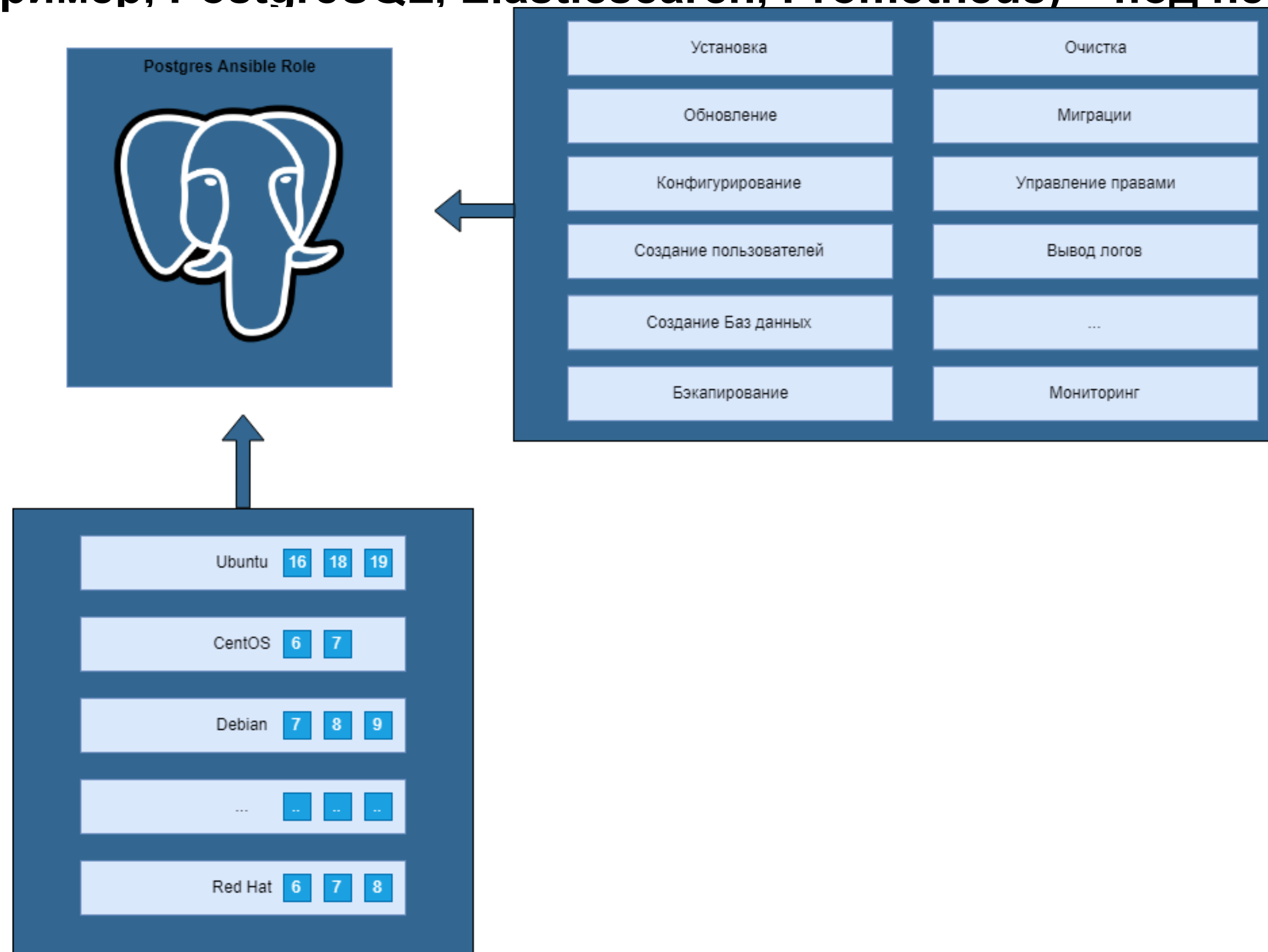
Что такое «Тракторный режим»
Одна роль – один сервис

- Если есть сервис (например, PostgreSQL, Elasticsearch, Prometheus) – под неё пишется отдельная роль.
- Linux = Сервис; Платформа = Сервис; Уникальная библиотека или приложение != Сервис
- Между ролями описывается дерево зависимостей.

Ansible: Тракторный режим

Что такое «Тракторный режим»
Одна роль – один сервис

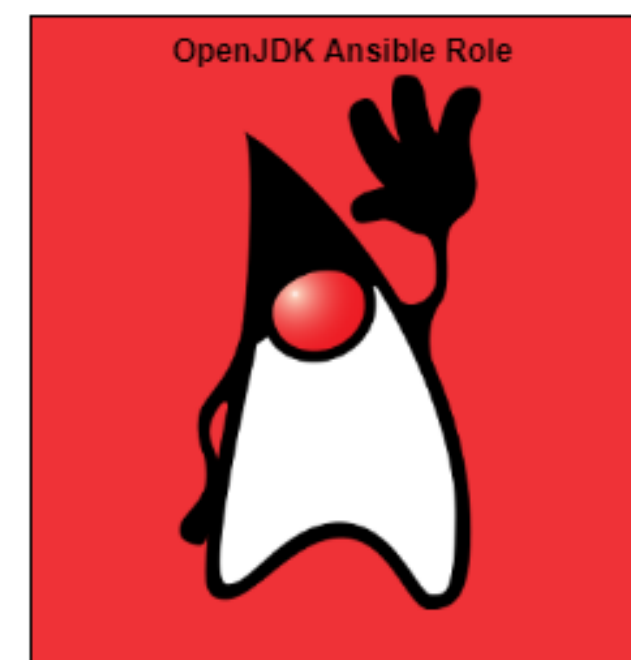
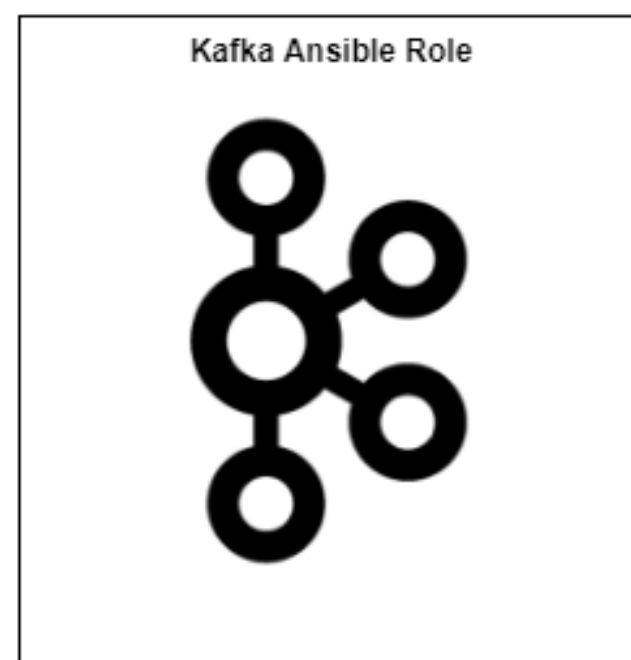
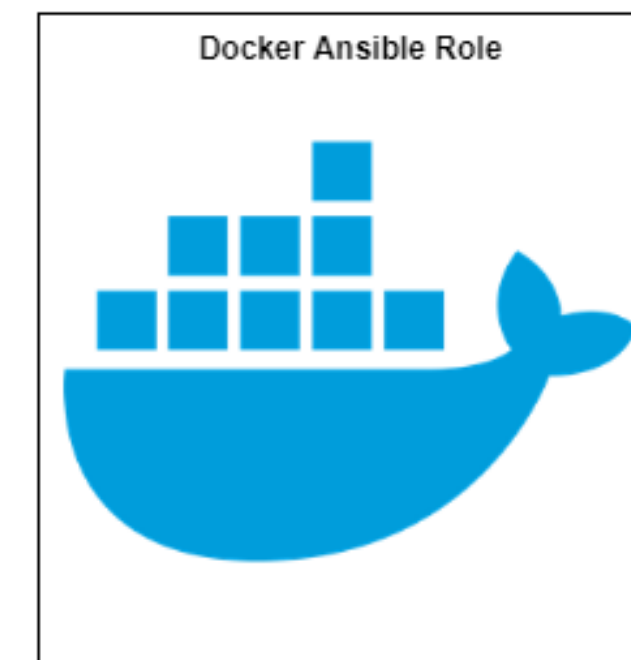
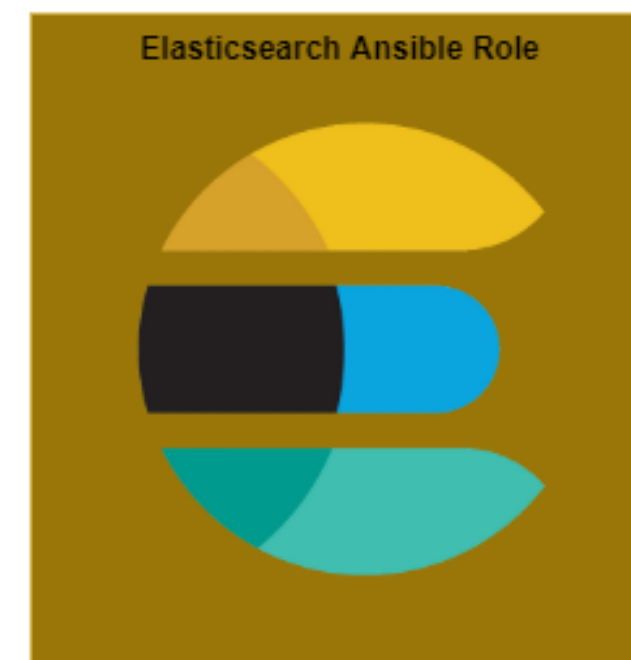
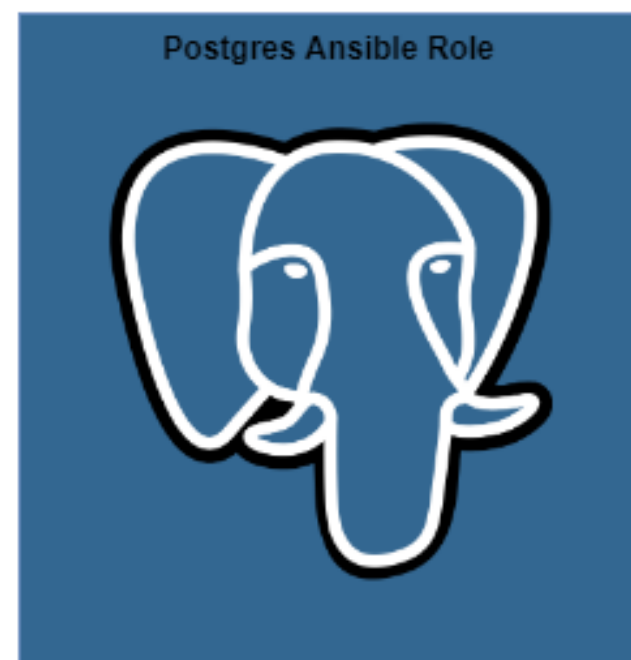
Если есть сервис (например, PostgreSQL, Elasticsearch, Prometheus) – под неё пишется отдельная роль.



Ansible: Тракторный режим

Что такое «Тракторный режим»
Одна роль – один сервис

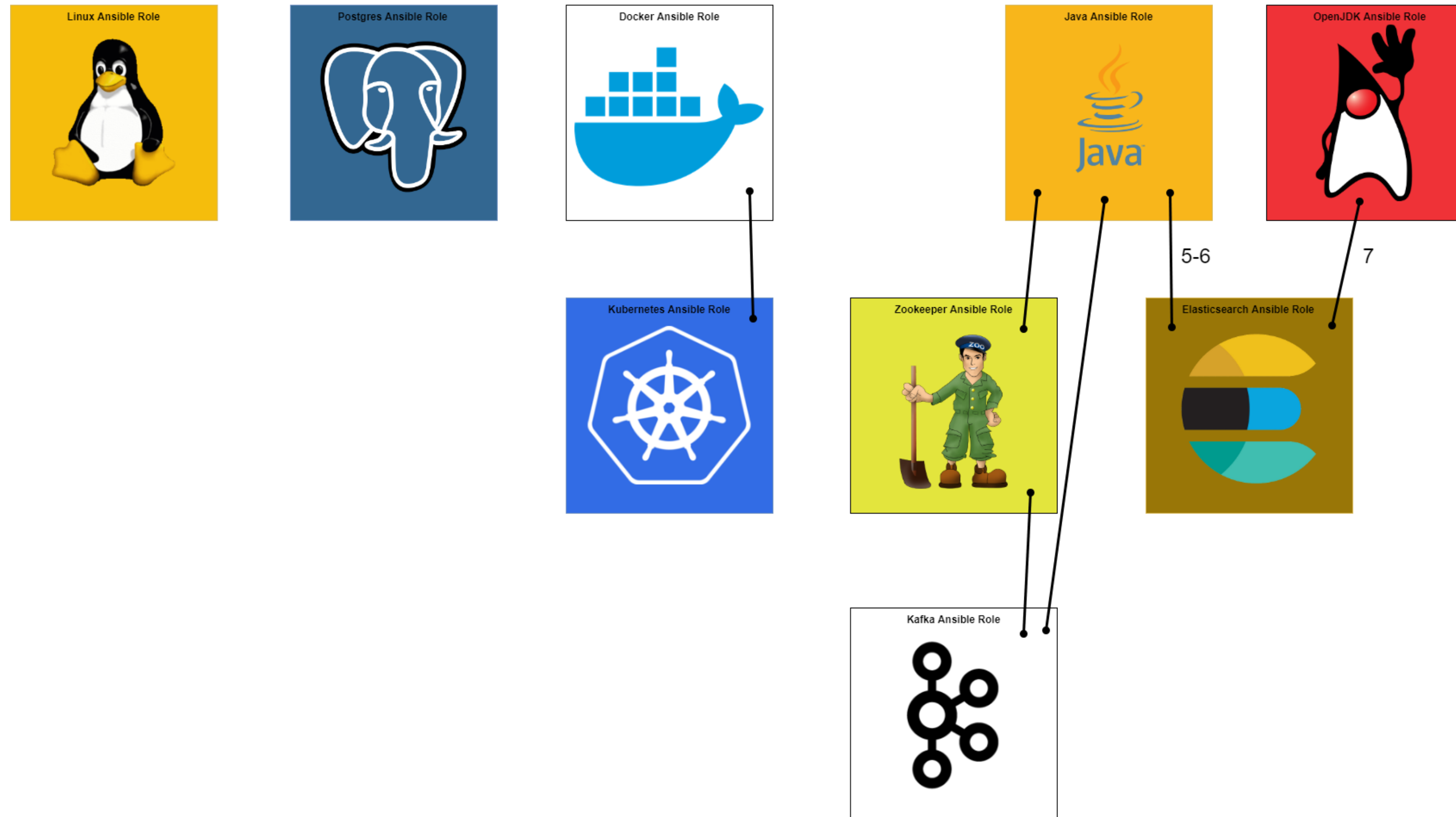
Linux = Сервис; Платформа = Сервис; Уникальная библиотека или приложение != Сервис



Ansible: Тракторный режим

Что такое «Тракторный режим»
Одна роль – один сервис

Между ролями описывается дерево зависимостей.



Ansible: Тракторный режим

Что такое «Тракторный режим»
Одна роль – один сервис

Между ролями описывается дерево зависимостей.



Dependencies

For Elasticsearch 7:

- `openjdk`

For Elasticsearch 5:

- `oracle-java`



Dependencies

- `elasticsearch`

For Logstash 7:

- `openjdk`

For Logstash 5:

- `oracle-java`



Dependencies

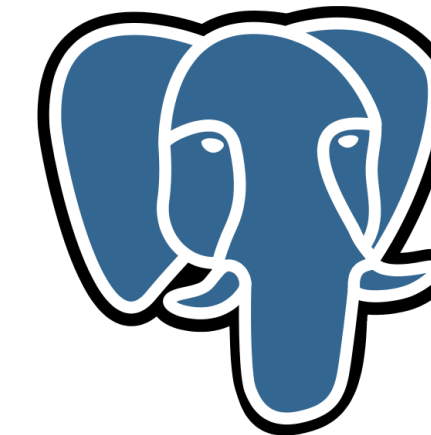
- `elasticsearch`
- `nginx`

For Kibana 7:

- `openjdk`

For Kibana 5:

- `oracle-java`



Dependencies

- `none`

Ansible: Тракторный режим

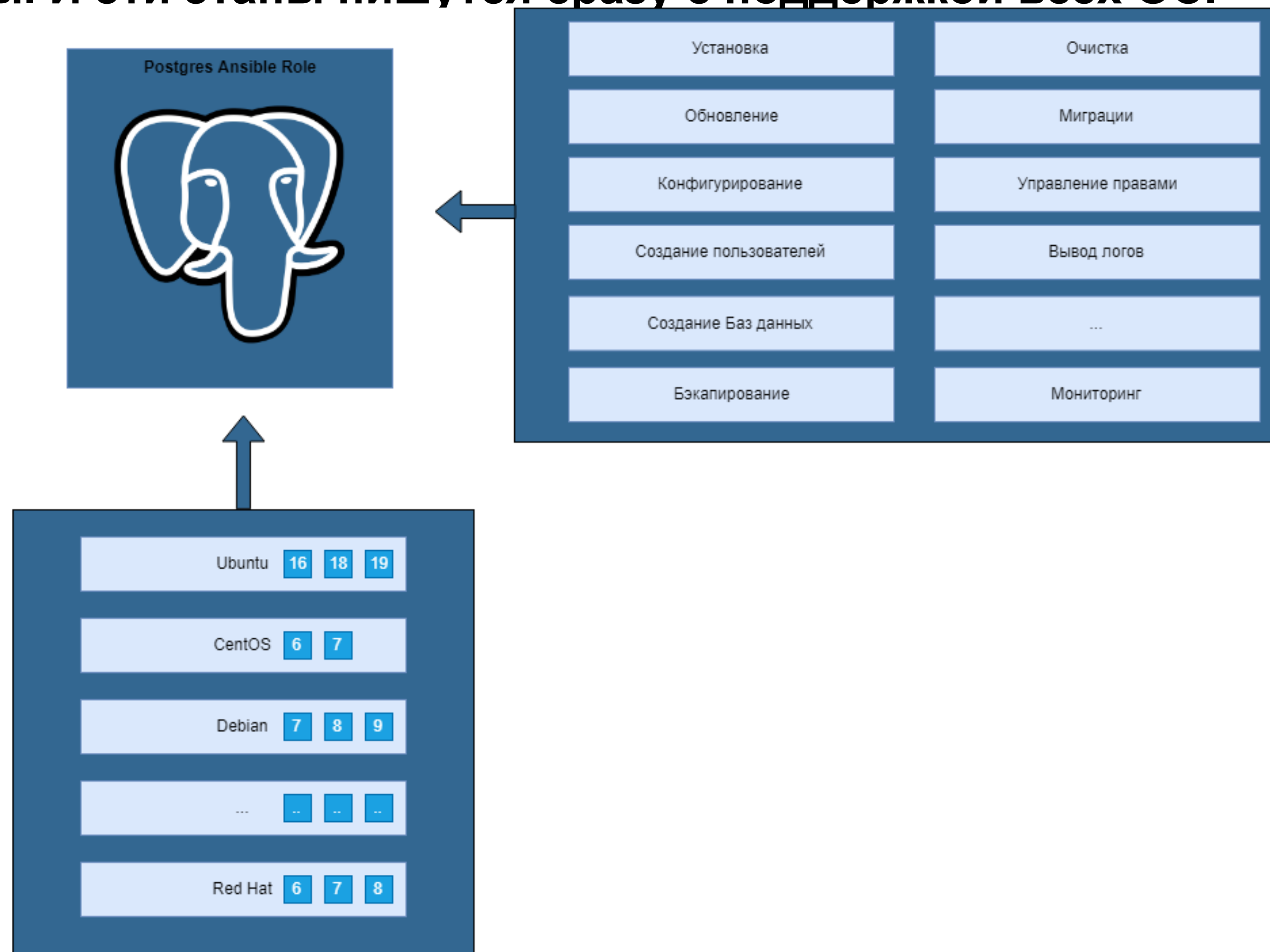
Что такое «Тракторный режим»
Роль должна уметь всё!

- Роль делится на этапы;
- Один вид взаимодействия с сервисом – один этап;
- Всё, что может меняться, выносим в переменные;
- Учитываем разные уровни умения писать роли.

Ansible: Тракторный режим

Что такое «Тракторный режим»
Роль должна уметь всё!

Роль делится на этапы. И эти этапы пишутся сразу с поддержкой всех ОС.



Ansible: Тракторный режим

Что такое «Тракторный режим»

Роль должна уметь всё!

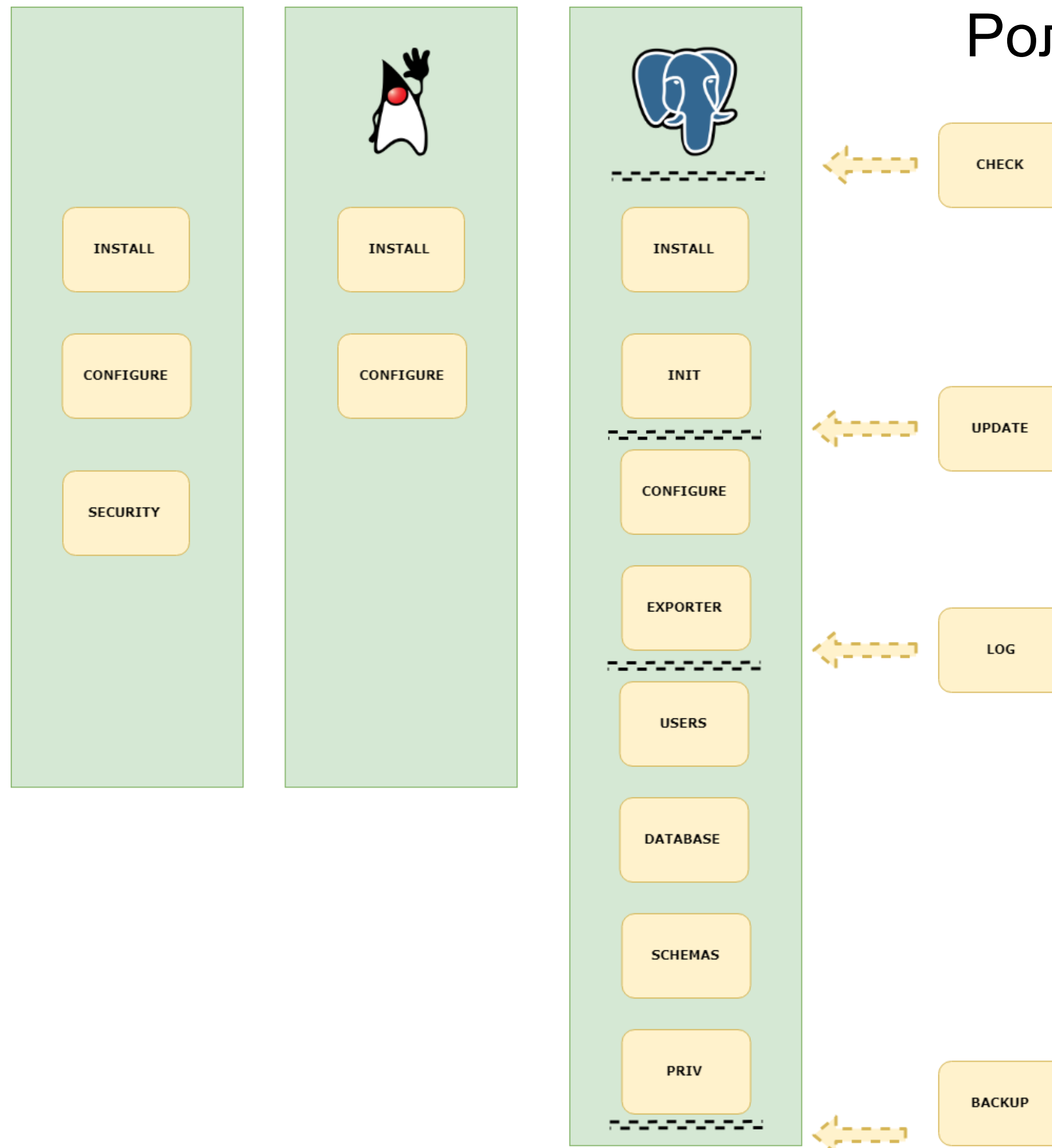
Роль делится на этапы.
И эти этапы пишутся сразу
с поддержкой всех ОС и
тегами.

```
graph TD; Main[Main tasks] --> Install[INSTALL]; Main --> Config[CONFIGURE]; Main --> Security[SECURITY]; Main --> Other[*OTHER*]; Main --> End[END]; Install --- InstallFile["<rolename>/tasks/install-<OS>.yaml"]; Config --- ConfigFile["<rolename>/tasks/configure.yaml"]; Security --- SecurityFile["<rolename>/tasks/security-<OS>.yaml"]; Other --- OtherFile["<rolename>/tasks/*other*.yaml"]; MainFile["<rolename>/tasks/main.yaml"] --- Main;
```

```
main.yml 707 Bytes
1 ---
2 # main tasks file
3
4 # Exec tasks Install
5 - block:
6   - name: "Any | Running the tasks Install"
7     include_tasks: "{{ item }}"
8     with_items:
9       - "install-{{ ansible_distribution | lower }}.yaml"
10    environment: "{{ property_env }}"
11
12 # Exec tasks Configure
13 - block:
14   - name: "Any | Running the tasks Configure"
15     include_tasks: "{{ item }}"
16     with_items:
17       - "configure.yaml"
18    environment: "{{ property_env }}"
19    tags:
20      - conf
21
22 # Exec tasks Security
23 - block:
24   - name: "Any | Running the tasks Security"
25     include_tasks: "{{ item }}"
26     with_items:
27       - "security-{{ ansible_distribution | lower }}.yaml"
28    environment: "{{ property_env }}"
29    tags:
30      - conf
31      - security
32
```

| Name |
|---------------------------|
| .. |
| configure.yaml |
| install-centos.yaml |
| install-oraclelinux.yaml |
| install-ubuntu.yaml |
| main.yml |
| security-centos.yaml |
| security-oraclelinux.yaml |
| security-ubuntu.yaml |

Роль делится на этапы.
И эти этапы пишутся сразу
с поддержкой всех ОС и
тегами.



Ansible: Тракторный режим
Что такое «Тракторный режим»
Роль должна уметь всё!



INSTALL

INIT

CONFIGURE

EXPORTER

USERS

DATABASE

SCHEMAS

PRIV

```

1 ---
2 # Main tasks
3
4 # Run "Install" tasks Install
5 - block:
6   - name: "Any | Running Install"
7     include_tasks: "{{ item }}"
8     with_items:
9       - "install-{{ ansible_distribution | lower }}.yaml"
10    environment: "{{ property_env }}"
11
12 # Run "Init" tasks Init
13 - block:
14   - name: "Any | Running Configure"
15     include_tasks: "{{ item }}"
16     with_items:
17       - "init-{{ ansible_distribution | lower }}.yaml"
18    environment: "{{ property_env }}"
19    tags:
20      - init
21
22 # Run "Configure" tasks Configure
23 - block:
24   - name: "Any | Running Configure"
25     include_tasks: "{{ item }}"
26     with_items:
27       - "configure.yaml"
28    environment: "{{ property_env }}"
29    tags:
30      - conf
31
32 # Run "Security" tasks Security
33 - block:
34   - name: "Any | Running Security"
35     include_tasks: "{{ item }}"
36     with_items:
37       - "security-{{ ansible_distribution | lower }}.yaml"
38    environment: "{{ property_env }}"
39    tags:
40      - conf
41      - security
42

```

```

43 # Run "Exporter" tasks Exporter
44 - block:
45   - name: "Any | Running Exporter"
46     include_tasks: "{{ item }}"
47     with_items:
48       - "exporter.yaml"
49    environment: "{{ property_env }}"
50    tags:
51      - exporter
52
53 # Run "User" tasks Users
54 - block:
55   - name: "Any | Running Users"
56     include_tasks: "{{ item }}"
57     with_items:
58       - "users.yaml"
59    environment: "{{ property_env }}"
60    tags:
61      - user
62
63 # Run "Database" tasks Database
64 - block:
65   - name: "Any | Running Database"
66     include_tasks: "{{ item }}"
67     with_items:
68       - "db.yaml"
69    environment: "{{ property_env }}"
70    tags:
71      - db
72
73 # Run "Scheme" tasks Schemas
74 - block:
75   - name: "Any | Running Database"
76     include_tasks: "{{ item }}"
77     with_items:
78       - "schemas.yaml"
79    environment: "{{ property_env }}"
80    tags:
81      - scheme
82

```

```

83 # Run "Priv" tasks Priv
84 - block:
85   - name: "Any | Running Database"
86     include_tasks: "{{ item }}"
87     with_items:
88       - "priv.yaml"
89    environment: "{{ property_env }}"
90    tags:
91      - priv

```

Ansible: Тракторный режим

Что такое «Тракторный режим»

Роль должна уметь всё!

| Name |
|------|
| .. |

- configure.yaml
- db.yaml
- exporter.yaml
- init-centos.yaml
- init-oraclelinux.yaml
- init-ubuntu.yaml
- install-centos.yaml
- install-oraclelinux.yaml
- install-ubuntu.yaml
- main.yaml
- priv.yaml
- schemas.yaml
- security-centos.yaml
- security-oraclelinux.yaml
- security-ubuntu.yaml
- users.yaml

Ansible: Тракторный режим

Что такое «Тракторный режим»

Роль должна уметь всё!

Теги:

- Любой этап должен быть способен вызываться как отдельно, так и как часть всей роли.

```
53 # Run "User" tasks
54 - block:
55   - name: "Any | Running Users"
56     include_tasks: "{{ item }}"
57     with_items:
58       - "users.yml"
59   environment: "{{ property_env }}"
60   tags:
61     - user
```

| | | |
|-------------------------|--------------|-------------------------|
| postgresql-1-0 | Job Template | ■ □ □ □ □ □ □ □ □ □ □ □ |
| postgresql-1-0-conf | Job Template | ■ □ □ □ □ □ □ □ □ □ □ □ |
| postgresql-1-0-security | Job Template | □ □ □ □ □ □ □ □ □ □ □ □ |
| postgresql-1-1 | Job Template | ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ |
| postgresql-1-1-conf | Job Template | ■ ■ □ □ □ □ □ □ □ □ □ □ |
| postgresql-1-1-db | Job Template | ■ ■ ■ ■ ■ ■ ■ ■ □ □ □ □ |
| postgresql-1-1-exporter | Job Template | ■ ■ □ □ □ □ □ □ □ □ □ □ |
| postgresql-1-1-security | Job Template | ■ ■ □ □ □ □ □ □ □ □ □ □ |
| postgresql-1-1-user | Job Template | ■ ■ ■ ■ □ □ □ □ □ □ □ □ |
| postgresql-dev | Job Template | ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ |

Ansible: Тракторный режим

Что такое «Тракторный режим»

Роль должна уметь всё!

install-centos.yml

install-oraclelinux.yml

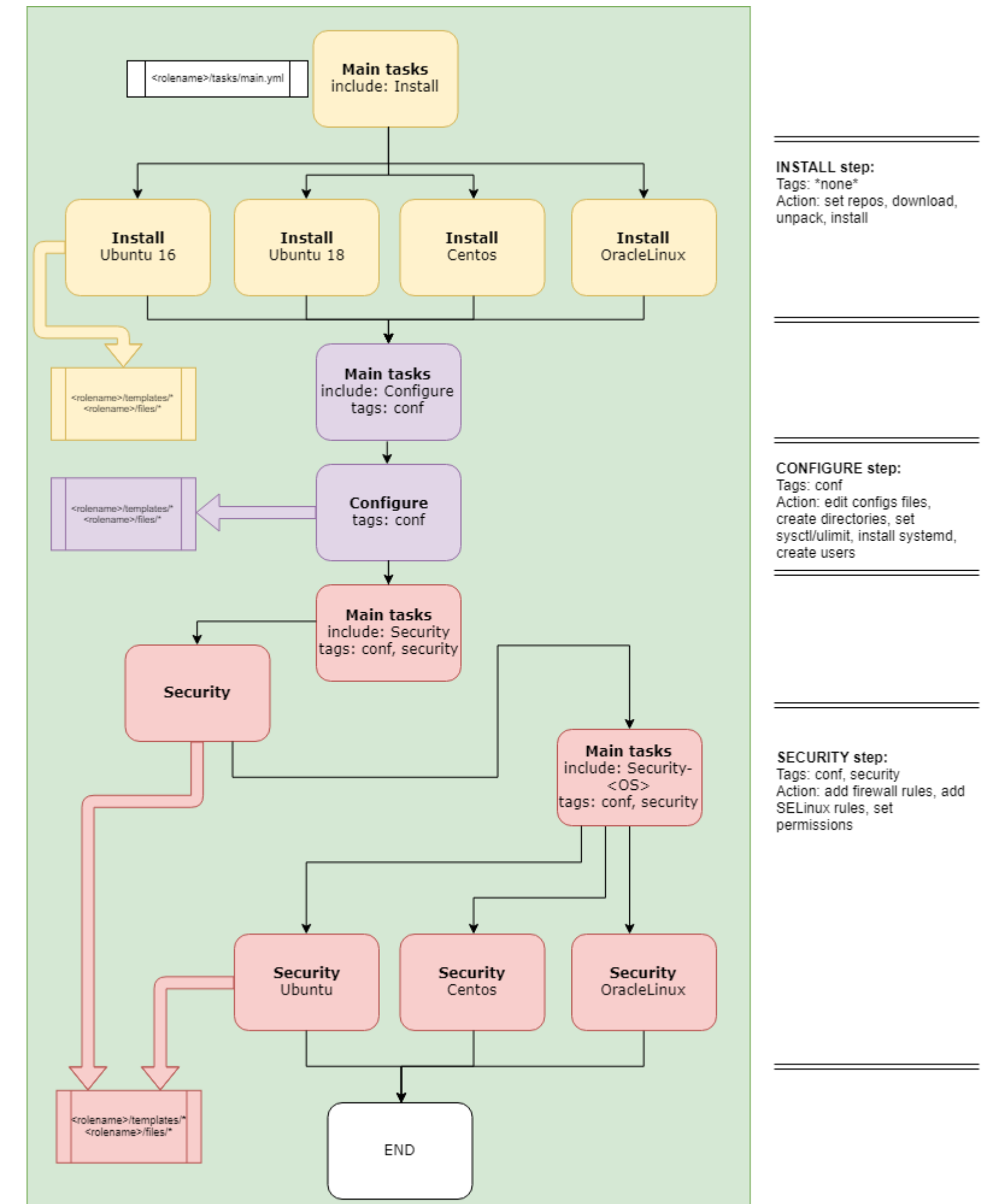
install-ubuntu.yml

Универсальность:

- Роль должна поддерживать все необходимые на данный момент среды (ОС)

Вариации:

- `install-{{ansible_distribution | lower}}.yml`
-> **install install-ubuntu.yml**
- `install-{{ansible_distribution | lower}}-{{ansible_distribution_major_version}}.yml`
-> **install install-ubuntu-18.yml**
- `install-{{ansible_distribution | lower}}-{{ansible_architecture}}.yml`
-> **install-centos-x86_64.yml**



Ansible: Тракторный режим

Что такое «Тракторный режим»

Роль должна уметь всё!

Универсальность:

- Роль должна поддерживать все необходимые на данный момент среды (ОС)

```
4 # Run "Install" tasks
5 - block:
6   - name: "Any | Running Install"
7     include_tasks: "{{ item }}"
8     with_items:
9       - "install-{{ ansible_distribution | lower }}.yaml"
10    environment: "{{ property_env }}"
```

install-centos.yml CentOS

install-oraclelinux.yml

install-ubuntu.yml Ubuntu

```
install-ubuntu.yml 1.68 KB
1 ---
2 # Set repo of the PostgreSQL
3 - name: "Ubuntu | Add key for basic Ubuntu repo"
4   apt_key:
5     url: "{{ item.apt_key_url | default(omit) }}"
6     state: present
7   with_items: "{{ postgresql_ubuntu_18_apt_repo }}"
8   when: ansible_distribution_major_version == '18'
9
10 - name: "Ubuntu | Add key for basic Ubuntu repo"
11   apt_key:
12     url: "{{ item.apt_key_url | default(omit) }}"
13     state: present
14   with_items: "{{ postgresql_ubuntu_16_apt_repo }}"
15   when: ansible_distribution_major_version == '16'
16
17 - name: "Ubuntu | Install basic Ubuntu repo file"
18   apt_repository:
19     repo: "{{ item.repo }}"
20     validate_certs: "{{ item.validate_certs | default('yes') }}"
21     update_cache: "{{ item.update_cache | default('yes') }}"
22     state: "{{ item.state | default('present') }}"
23   with_items: "{{ postgresql_ubuntu_18_apt_repo }}"
24   when: ansible_distribution_major_version == '18'
25
26 - name: "Ubuntu | Install basic Ubuntu repo file"
27   apt_repository:
28     repo: "{{ item.repo }}"
29     validate_certs: "{{ item.validate_certs | default('yes') }}"
30     update_cache: "{{ item.update_cache | default('yes') }}"
31     state: "{{ item.state | default('present') }}"
32   with_items: "{{ postgresql_ubuntu_16_apt_repo }}"
33   when: ansible_distribution_major_version == '16'
34
35 # Intall PostgreSQL
36 - name: "Ubuntu | Install PostgreSQL"
37   apt:
38     name: postgresql-{{ postgresql_version_main | default('11') }}{{ (postgresql_version != 'latest') | ternary('11', '') }}
39     allow_unauthenticated: "{{ (check_security) | ternary('no', 'yes') }}"
40     state: "{{ (postgresql_version != 'latest') | ternary('present', 'latest') }}"
41     retries: 3
42     delay: 3
43     register: result

install-centos.yml 1.1 KB
1 ---
2
3 # Add repos in OS
4 - name: "CentOS | Install basic Centos 7 repos"
5   yum_repository:
6     name: "{{ item.name }}"
7     description: "{{ item.description | default('No descr') }}"
8     baseurl: "{{ item.baseurl }}"
9     gpgcheck: "{{ item.gpgcheck | default('0') }}"
10    gpgkey: "{{ item.gpgkey | default('file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5') }}"
11    state: "{{ item.state | default('present') }}"
12    enabled: "{{ item.enabled | default('yes') }}"
13    priority: "{{ item.priority | default('20') }}"
14    with_items: "{{ postgresql_centos_yum_repo }}"
15
16 # Install PostgreSQL
17 - name: "CentOS | Install PostgreSQL"
18   yum:
19     name: "{{ item }}"
20     validate_certs: "{{ check_security }}"
21     update_cache: yes
22     allow_downgrade: yes
23     state: "{{ (postgres_version != 'latest') | ternary('present', 'latest') }}"
24   with_items:
25     - "{{ postgresql_dependencies }}"
26     - "postgresql{{ postgresql_version_main | default('11') | replace('.', '') }}-server{{ (postgresql
27   retries: 3
28   delay: 3
29   register: result
30   until: result is success
```

Ansible: Тракторный режим

Что такое «Тракторный режим»

Роль должна уметь всё!

Универсальность:

- Роль должна поддерживать все необходимые на данный момент среды (ОС)

```
4 # Run "Install" tasks
5 - block:
6   - name: "Any | Running Install"
7     include_tasks: "{{ item }}"
8     with_items:
9       - "install-{{ ansible_distribution | lower }}.yaml"
10    environment: "{{ property_env }}"
```

install-centos.yml CentOS

install-oraclelinux.yml

install-ubuntu.yml Ubuntu

```
install-ubuntu.yml 1.68 KB
1 ---
2 # Set repo of the PostgreSQL
3 - name: "Ubuntu | Add key for basic Ubuntu repo"
4   apt_key:
5     url: "{{ item.apt_key_url | default(omit) }}"
6     state: present
7   with_items: "{{ postgresql_ubuntu_18_apt_repo }}"
8   when: ansible_distribution_major_version == '18'
9
10 - name: "Ubuntu | Add key for basic Ubuntu repo"
11   apt_key:
12     url: "{{ item.apt_key_url | default(omit) }}"
13     state: present
14   with_items: "{{ postgresql_ubuntu_16_apt_repo }}"
15   when: ansible_distribution_major_version == '16'
16
17 - name: "Ubuntu | Install basic Ubuntu repo file"
18   apt_repository:
19     repo: "{{ item.repo }}"
20     validate_certs: "{{ item.validate_certs | default('yes') }}"
21     update_cache: "{{ item.update_cache | default('yes') }}"
22     state: "{{ item.state | default('present') }}"
23   with_items: "{{ postgresql_ubuntu_18_apt_repo }}"
24   when: ansible_distribution_major_version == '18'
25
26 - name: "Ubuntu | Install basic Ubuntu repo file"
27   apt_repository:
28     repo: "{{ item.repo }}"
29     validate_certs: "{{ item.validate_certs | default('yes') }}"
30     update_cache: "{{ item.update_cache | default('yes') }}"
31     state: "{{ item.state | default('present') }}"
32   with_items: "{{ postgresql_ubuntu_16_apt_repo }}"
33   when: ansible_distribution_major_version == '16'
34
35 # Intall PostgreSQL
36 - name: "Ubuntu | Install PostgreSQL"
37   apt:
38     name: postgresql-{{ postgresql_version_main | default('11') }}{{ (postgresql_version != 'latest') | ternary('11', '') }}
39     allow_unauthenticated: "{{ (check_security) | ternary('no', 'yes') }}"
40     state: "{{ (postgresql_version != 'latest') | ternary('present', 'latest') }}"
41     retries: 3
42     delay: 3
43     register: result

install-centos.yml 1.1 KB
1 ---
2
3 # Add repos in OS
4 - name: "CentOS | Install basic Centos 7 repos"
5   yum_repository:
6     name: "{{ item.name }}"
7     description: "{{ item.description | default('No descr') }}"
8     baseurl: "{{ item.baseurl }}"
9     gpgcheck: "{{ item.gpgcheck | default('0') }}"
10    gpgkey: "{{ item.gpgkey | default('file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5') }}"
11    state: "{{ item.state | default('present') }}"
12    enabled: "{{ item.enabled | default('yes') }}"
13    priority: "{{ item.priority | default('20') }}"
14    with_items: "{{ postgresql_centos_yum_repo }}"
15
16 # Install PostgreSQL
17 - name: "CentOS | Install PostgreSQL"
18   yum:
19     name: "{{ item }}"
20     validate_certs: "{{ check_security }}"
21     update_cache: yes
22     allow_downgrade: yes
23     state: "{{ (postgres_version != 'latest') | ternary('present', 'latest') }}"
24   with_items:
25     - "{{ postgresql_dependencies }}"
26     - "postgresql{{ postgresql_version_main | default('11') | replace('.', '') }}-server{{ (postgresql
27   retries: 3
28   delay: 3
29   register: result
30   until: result is success
```

~~Package!~~

Ansible: Тракторный режим

Что такое «Тракторный режим»
Роль должна уметь всё!

Универсальность:

- Роль должна поддерживать все необходимые на данный момент среды (ОС)

| | |
|-------------------------------|-----------------------------|
| * HOST NAME ? 10.16.75.108 | DESCRIPTION Centos |
| * HOST NAME ? 10.16.75.36 | DESCRIPTION Ubuntu |
| * HOST NAME ? 10.16.75.39 | DESCRIPTION Oracle Linux |

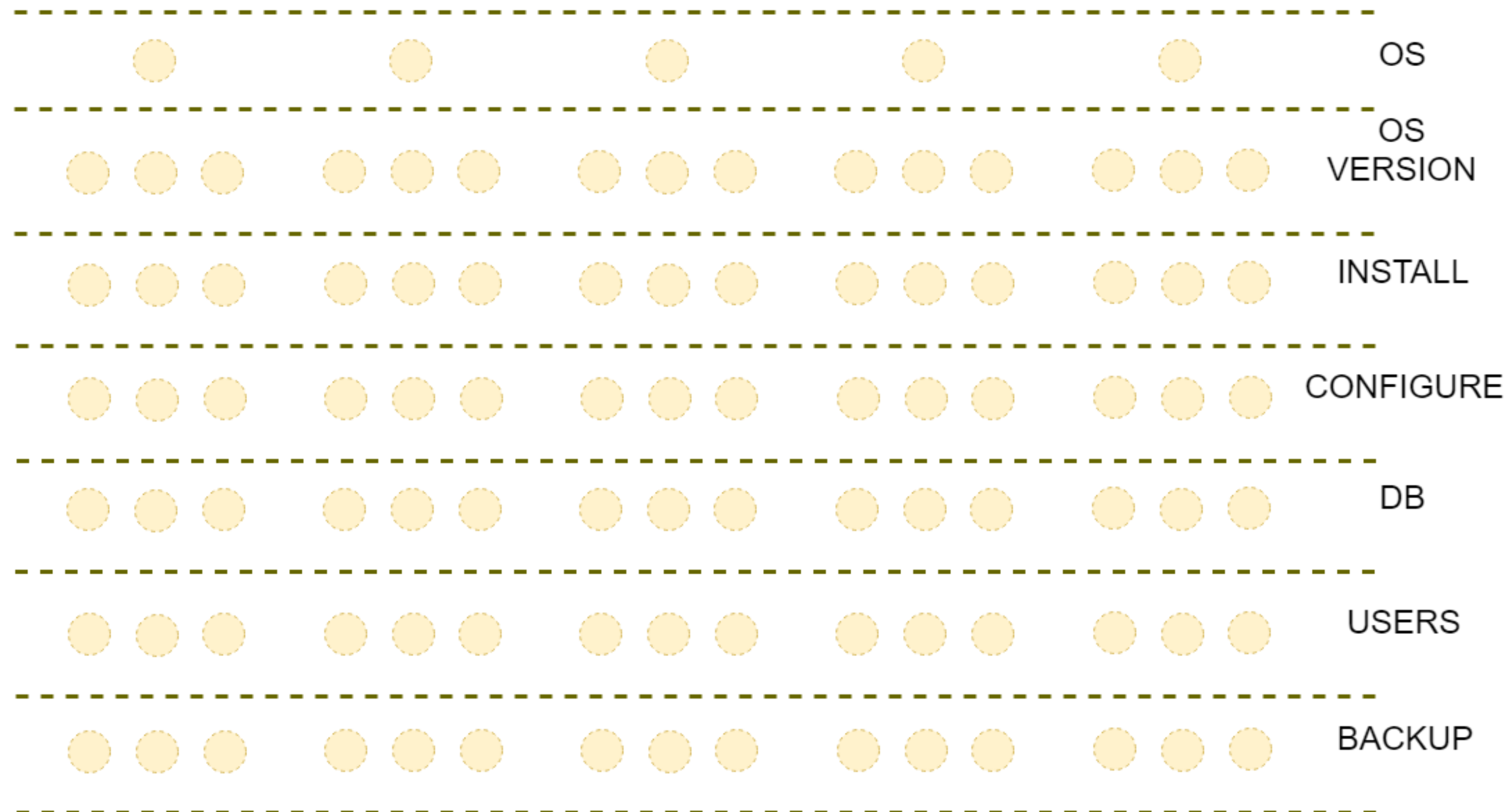
```
TASK [postgresql : Any | Running Install] ***** 17:26:27
task path: /var/lib/awx/projects/_6_postgresql_dev/tasks/main.yml:6
included: /var/lib/awx/projects/_6_postgresql_dev/tasks/install-centos.yml for 10.16.75.108 => (item=install-centos.yml)
included: /var/lib/awx/projects/_6_postgresql_dev/tasks/install-ubuntu.yml for 10.16.75.36 => (item=install-ubuntu.yml)
included: /var/lib/awx/projects/_6_postgresql_dev/tasks/install-oraclelinux.yml for 10.16.75.39 => (item=install-oraclelinux.yml)
```

Ansible: Тракторный режим
Что такое «Тракторный режим»
Роль должна уметь всё!



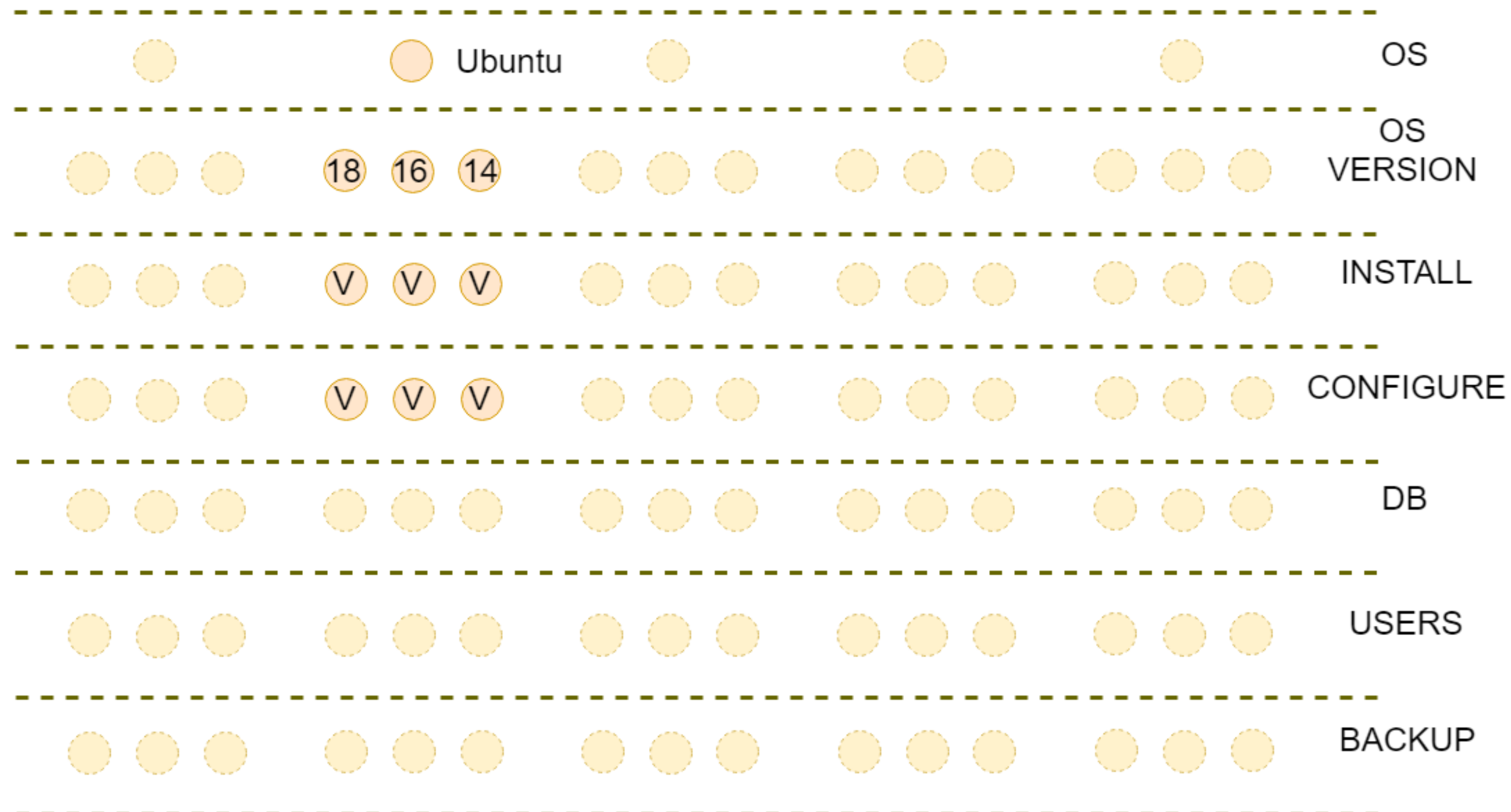
Ansible: Тракторный режим

Что такое «Тракторный режим»
Роль должна уметь всё!



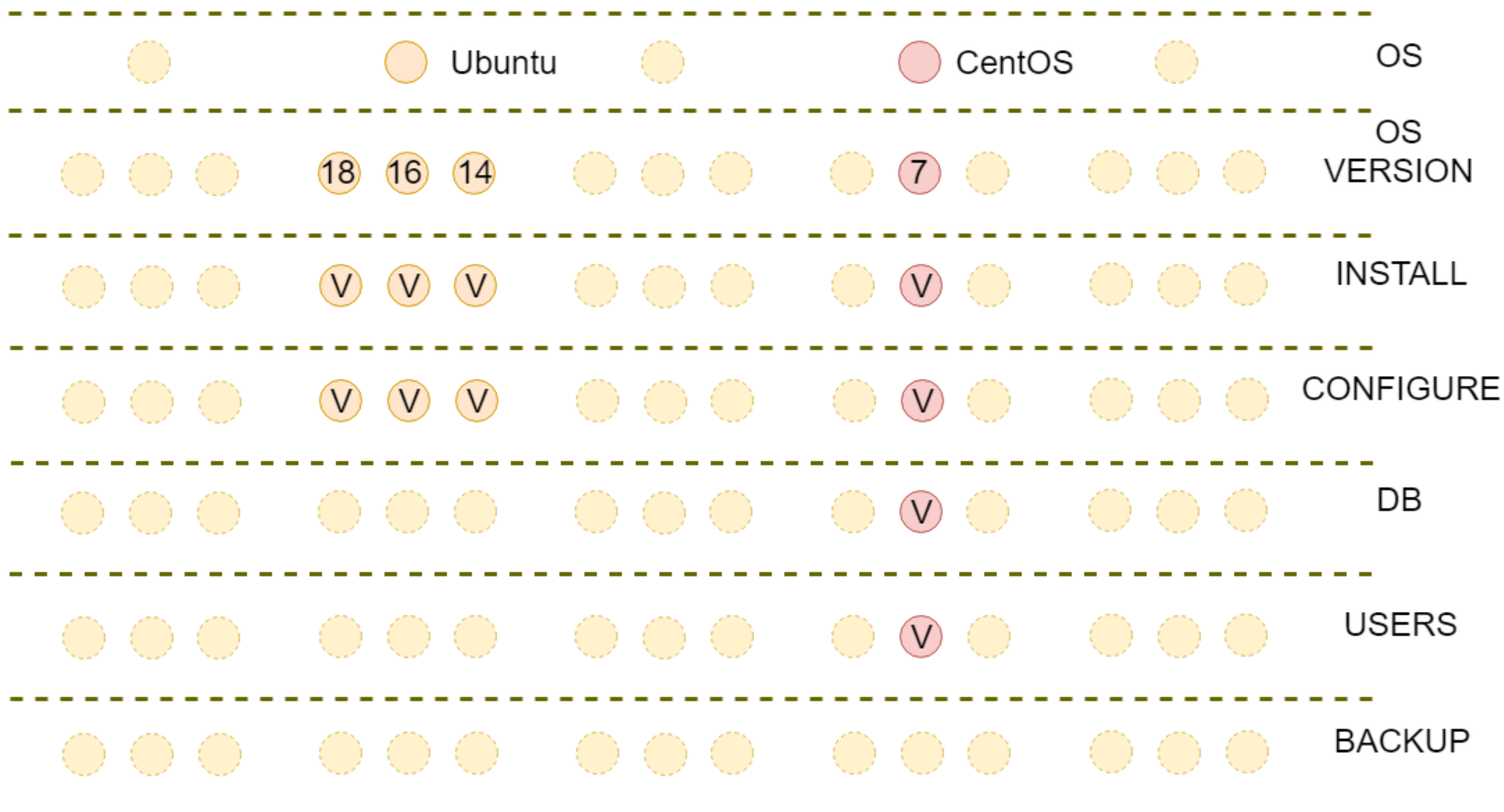
Ansible: Тракторный режим

Что такое «Тракторный режим»
Роль должна уметь всё!



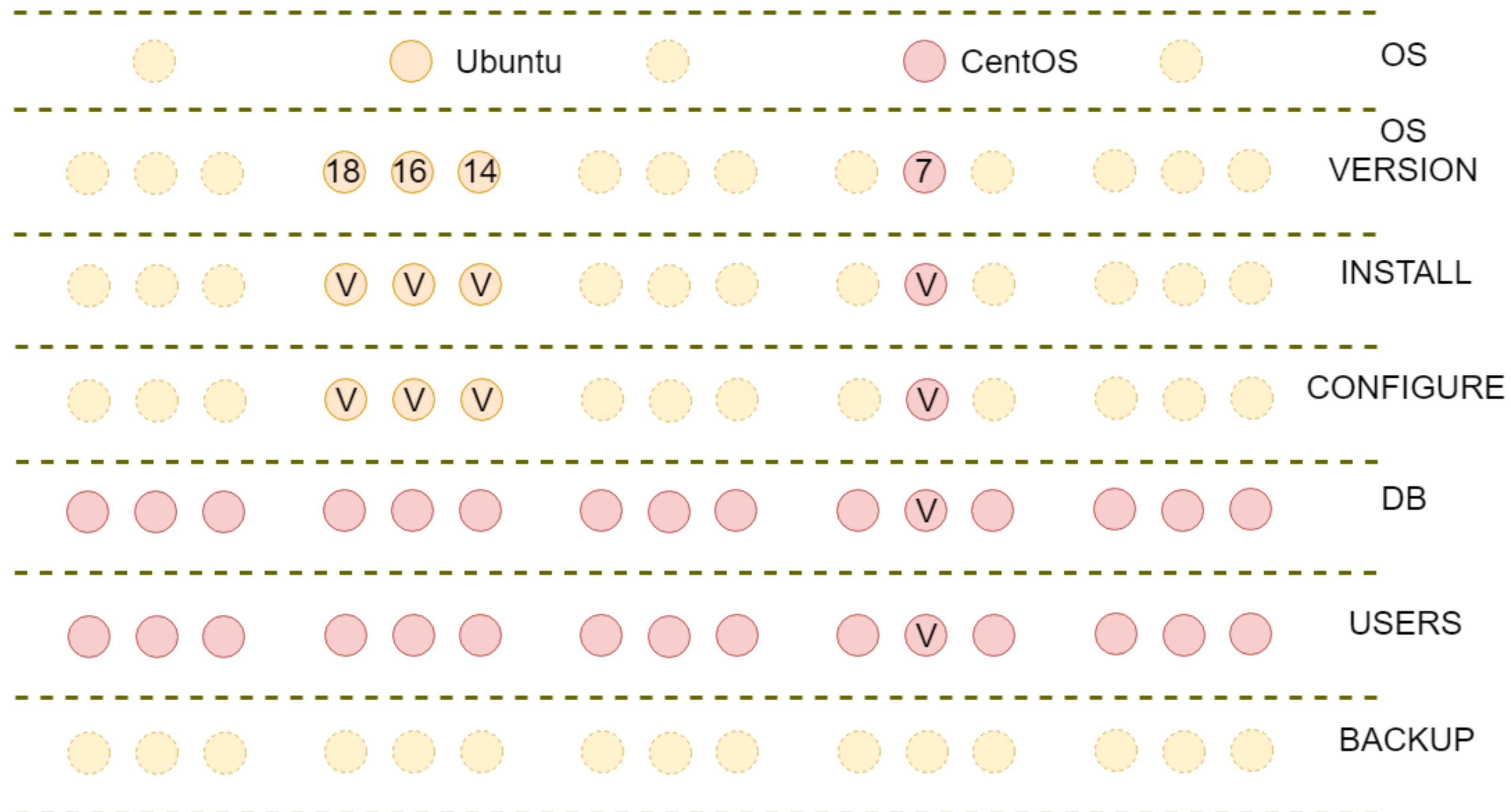
Ansible: Тракторный режим

Что такое «Тракторный режим»
Роль должна уметь всё!



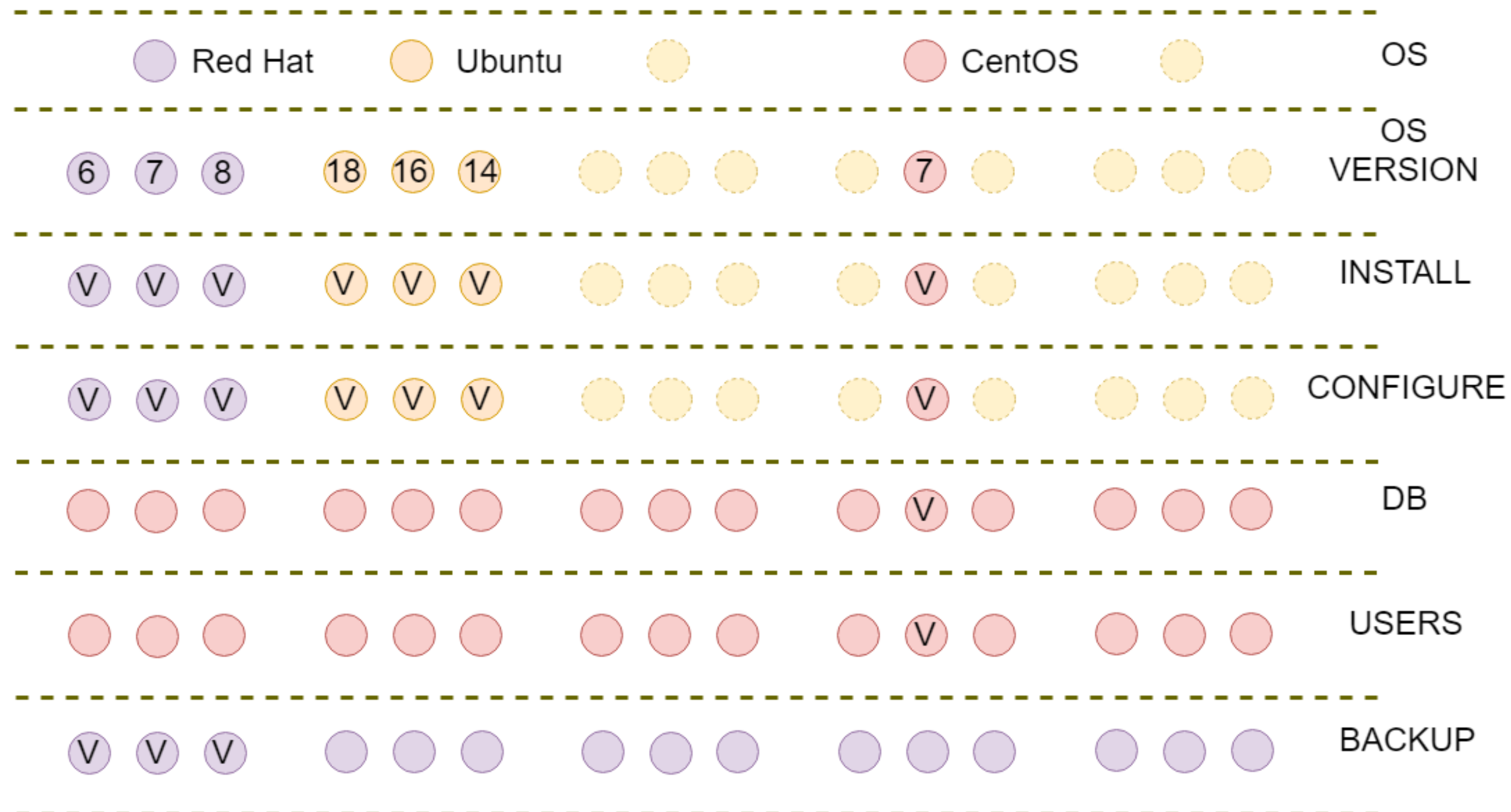
Ansible: Тракторный режим

Что такое «Тракторный режим»
Роль должна уметь всё!



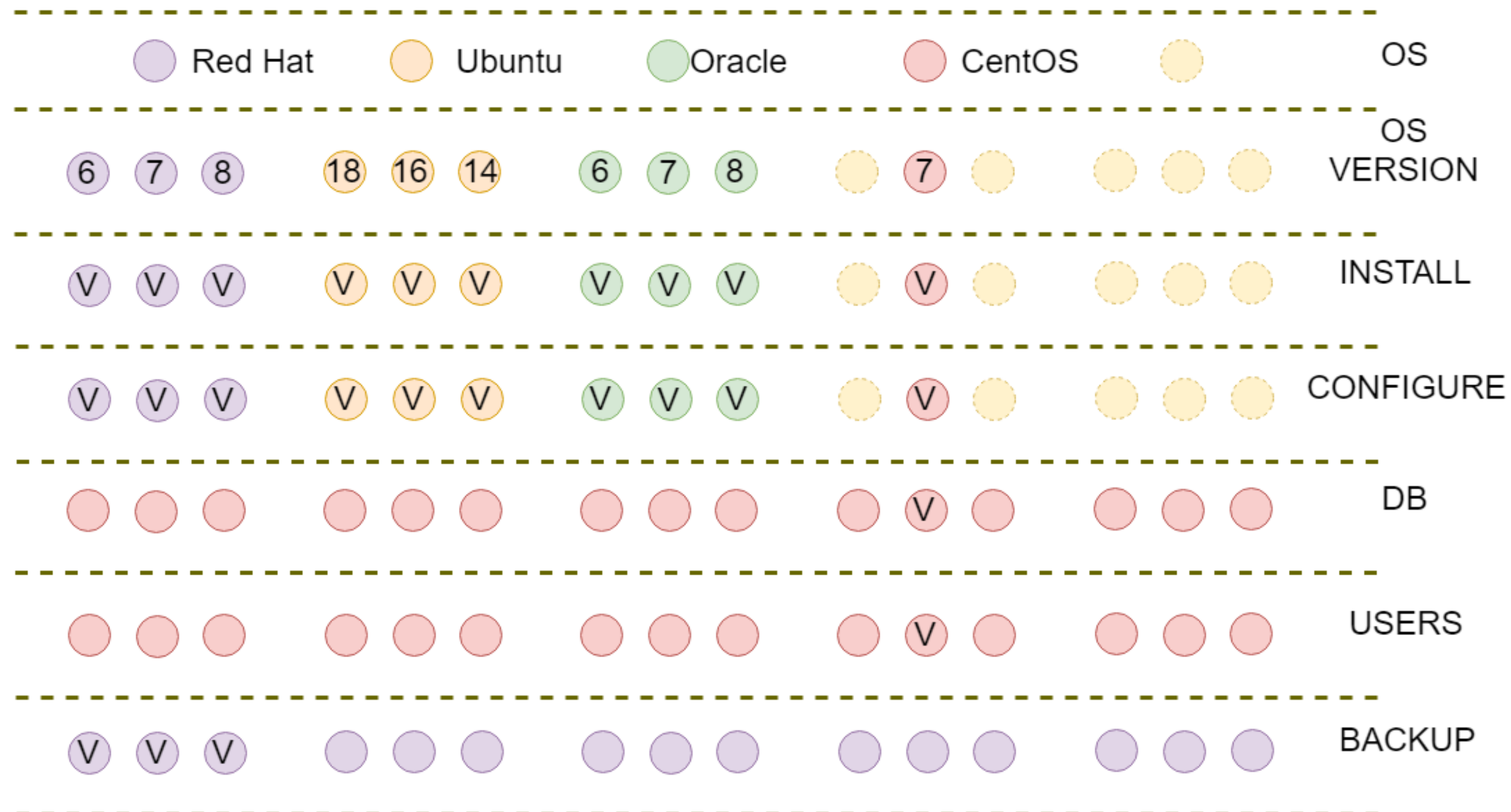
Ansible: Тракторный режим

Что такое «Тракторный режим»
Роль должна уметь всё!



Ansible: Тракторный режим

Что такое «Тракторный режим»
Роль должна уметь всё!



Ansible: Тракторный режим

Что такое «Тракторный режим»

Всё в переменные.

Переменные:

- **Всё, что может меняться, выносим в переменные!**
- **По умолчанию задаём значения по умолчанию!**
- **Значения, которые невозможно предугадать указываем как «обязательные»**

```
3 # Репозитории для Centos
4 postgresql_centos_yum_repo:
5   - name: base
6     description: Default base repository for Centos 7
7     baseurl: http://mirror.centos.org/centos/$releasever/os/$basearch/
8     gpgcheck: "{{ check_security }}"
9     gpgkey: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5
10    enabled: yes
11    state: present
12   - name: updates
13     description: Default update repository for Centos 7
14     baseurl: http://mirror.centos.org/centos/$releasever/updates/$basearch/
15     gpgcheck: "{{ check_security }}"
16     gpgkey: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5
17     enabled: yes
18     state: present
19   - name: postgresql-{{ (postgresql_version != 'latest') | ternary(postgresql_version, '11') }}
20     description: Official repository for PostgreSQL {{ (postgresql_version != 'latest') | ternary(postgresql_version, '11') }}
21     baseurl: https://download.postgresql.org/pub/repos/yum/{{ (postgresql_version != 'latest') | ternary(postgresql_version, '11') }}/redhat/rhel-$releasever
22     gpgcheck: "{{ check_security }}"
23     gpgkey: https://raw.githubusercontent.com/puppetlabs/puppetlabs-postgresql/97af59e701a2aba1187f1fdd10f1c086a59c2a74/files/RPM-GPG-KEY-PGDG
24     enabled: yes
25     state: present
26
```

```
3 # Add repos in OS
4 - name: "CentOS | Install basic Centos 7 repos"
5   yum_repository:
6     name: "{{ item.name }}"
7     description: "{{ item.description | default('No descr') }}"
8     baseurl: "{{ item.baseurl }}"
9     gpgcheck: "{{ item.gpgcheck | default('0') }}"
10    gpgkey: "{{ item.gpgkey | default('file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5') }}"
11    state: "{{ item.state | default('present') }}"
12    enabled: "{{ item.enabled | default('yes') }}"
13    priority: "{{ item.priority | default('20') }}"
14    with_items: "{{ postgresql_centos_yum_repo }}"
```


Ansible: Тракторный режим

Что такое «Тракторный режим»

Всё в переменные.

Переменные:

- Всё, что может меняться, выносим в переменные!
- По умолчанию задаём значения по умолчанию!
- Значения, которые невозможно предугадать указываем как «обязательные»

```
133 # Main Version of PostgreSQL (Official version name of PostgreSQL)
134 postgresql_version_main: 11
135 #postgresql_version_main: 10
136 #postgresql_version_main: 9.6
137 #postgresql_version_main: 9.5
138
139 # Version of PostgreSQL ("latest" or some version)
140 postgresql_version: latest
141 #postgresql_version: 11
142 #postgresql_version: 11.3
143 #postgresql_version: 10
144 #postgresql_version: 10.7
145 #postgresql_version: 9.6.13
146
```

```
35 # Install PostgreSQL
36 - name: "Ubuntu | Install PostgreSQL"
37   apt:
38     name: postgresql-{{ postgresql_version_main | default('11') }}{{ (postgresql_version != 'latest') | ternary('=' ~ postgresql_version ~ '*', '') }}
39     allow_unauthenticated: "{{ (check_security) | ternary('no', 'yes') }}"
40     state: "{{ (postgresql_version != 'latest') | ternary('present', 'latest') }}"
41     retries: 3
42     delay: 3
43     register: result
44     until: result is success
45
```

Ansible: Тракторный режим

Что такое «Тракторный режим»

Всё в переменные.

Ansible Template Tester

Supply your variables in YAML format and template using jinja syntax, or try the [sample](#).

Переменные:

- **Всё, что может меняться, выносим в переменные!**
- **По умолчанию задаём значения по умолчанию!**
- **Значения, которые невозможно предугадать указываем как «обязательные»**

Variables

```
#postgresql_version_main: 9.5

#postgresql_version: 11
#postgresql_version: 11.3
postgresql_version: latest
#postgresql_version: 10
```

Template

Raw (does not require jinja2 delimiters)

```
yum install postgresql{{ postgresql_version_main | default ('11') | replace('.', '') }}-server{{
(postgresql_version != 'latest') | ternary('- ~ postgresql_version ~ '*' , '') }}
/usr/pgsql-{{ postgresql_version_main | default ('11') }}/bin/postgresql-{{ postgresql_version_main | default
('11') | replace('.', '') }}-setup initdb
systemctl start postgresql-{{ postgresql_version_main | default ('11') }}
```

Rendered

```
postgresql-11

yum install postgresql11-server
/usr/pgsql-11/bin/postgresql-11-setup initdb
systemctl start postgresql-11

yum install postgresql11-server
yum install postgresql11-server-11.3*
/usr/pgsql-11/bin/postgresql-11-setup initdb
systemctl start postgresql-11

yum install postgresql10-server
yum install postgresql10-server-10.7*
systemctl start postgresql-10
/usr/pgsql-10/bin/postgresql-10-setup initdb

yum install postgresql96-server
yum install postgresql96-server-9.6.13*
systemctl start postgresql-9.6
/usr/pgsql-9.6/bin/postgresql96-setup initdb
```

```
Terminal - ksnifani@nifaninks:~/ansible
File Edit View Terminal Tabs Help
PLAY [all] *****
TASK [Gathering Facts] *****
ok: [0700ngenie-inside-demo-exp02.ngenie.mtsit.com]

TASK [debug] *****
ok: [0700ngenie-inside-demo-exp02.ngenie.mtsit.com] => {
  "msg": "11"
}

PLAY RECAP *****
0700ngenie-inside-demo-exp02.ngenie.mtsit.com : ok=2    changed=0    un
=0    failed=0

[ksnifani@nifaninks ansible]$ cat testvars.yml
- hosts: all
  vars:
    postgresql_version: "11.3"
  tasks:
    - debug:
      msg: "{{ postgresql_version.split('.')[0] }}"

[ksnifani@nifaninks ansible]$ vi testvars.yml
[ksnifani@nifaninks ansible]$ vi testvars.yml
```


Ansible: Тракторный режим

Что такое «Тракторный режим»

Всё в переменные.

Переменные:

- **Всё, что может меняться, выносим в переменные!**
- **По умолчанию задаём значения по умолчанию!**
- **Значения, которые невозможно предугадать указываем как «обязательные»**

```
158 gitlab_ldap:
159   enable: false
160   label: "LDAP"
161   host: "ldap.exe.com"
162   port: 389 #'398' for unsecurity, '636' for security
163   uid: 'uid' #'uid' for OpenLDAP, 'sAMAccountName' for AD
164   ad: false #'true' for ActiveDirectory, 'false' for other
165   base_dn: "cn=user,dc=exe,dc=com"
166   base_pwd: "secret123"
167   base: "ou=People,dc=exe,dc=com"
168   base_group: "ou=Groups,dc=exe,dc=com"
169   admin_group: "admins"
170   verify: false #'false' for 389 port, 'true' for other
171   encrypt: "plain" #"plain" for unsecurity, "simple_tls" security
172
173 gitlab_email:
174   enable: false
175   host: "mail.exe.com"
176   port: 25
177   user: "no-reply@exe.com"
178   password: "secret123"
179   domain: "exe.com"
180   display_name: "Postal of GitLab service"
181   from: "no-reply@exe.com"
182   reply_to: "reply@exe.com"
183
```

```
185 # Global configuration options that will be set in postgresql.conf.
186 postgresql_global_config_options:
187   - option: unix_socket_directories
188     value: '{{ postgresql_unix_socket_directories | join(",") }}'
189   - option: port
190     value: "{{ postgresql_server_port }}"
191   - option: listen_addresses
192     value: '{{ postgresql_listen_addresses | join(",") }}'
193   - option: max_connections
194     value: "{{ postgresql_max_connections }}"
195
196 postgresql_default_auth_method: "trust"
197
198 # Host based authentication (hba) entries to be added to the pg_hba.conf. This
199 # variable's defaults reflect the defaults that come with a fresh installation.
200 postgresql_pg_hba_entries_default:
201   - { type: local, database: all, user: postgres, auth_method: peer, }
202   - { type: local, database: all, user: all, auth_method: "{{ postgresql_default_auth_method }}", comment: '"local" is for Unix domain socket connections o
203   - { type: host, database: all, user: all, address: '127.0.0.1/32', auth_method: md5, comment: 'IPv4 local connections:' }
204   - { type: host, database: all, user: all, address: ':::1/128', auth_method: "{{ postgresql_default_auth_method }}", comment: 'IPv6 local connections:' }
205
```

```
154 # Basic settings
155 postgresql_encoding: 'UTF-8'
156 # Used to execute initdb
157 postgresql_bin_directory: "/usr/pgsql-{{ postgresql_version }}/bin"
158
159 postgresql_locale_parts:
160   - 'en_US' # Locale
161   - 'UTF-8' # Encoding
162 postgresql_locale: "{{ postgresql_locale_parts | join('.') }}"
163 postgresql_ctype_parts:
164   - 'en_US' # Locale
165   - 'UTF-8' # Encoding
166 postgresql_ctype: "{{ postgresql_ctype_parts | join('.') }}"
167
168 # The user/group that will run postgresql process
169 postgresql_user: postgres
170 postgresql_group: postgres
171
172 postgresql_listen_addresses:
173   - localhost
174
175 postgresql_server_port: 5432
176
177 postgresql_database_path: "/var/lib/pgsql/"
178 postgresql_unix_socket_directories:
179   - /var/run/postgresql
180
```


Ansible: Тракторный режим
Что такое «Тракторный режим»
Всё в переменные

Как не запутаться?

Ansible: Тракторный режим

Что такое «Тракторный режим»

Обязательное использование default и запрещение использование vars

Сила Default

postgresql-nodes ✕

DETAILS GROUPS HOSTS

* NAME DESCRIPTION

postgresql-nodes

VARIABLES ? YAML JSON

```
1 ---
2 postgresql_version_main: 10
```

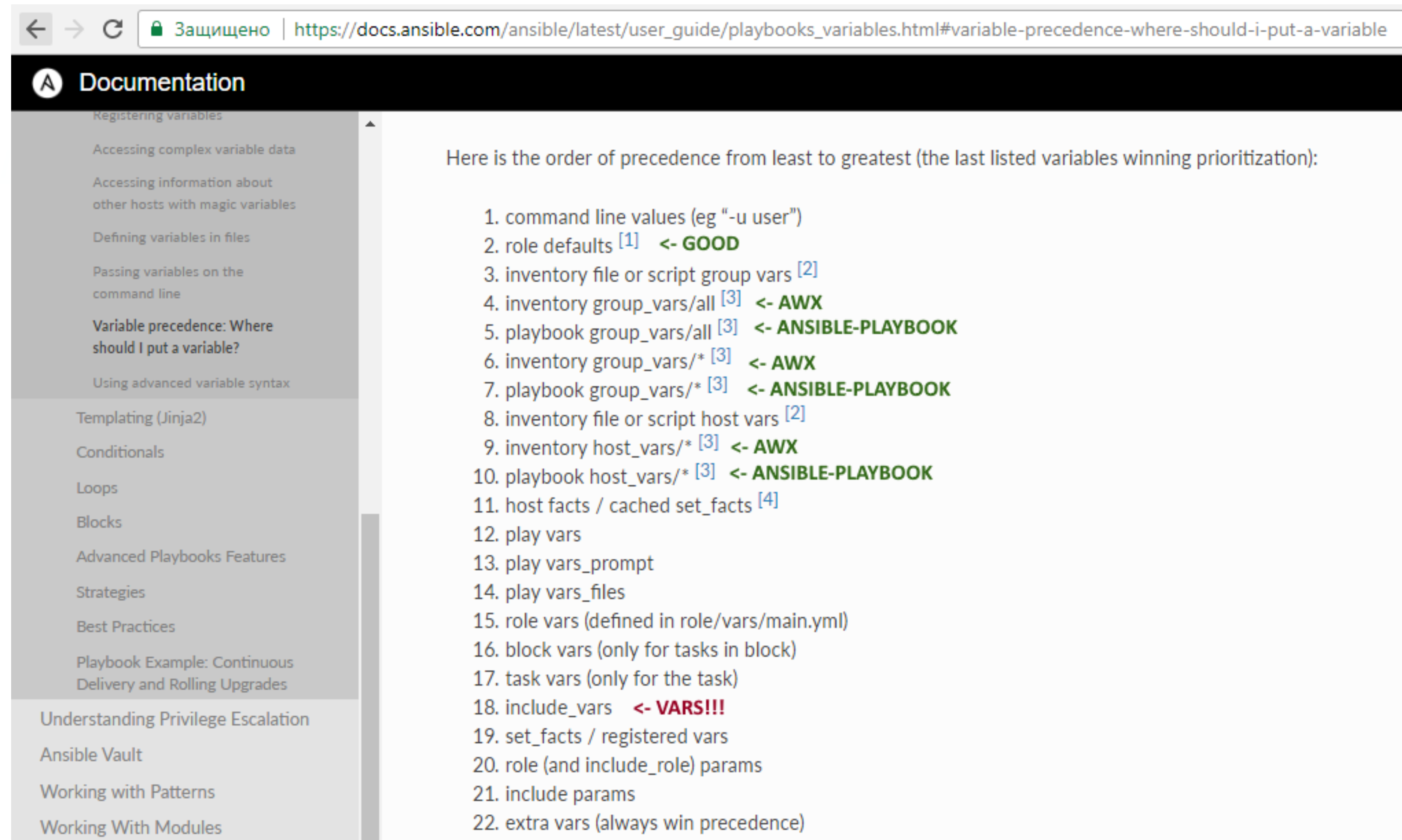
CANCEL SAVE

Ansible: Тракторный режим

Что такое «Тракторный режим»

Обязательное использование default и запрещение использование vars

Бессилие Vars



The screenshot shows a web browser window with the URL https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html#variable-precedence-where-should-i-put-a-variable. The page title is "Documentation" and the current page is "Variable precedence: Where should I put a variable?". The main content area lists the order of precedence from least to greatest (the last listed variables winning prioritization):

1. command line values (eg "-u user")
2. role defaults ^[1] <- **GOOD**
3. inventory file or script group vars ^[2]
4. inventory group_vars/all ^[3] <- **AWX**
5. playbook group_vars/all ^[3] <- **ANSIBLE-PLAYBOOK**
6. inventory group_vars/* ^[3] <- **AWX**
7. playbook group_vars/* ^[3] <- **ANSIBLE-PLAYBOOK**
8. inventory file or script host vars ^[2]
9. inventory host_vars/* ^[3] <- **AWX**
10. playbook host_vars/* ^[3] <- **ANSIBLE-PLAYBOOK**
11. host facts / cached set_facts ^[4]
12. play vars
13. play vars_prompt
14. play vars_files
15. role vars (defined in role/vars/main.yml)
16. block vars (only for tasks in block)
17. task vars (only for the task)
18. include_vars <- **VARS!!!**
19. set_facts / registered vars
20. role (and include_role) params
21. include params
22. extra vars (always win precedence)

Ansible: Тракторный режим

Что такое «Тракторный режим»

Обязательное использование default и запрещение использование vars

Но не всё должно
быть в default

| | | |
|-------------------------------------|-----|---|
| <code>gitlab_url</code> | Yes | Userway URL for GitLab |
| <code>kibana_elastic_host</code> | Yes | Elasticsearch host's IP address |
| <code>vrrp_virtual_router_id</code> | Yes | numerical (0..255) identifier for the Virtual Router instance. It must be the same on all Keepalived nodes in this cluster. |
| <code>ad_sync_user</code> | Yes | DN of user for sync |
| <code>ad_sync_user_password</code> | Yes | Password of user for sync |

Ansible: Тракторный режим

Что такое «Тракторный режим»

Преобладание `include_tasks` над `when` и `result`

Основа роли – `include_tasks`. `When` и `result` использовать НЕ запрещается, но их нужно использовать только для единичных случаев.

```
install-ubuntu.yml 1.68 KB
1 ---
2 # Set repo of the PostgreSQL
3 - name: "Ubuntu | Add key for basic Ubuntu repo"
4   apt_key:
5     url: "{{ item.apt_key_url | default (omit) }}"
6     state: present
7   with_items: "{{ postgresql_ubuntu_18_apt_repo }}"
8   when: ansible_distribution_major_version == '18'
9
10 - name: "Ubuntu | Add key for basic Ubuntu repo"
11   apt_key:
12     url: "{{ item.apt_key_url | default (omit) }}"
13     state: present
14   with_items: "{{ postgresql_ubuntu_16_apt_repo }}"
15   when: ansible_distribution_major_version == '16'
16
17 - name: "Ubuntu | Install basic Ubuntu repo file"
18   apt_repository:
19     repo: "{{ item.repo }}"
20     validate_certs: "{{ item.validate_certs | default ('yes') }}"
21     update_cache: "{{ item.update_cache | default ('yes') }}"
22     state: "{{ item.state | default('present') }}"
23   with_items: "{{ postgresql_ubuntu_18_apt_repo }}"
24   when: ansible_distribution_major_version == '18'
25
26 - name: "Ubuntu | Install basic Ubuntu repo file"
27   apt_repository:
28     repo: "{{ item.repo }}"
29     validate_certs: "{{ item.validate_certs | default ('yes') }}"
30     update_cache: "{{ item.update_cache | default ('yes') }}"
31     state: "{{ item.state | default('present') }}"
32   with_items: "{{ postgresql_ubuntu_16_apt_repo }}"
33   when: ansible_distribution_major_version == '16'
34
```


Ansible: Тракторный режим


Что такое «Тракторный режим»


Преобладание `include_tasks` над `when` и `result`

С другой стороны, уход в `include_tasks` может принести больше проблемам с множеством файлов.


В итоге, тут выбор `when` или `include_tasks` остаётся на том, кто пишет роль. Основная цель – простота.


 `install-centos.yml`


 `install-oraclelinux.yml`


 `install-ubuntu.yml`





 `install-centos-6.yml`


 `install-centos-7.yml`

 `install-oraclelinux-6.yml`

 `install-oraclelinux-7.yml`

 `install-oraclelinux-8.yml`

 `install-ubuntu-16.yml`

 `install-ubuntu-18.yml`

Ansible: Тракторный режим

Что такое «Тракторный режим»

Обязательное заполнение README

```
README.md

# PostgreSQL

Version 1.2

An Ansible role for install and configure PostgreSQL, extensions, databases and users.

Requirements

- Ansible => 2.7
```

Ansible: Тракторный режим

Что такое «Тракторный режим»

Обязательное заполнение README

Role Tags

You can use follow a tags for controlling your exec

| Tag name | Description |
|-----------------------|--|
| <code>conf</code> | Configuring PosgreSQL service (include security) |
| <code>security</code> | Configure SELinux and/or Firewalld |
| <code>init</code> | Primary initialization of PosgreSQL Database |
| <code>exporter</code> | Install Exporter for PostgreSQL |
| <code>user</code> | Creating users for PostgreSQL |
| <code>db</code> | Creating databases for PostgreSQL |
| <code>scheme</code> | Creating schemas for PostgreSQL |
| <code>priv</code> | Alter privileges for PostgreSQL |

Ansible: Тракторный режим

Что такое «Тракторный режим»

Обязательное заполнение README

Role Variables

Available variables are listed below, along with default values. See [defaults/main.yml](#).

You can change all variables by `group_vars` or `host_vars` for `ansible-playbook` or by `Inventory`, `Inventory-group` or `Inventory-host` for `AWX`

| Variable name | Required* | Description | Default Value | Multivalue |
|--|-----------|---|---------------------------------------|------------|
| <code>property_env</code> | No | Other, input parameters (Proxy, locale, lang, etc) | | No |
| <code>check_security</code> | No | Check SSL for HTTPS, GPG for yum, etc | yes | No |
| <code>selinux</code> | No | State of the selinux | enforcing | No |
| <code>firewall</code> | No | State of firewalld service | yes | No |
| <code>postgresql_centos_yum_repo</code> | No | Repos for Centos. | See defaults/main.yml | Yes |
| <code>postgresql_oraclelinux_yum_repo</code> | No | Repos for Oracle Linux. | See defaults/main.yml | Yes |
| <code>postgresql_ubuntu_18_apt_repo</code> | No | Repos for Ubuntu 18. | See defaults/main.yml | Yes |
| <code>postgresql_ubuntu_16_apt_repo</code> | No | Repos for Ubuntu 16. | See defaults/main.yml | Yes |

Ansible: Тракторный режим

Что такое «Тракторный режим»

Обязательное заполнение README

Parameters of the pg_hba.conf file set in `postgresql_pg_hba_entries_default`

Example:

```
postgresql_pg_hba_entries_default:
- { type: local, database: all, user: postgres, auth_method: peer, }
- { type: local, database: all, user: all, auth_method: "{{ postgresql_default_auth_method }}", comment: '"local connections"' }
- { type: host, database: all, user: all, address: '127.0.0.1/32', auth_method: md5, comment: 'IPv4 local connections' }
- { type: host, database: all, user: all, address: '::1/128', auth_method: "{{ postgresql_default_auth_method }}", comment: '"IPv6 local connections"' }
```

If you have other parameters set `property_env`

Examples:

```
# No one (empty)
property_env: []

# Set proxy
property_env:
  http_proxy: "http://LOGIN:PASSWORD@HOST:PORT"
  HTTP_PROXY: "http://LOGIN:PASSWORD@HOST:PORT"
  https_proxy: "http://LOGIN:PASSWORD@HOST:PORT"
  HTTPS_PROXY: "http://LOGIN:PASSWORD@HOST:PORT"
  ftp_proxy: "http://LOGIN:PASSWORD@HOST:PORT"
  FTP_PROXY: "http://LOGIN:PASSWORD@HOST:PORT"
  no_proxy: "localhost,127.0.0.1,10.0.0.0/8,172.16.0.0/12,192.168.0.0/16"
  NO_PROXY: "localhost,127.0.0.1,10.0.0.0/8,172.16.0.0/12,192.168.0.0/16"

# Set lang and locale
property_env:
  LANG: "ru_RU.UTF-8"
```

Ansible: Тракторный режим

Что такое «Тракторный режим»

Обязательное заполнение README

Dependencies

- none

Example Playbook

```
- name: Install PostgreSQL
  hosts:
    - postgresql-nodes
  become: yes
  become_method: sudo
  roles:
    - postgresql
```

TODO

More information

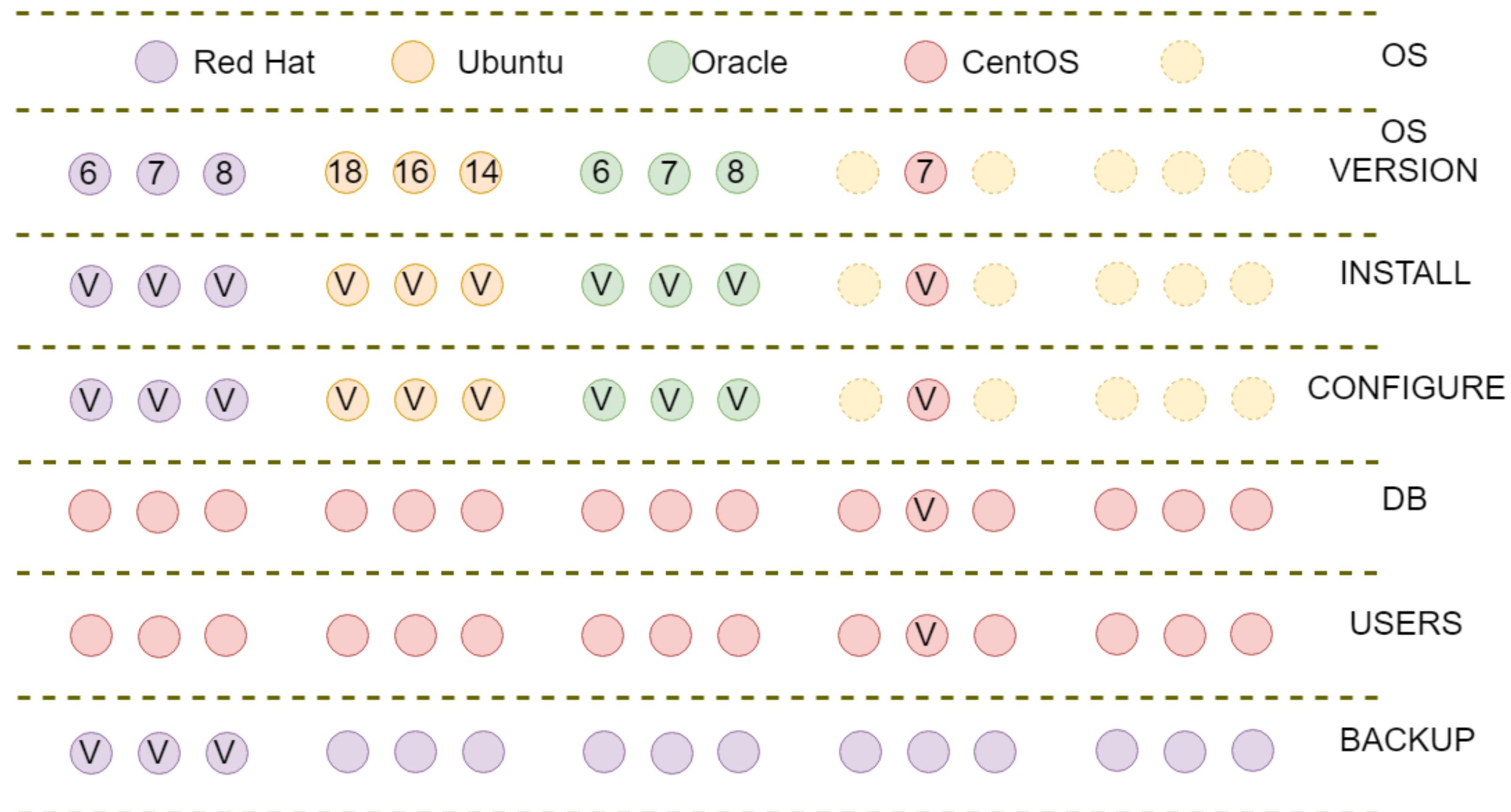
(~~Strike through~~ the implemented)

- ~~Ubuntu 18 support~~
- ~~Ubuntu 16 support~~
- ~~Centos 7 support~~
- ~~Oracle Linux 7 support~~
- ~~Comments for any tasks~~
- ~~Name (OS|DO) for any tasks~~
- ~~Support multivalue for major variables~~
- ~~Hidden passwords~~
- ~~Enable/Disable check of SSL(GPG) certs(keys)~~

Ansible: Тракторный режим

Что такое «Тракторный режим»

Каждый пишет по мере своих знаний, но не ломай чужое



Ansible: Тракторный режим Что такое «Тракторный режим»

Каждый пишет по мере своих знаний, но не ломай чужое

```
175 postgresql_unix_socket_directories:
176   - /var/run/postgresql
177
178 postgresql_max_connections: "300"
179 #not use /tmp dir ansible not work because permissions issue
180 # - /tmp
181
182 # Global configuration options that will be set in postgresql.conf.
183 postgresql_global_config_options:
184   - option: unix_socket_directories
185     value: '{{ postgresql_unix_socket_directories | join(",") }}'
186   - option: port
187     value: "{{ postgresql_server_port }}"
188   - option: listen_addresses
189     value: '{{ postgresql_listen_addresses | join(",") }}'
190   - option: max_connections
191     value: "{{ postgresql_max_connections }}"
192
```

```
1 ---
2 - name: configure | Ensure PostgreSQL unix socket dirs exist.
3   file:
4     path: "{{ item }}"
5     state: directory
6     owner: "{{ postgresql_user }}"
7     group: "{{ postgresql_group }}"
8     mode: 02775
9   with_items: "{{ postgresql_unix_socket_directories }}"
10
```

Ansible: Тракторный режим

Что такое «Тракторный режим»

Роль должна запускаться как под ansible-playbook, так и через AWX



```
[ksnifani@nifaninks postgresql]$ ls -la roles/  
total 8  
drwxr-xr-x.  2 ksnifani admsk 4096 Aug 29 15:34 .  
drwxr-xr-x. 11 ksnifani admsk 4096 Aug 29 15:33 ..  
lrwxrwxrwx.  1 ksnifani admsk    2 Aug 29 15:34 postgresql -> ..
```

Ansible: Тракторный режим

Какие сложности возникли при переходе на него

3 проблемы:

- **Высокий порог входа**
- **Проблема сторожил**
- **Контроль**

Ansible: Тракторный режим

Какие сложности возникли при переходе на него

Высокий порог входа

```
7 - name: database | Ensure PostgreSQL databases are present.
8 postgresql_db:
9   name: "{{ item.name }}"
10  lc_collate: "{{ item.lc_collate | default(postgresql_locale) }}"
11  lc_ctype: "{{ item.lc_ctype | default(postgresql_ctype) }}"
12  encoding: "{{ item.encoding | default(postgresql_encoding) }}"
13  template: "{{ item.template | default('template0') }}"
14  login_host: "{{ item.login_host | default('localhost') }}"
15  login_password: "{{ item.login_password | default(omit) }}"
16  login_user: "{{ item.login_user | default(postgresql_user) }}"
17  login_unix_socket: "{{ item.login_unix_socket | default(postgresql_unix_socket_directories[0]) }}"
18  port: "{{ item.port | default(postgresql_server_port) }}"
19  owner: "{{ item.owner | default(postgresql_user) }}"
20  state: "{{ item.state | default('present') }}"
21 with_items: "{{ postgresql_databases }}"
22 become: yes
23 become_user: "{{ postgresql_user }}"
24 # See: https://github.com/ansible/ansible/issues/16048#issuecomment-229012509
25 vars:
26   ansible_ssh_pipelining: true
27 when: postgresql_databases|length > 0
```



```
# Install PostgreSQL
- name: "CentOS | Install PostgreSQL"
  yum:
    name: "{{ item }}"
    validate_certs: "{{ check_security }}"
    update_cache: yes
    allow_downgrade: yes
    state: "{{ (postgresql_version != 'latest') | ternary('present', 'latest') }}"
  with_items:
    - "{{ postgresql_dependencies }}"
    - "postgresql{{ postgresql_version_main | default('11') | replace('.', '') }}-server"
  retries: 3
  delay: 3
  register: result
  until: result is success
```

README.md

POSTGRESQL

Version 1.2

An Ansible role for install and configure PostgreSQL, extensions, databases and users.

Ansible: Тракторный режим

Какие сложности возникли при переходе на него

Высокий порог входа

Работа с ролями Ansible (Ngenie)

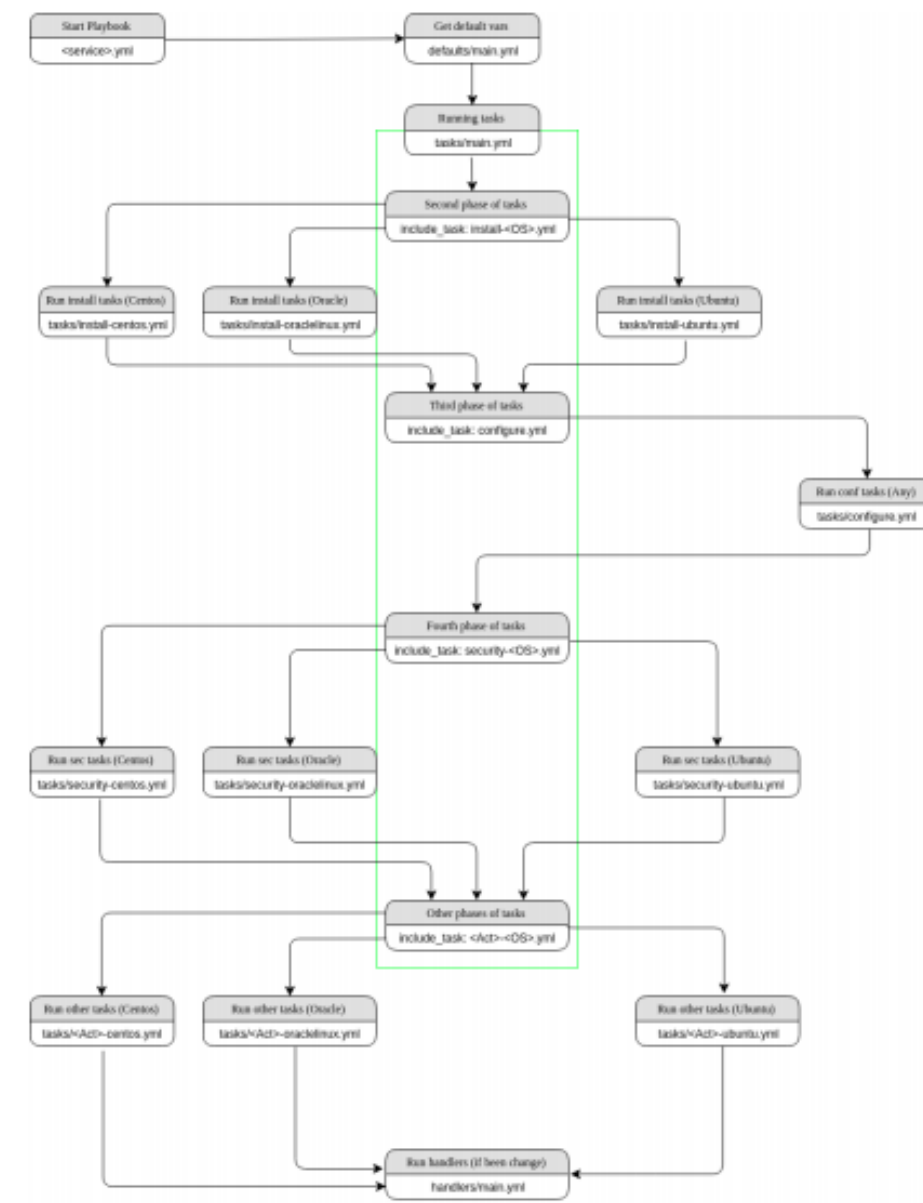
Работа с ролями Ansible (Ngenie)

- Работа с ролями Ansible (Ngenie)
 - Описание
 - Основные понятия
 - Структура и использование роли
 - Краткое описание
 - Схема
 - Зелёная зона
 - GitLab
 - НУЖНО
 - НЕЛЬЗЯ
 - МОЖНО
 - Структура файлов роли
 - НУЖНО
 - НЕЛЬЗЯ
 - МОЖНО
 - Поддержка разных сред
 - НУЖНО
 - НЕЛЬЗЯ
 - МОЖНО
 - Переменные
 - НУЖНО
 - НЕЛЬЗЯ
 - МОЖНО
 - Задачи
 - НУЖНО
 - НЕЛЬЗЯ
 - МОЖНО
 - Зависимости
 - НУЖНО
 - НЕЛЬЗЯ
 - МОЖНО
 - Синяя зона
 - Структура AWX
 - Credentials
 - Projects
 - Inventories
 - Template
 - Jobs (запуск)
 - Процесс релиза
- Вопросы

Описание

В данной статье перечислены основные правила по созданию и запуску Ansible ролей.

1 Изначально данные правила приняты в трайбе Ngenie, однако они могут быть применены и в других проектах.
Правила для BSS отличаются от этих и находятся в Space "Интеграция"



Порядок и зоны

- Использовать заглавные буквы в именах каталогов и файлов (кроме файлов *.md)

МОЖНО

- Добавлять дополнительные каталоги и файлы Ansible-иерархии, если это является необходимым для работы роли (например, каталог files, tasks/ingress.yml, etc).
- Добавлять дополнительные *.md файлы в корень роли
- Файл <rolename>.yml может содержать следующие параметры:
 - hosts - называться <rolename>-server, <rolename>-client, <rolename>-slave и подобное, если используется сложная структура сервиса из нескольких Ansible-ролей.
 - Fix Ubuntu Python - отдельная мидироль, которая устранил несовместимость Ansible 2.7 и Ubuntu 16/18

Пример nfs-server/nfs-server.yml

```

---
- name: Fix Ubuntu Python
  hosts:
    - all
  become: yes
  become_method: sudo
  gather_facts: no
  tasks:
    - name: raw | Install Python2 | Ubuntu
      raw: test -e /usr/bin/python || (apt -y update && apt install -y python-minimal)

- name: NFS Server
  hosts:
    - nfs-server
  become: yes
  become_method: sudo
  roles:
    - nfs-server
    
```

Поддержка разных сред

НУЖНО

- Иметь базу для выполнения одной роли на разных, популярных ОС. Выбор ОС осуществляется в модуле 'include_tasks' с подстановкой названия (опционально, с версией) ОС в имя файла. При этом в каждом "включённом" файле выполняются задачи под свою ОС

Пример skeleton/tasks/main.yml

```

# Exec tasks Install
- block:
  - name: 'Any | Running the tasks Install'
    include_tasks: '{{ item }}'
  with_items:
    - 'install-{{ ansible_distribution | lower }}.yml'
  environment: '{{ property_env }}'
    
```

```

tasks
! configure.yml
! install-centos.yml
! install-oraclelinux.yml
! install-ubuntu.yml
! main.yml
! security-centos.yml
! security-oraclelinux.yml
! security-ubuntu.yml
    
```

- Модуль 'include_tasks' необходимо оборачивать в модуль 'block'. В модуль 'block' добавлять параметры 'environment' и 'tags'
- причина - bar, не передающая переменные из 'environment' во внутрь модуля 'include_tasks' Ссылка

Пример skeleton/tasks/configure.yml

```

- block:
  - name: 'Any | Running the tasks Configure'
    include_tasks: '{{ item }}'
  with_items:
    - 'configure.yml'
  environment: '{{ property_env }}'
  tags:
    - conf
    
```

- Каждый этап помечаться одним или несколькими тегами. Это должно позволить выполнить только часть роли (например, запустить тег "security" и изменить параметры Firewalld и/или SELinux). По названию тега должно быть понятно, какой этап он выполняет.
- Иметься название (name) в следующем формате: <OS> | <Short description> (если точная ОС неизвестна, пишется слово "Any")
- причина - полная обратная связь при выполнении роли (видно, на какой ОС и на какой задаче возникли проблемы)

```

TASK [elasticsearch : Any | Copy jvm.options] 13:15:00
*****
task path: /var/lib/awx/projects/_80_elasticsearch_dev/tasks/configure.yml:52
fatal: [8700ngenie-inside-demo-exp02.ngenie.mtsit.com]: FAILED! => [{"changed": false, "msg": "Could not find or access 'jvm.options-6.8.j2'\nSearcher did not find the file to copy to the remote. If you are using a module and expect the file to exist on the remote, see the remote_src option"}]
    
```

- Использовать специально написанные под задачу Ansible-модули, а модули 'shell' и/или 'command' использовать только в случае, если нет других вариантов;
- Все использования 'shell' и/или 'command' должны помечаться комментарием, почему использовался именно данный модуль
- причина - проце проводить Code Review и другие люди не будут тратить время на попытку перевести shell/command на другой модуль

Ansible: Тракторный режим

Какие сложности возникли при переходе на него

Высокий порог входа

| | |
|------------------------------|-----------------------|
| 🔖 A ansible | ★ 0 🗝 1 month ago |
| 🔖 A awx | ★ 0 🗝 1 month ago |
| 🔖 P prometheus-node-exporter | ★ 0 🗝 2 months ago |
| 🔖 O openjdk | ★ 0 🗝 2 months ago |
| 🔖 J jackrabbit | ★ 0 🗝 3 months ago |
| 🔖 E etcd | ★ 0 🗝 4 months ago |
| 🔖 H harbor | ★ 0 🗝 5 months ago |
| 🔖 S sonar | ★ 0 🗝 5 months ago |
| 🔖 G gitlab | ★ 0 🗝 5 months ago |

Ansible: Тракторный режим
Какие сложности возникли при переходе на него
Проблема сторожил



Ansible: Тракторный режим

Какие сложности возникли при переходе на него
Проблема сторожил



Ansible: Тракторный режим
Какие сложности возникли при переходе на него
Контроль



Ansible: Тракторный режим

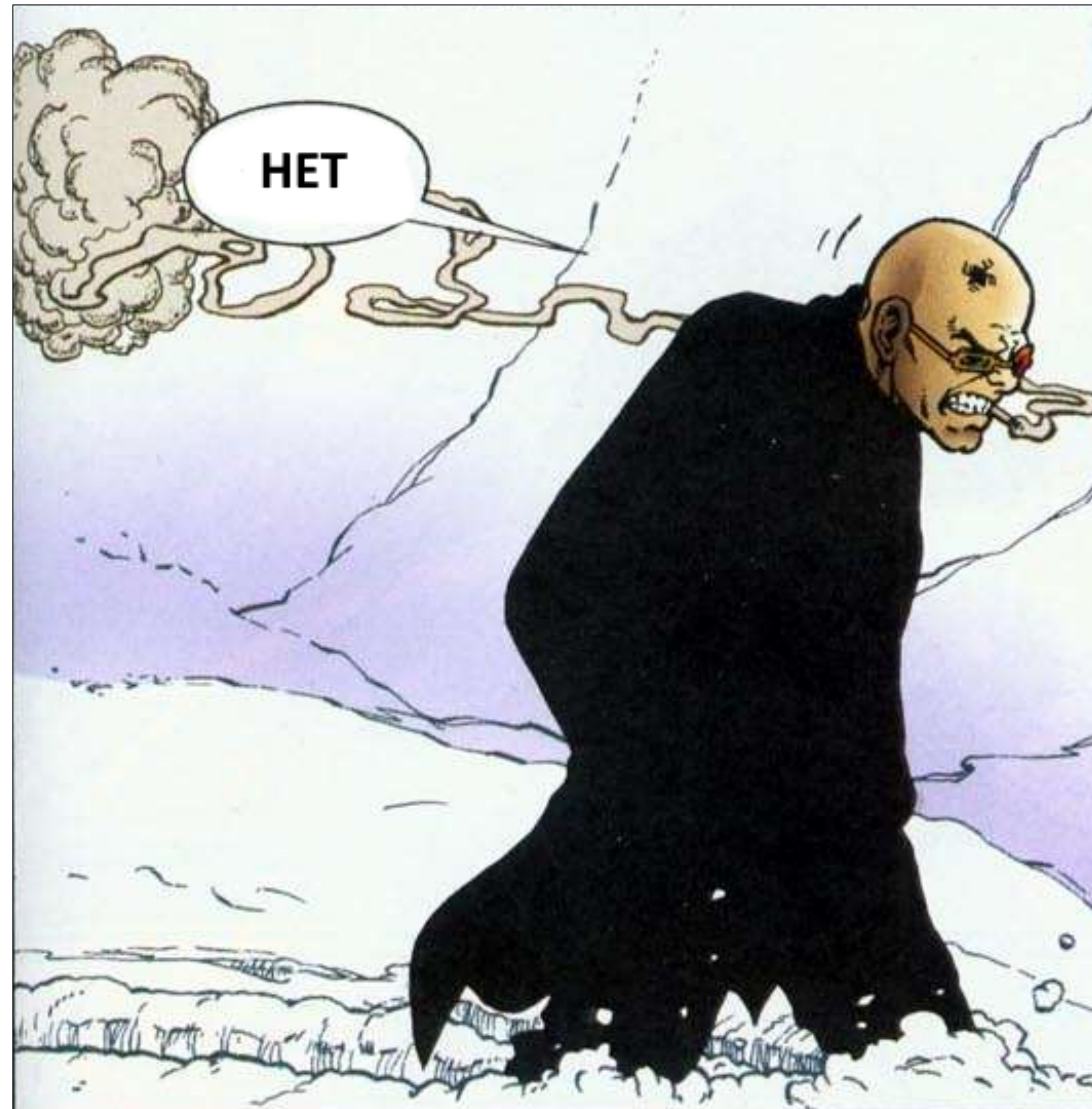
Какие сложности возникли при переходе на него

Контроль



```
main.yml 289 Bytes
1 ---
2 - name: change password
3   user:
4     name: "{{ ch_main_user }}"
5     update_password: always
6     password: "{{ main_user_password }}"
7
8
9 - name: change password
10  user:
11    name: "{{ ch_user_name }}"
12    update_password: always
13    password: "{{ ch_user_password }}"
```


Ansible: Тракторный режим
Это идеал?



Ansible: Тракторный режим

Особые благодарности:

- ШАТАЕВ Илья (🚜)
- ПЕТРОВ Алексей
- МАРЧЕНКО Михаил



НИФАНИН Константин
МТС ИТ



ksnifani@mts.ru