

ОСНОВЫ Ansible

 [linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)

 [facebook.com/redhatinc](https://www.facebook.com/redhatinc)

 [youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)

 twitter.com/RedHat

Alexey Ivanov
Solution Architect
24.03.2020



Автоматизация - это одно из направлений научно-технического прогресса, использующее саморегулирующие технические средства и математические методы с целью освобождения человека от участия в процессах получения, преобразования, передачи и использования энергии, материалов, изделий или информации, либо существенного уменьшения степени этого участия или трудоёмкости выполняемых операций.

Update status

Updates are available.

- Security Intelligence Update for Windows Defender Antivirus - KB2267602 (Version 1.309.1206.0).
- Windows Malicious Software Removal Tool x64 - February 2020 (KB890830).
- 2020-02 Cumulative Update for Windows Server 2016 for x64-based Systems (KB4537764).

Updates are ready to install

Install now

```
[root@util ~]# virt-install --name=linuxconfig-vm \
> --vcpus=1 \
> --memory=1024 \
> --cdrom=/tmp/debian-9.0.0-amd64-netinst.iso \
> --disk size=5 \
> --os-variant=debian8
```

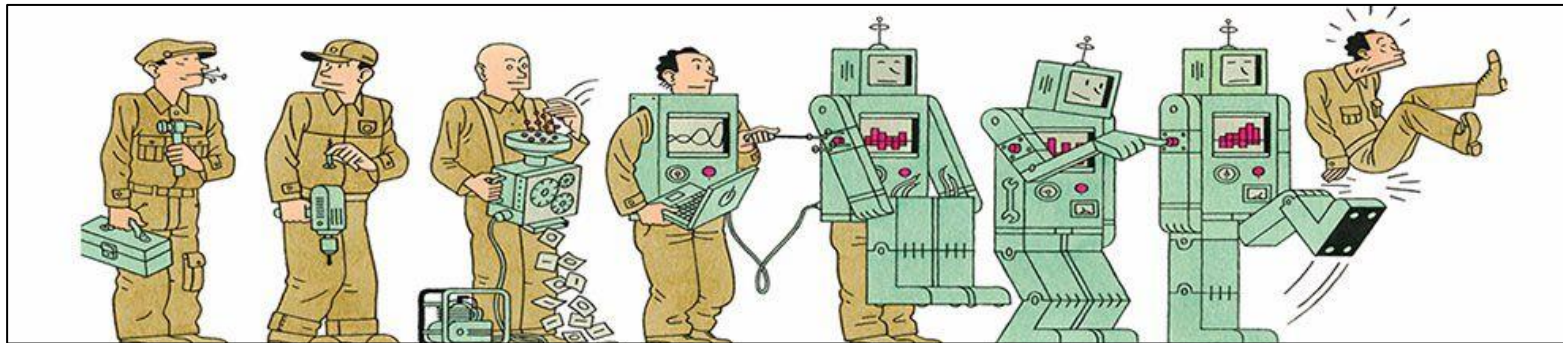
```
enable
vlan database
vlan 2 name sales
vlan 3 name r&d
exit

configure terminal
interface vlan 1
ip address 192.168.1.10 255.255.255.0
no shutdown
ip default-gateway 192.168.1.1

interface fastethernet1/0
switchport mode trunk
interface range fastethernet1/1 - 2
switchport access vlan 2
interface range fastethernet1/3 - 4
switchport access vlan 3
exit
exit

show vlan-switch brief
show interface trunk
```

- Меньше ресурсов на рутинные задачи
- Снижение вероятности ошибки при выполнении рутинных задач
- Передача знаний и опыта
- Стандартизация решения задач
- Стандартизация инфраструктуры



```
$ScriptName = '[CUSTOM] RAM usage exceeds 80%';
$EventID = 1234;
$StartTime = Get-Date;
$whoami = whoami;
$ThresholdBreached = 0;
$Return = '';
$Top10 = @();
$momapi = new-object -comObject "MOM.ScriptAPI";
$PropertyBag = $momapi.CreatePropertyBag();
#Starting
$momapi.LogScriptEvent($ScriptName,$EventID,0,"`n Script is starting. `n Running as ($whoami).")
#Calculating
$Error.Clear();
$os = Get-CimInstance Win32_OperatingSystem;
$RAMAllocated = 100 - [math]::Round(($os.FreePhysicalMemory/$os.TotalVisibleMemorySize)*100,2);
if($RAMAllocated -gt 80){
    $ThresholdBreached = 1;
    $Top10 = Get-Process | sort WS -Descending | select ProcessName, @{n='RAM';e=[math]::Round($_.WS/1MB,0)} -First 10;
    foreach($item in $Top10){
        $Return += "$($item.ProcessName) - $($item.RAM) MB RAM`n";
    }
}
if($Error){
    $momapi.LogScriptEvent($ScriptName,$EventID,1,"`n FATAL ERROR: Unable to load collect performance data. `n Terminating script. `n Error is: ($Error).");
    EXIT;
}
#Filling bag
if(-not$ThresholdBreached){
    $PropertyBag.AddValue("State","Success");
}else{
    $PropertyBag.AddValue("State","Error");
    $PropertyBag.AddValue("Top10",$Return);
}
$EndTime = Get-Date;
$ScriptTime = ($EndTime - $StartTime).TotalSeconds;
$momapi.LogScriptEvent($ScriptName,$EventID,0,"`n Script Completed. `n ThresholdBreached: $ThresholdBreached `n Top 10 processes using RAM: `n $Return `n Script Runtime: ($ScriptTime) seconds.");
return($PropertyBag);
```

```
STORAGE_NAME=$1
STORAGE_KEY=$2
FILESHARE_NAME=$3
MOUNT_POINT=$4

if [ "$#" -ne 4 ]; then
    echo ""
    echo "ERROR:Incorrect number of arguments"
    echo "Usage:"
    echo "sh mount_azure_fileshare.sh storage_name storage_key fileshare_name mount_point"
    echo ""
    exit 1
fi

sudo mkdir -p $MOUNT_POINT
sudo mount -t cifs //$STORAGE_NAME.file.core.windows.net/$FILESHARE_NAME $MOUNT_POINT -o
vers=3.0,username=$STORAGE_NAME,password=$STORAGE_KEY,dir_mode=0777,file_mode=0777

read -r -p "Do you want to Persist the fileshare mount through reboots? [y/n] " RESP
RESP=${RESP,,}      # tolower
if [[ $RESP =~ ^(yes|y)$ ]]
then
    sudo echo "
#####
#Line automatically added by the script: mount_azure_fileshare.sh
//$STORAGE_NAME.file.core.windows.net/$FILESHARE_NAME $MOUNT_POINT cifs
vers=3.0,username=$STORAGE_NAME,password=$STORAGE_KEY,dir_mode=0777,file_mode=0777
" >> /etc/fstab
fi
echo "Fileshare configured"
```



```
---  
-  
  hosts: webservers  
  remote_user: root  
  tasks:  
    -  
      name: ensure apache is at the latest version  
      yum:  
        name: httpd  
        state: latest  
    -  
      name: ensure apache is running  
      service:  
        name: httpd  
        state: started  
...  

```



Просто

- Простой язык на основе YAML
- Не требует навыков программирования
- Последовательное выполнение задач
- Сценарий прост и понятен для других команд



Функционально

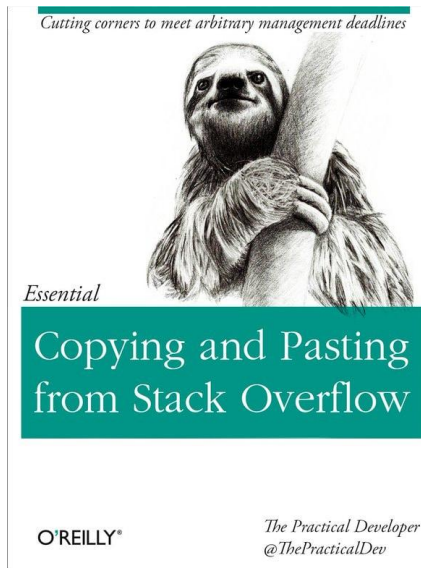
- Установка приложений
- Управление изменениями
- Автоматизация рабочих процессов
- Настройка активного сетевого оборудования



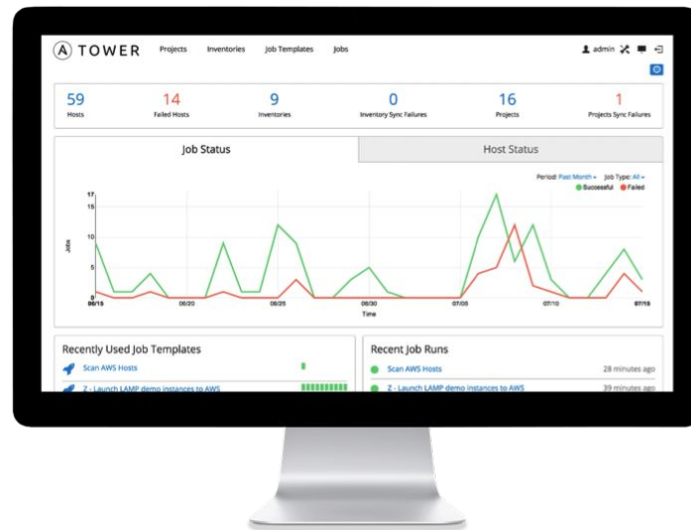
Безопасно

- Не требует агента на клиенте
- OpenSSH & WinRM в качестве транспорта
- Эскалация привилегий
- Проверка на соответствие стандартам компании

- Наиболее популярное средство автоматизации на GitHub
- 13,000+ участников и 4,000+ форков
- 2,000+ контрибуторов
- Ansible Galaxy: 18,000+ подписчиков
- 250,000+ скачиваний в месяц



- **Язык** для автоматизации ИТ-инфраструктуры и приложений с помощью сценариев
- **Движок** Ansible для запуска и работы сценариев
- **Ansible Tower** предоставляет:
 - возможность автоматизировать рабочие процессы и определить взаимосвязь между сценариями
 - ролевая модель доступа для запуска сценариев
 - централизованное выполнение задач с регистрацией событий и их статусом
 - графический интерфейс и REST API
 - Логирование действий пользователей



Ansible - проект сообщества, спонсируемый **Red Hat**. Это простой язык автоматизации, который описывает желаемое состояние инфраструктуры с помощью **Ansible Playbooks**.

Ansible Tower - это фреймворк корпоративного применения **Ansible** для контроля, управления и обеспечения безопасности автоматизации с функциями **UI** и **RESTful API**.



CLOUD

AWS
Azure
Digital Ocean
Google
OpenStack
Rackspace
+more

OPERATING SYSTEMS

RHEL and Linux
UNIX
Windows
+more

VIRT & CONTAINER

Docker
VMware
RHV
OpenStack
OpenShift
+more

STORAGE

NetApp
Red Hat Storage
Infinidat
+more

WINDOWS

ACLs
Files
Packages
IIS
Regedits
Shares
Services
Configs
Users
Domains
+more

NETWORK

Arista
A10
Cumulus
Bigswitch
Cisco
Cumulus
Dell
F5
Juniper
Palo Alto
OpenSwitch
+more

DEVOPS

Jira
GitHub
Vagrant
Jenkins
Bamboo
Atlassian
Subversion
Slack
Hipchat
+more

MONITORING

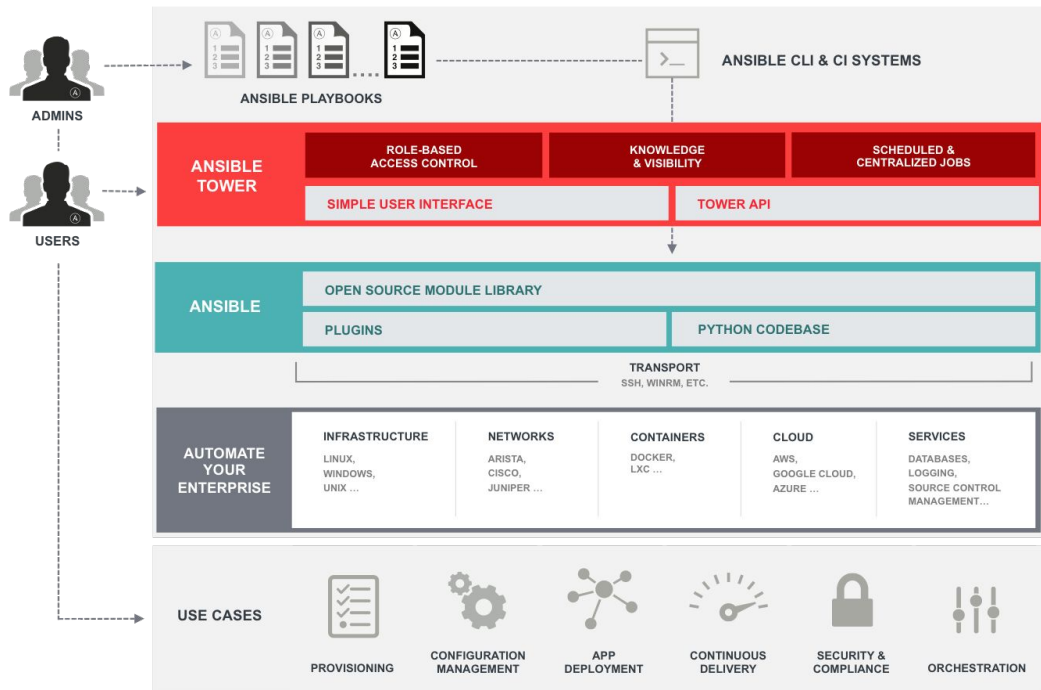
Dynatrace
Airbrake
BigPanda
Datadog
LogicMonitor
Nagios
New Relic
PagerDuty
Sensu
StackDriver
Zabbix
+more

IDC: Business value of Red Hat Ansible Automation

TABLE 2

Red Hat Ansible Automation Usage									
Ansible Usage	Ansible Customer Respondent								
	1	2	3	4	5	6	7	8	9
Automating DevOps workflows	✓	✓	✓	✓	✓	✓	✓	✓	✓
Standardizing infrastructure config/provisioning	✓	✓		✓	✓	✓	✓	✓	✓
Run complex IT environments	✓	✓	✓		✓	✓	✓	✓	✓
Deploy private/hybrid/multicloud	✓	✓	✓		✓	✓	✓	✓	
Application modernization	✓			✓	✓	✓	✓	✓	✓
Bring different IT groups closer together	✓				✓	✓			✓

Source: IDC, 2019





Базовые сущности

1. Control node - any machine with Ansible installed
2. Playbook - ordered lists of tasks
3. Play - combination of hosts and tasks
4. Task - the units of action in Ansible (call to the Ansible module)
5. Inventory - file, that defines the hosts and groups of hosts upon which commands, modules, and tasks in a playbook operate
6. Host - The network devices (and/or servers) you manage with Ansible
7. Module - reusable, standalone script that Ansible runs on your behalf, either locally or remotely

Playbook

```
---  
  
- hosts: web  
  
  tasks:  
  
    - name: install latest apache  
  
      yum:  
  
        name: httpd  
  
        state: latest  
  
  
- hosts: all  
  
  tasks:  
  
    - name: update packages  
  
      yum:  
  
        name: '*'  
  
        state: latest
```

Ссылки по теме:

https://docs.ansible.com/ansible/latest/user_guide/basic_concepts.html

- hosts: web

tasks:

- name: install latest apache

yum:

name: httpd

state: latest

- hosts: all

tasks:

- name: update packages

yum:

name: '*'

state: latest

} Play

```
---  
  
- hosts: web  
  
  tasks:  
    - name: install latest apache  
      yum:  
        name: httpd  
        state: latest  
  
- hosts: all  
  
  tasks:  
    - name: update packages  
      yum:  
        name: '*'  
        state: latest
```

} Task

```
---  
  
- hosts: web  
  
  tasks:  
  
    - name: install latest apache  
  
      yum:  
  
        name: httpd  
  
        state: latest  
  
  
- hosts: all  
  
  tasks:  
  
    - name: update packages  
  
      yum: ← module  
  
        name: '*'  
  
        state: latest
```

```
mail.example.com
```

```
[webservers]
```

```
foo.example.com
```

```
bar.example.com
```

```
[dbservers]
```

```
one.example.com
```

```
two.example.com
```

```
three.example.com
```

```
all:
```

```
  hosts:
```

```
    mail.example.com:
```

```
  children:
```

```
    webservers:
```

```
      hosts:
```

```
        foo.example.com:
```

```
        bar.example.com:
```

```
    dbservers:
```

```
      hosts:
```

```
        one.example.com:
```

```
        two.example.com:
```

```
        three.example.com:
```

Ссылки по теме:

https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html#intro-inventory



YAML

Список

- Apple
- Orange
- Strawberry
- Mango

Начало файла: ---

Конец файла: ...

Комментарий: #

Словарь

Car: Maybach

Owner: John

Color: Red

Ссылки по теме:

<https://habr.com/ru/company/redhatrussia/blog/462125/>

https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html#yaml-syntax

```

---
-
  - House
  - Car
  - Dog
-
  - House
  - Wife
  - Cat
...

```

```

---
- me: Alex
  house: moscow
  car: volga
  dog: Rex
- you: Not Alex
  house: spb
  wife: young
  cat: furry
...

```

```

---
-
  me: Alex
  house: moscow
  car: volga
  dog: Rex
-
  you: Not Alex
  house: spb
  wife: young
  cat: furry
...

```

```
---  
  
# What do we have?  
mine:  
  - house  
  - car  
  - dog  
yours:  
  - house  
  - wife  
  - cat  
  
...
```

```
---  
  
# What do we have in detail?  
mine:  
  property: house  
  item: car  
  pet: dog  
yours:  
  property: house  
  companion: wife  
  pet: cat  
  
...
```

```
---  
mine:  
  property:  
    - house  
    - car  
  pet: dog  
yours:  
  - house  
  - cat  
  -  
  company: wife  
...
```



Playbook

1. Список из play
2. В общем виде play это словарь с обязательными хэшами tasks, hosts, а также возможностью указать дополнительные параметры:
 - a. remote_user
 - b. become
 - c. order
 - d. gather_facts
 - e. И другие

Ссылки по теме:

https://docs.ansible.com/ansible/latest/user_guide/playbooks_intro.html

```
---  
-  
  hosts: webservers  
  remote_user: root  
  tasks:  
    -  
      name: ensure apache is at the latest version  
      yum:  
        name: httpd  
        state: latest  
    -  
      name: ensure apache is running  
      service:  
        name: httpd  
        state: started  
...  
...
```

-

```
hosts: webserver ← Где выполним
remote_user: root
tasks:
```

-

```
  name: ensure apache is at the latest version
  yum:
    name: httpd
    state: latest
```

-

```
  name: ensure apache is running
  service:
    name: httpd
    state: started
```

...

Ссылки по теме:

https://docs.ansible.com/ansible/latest/user_guide/intro_patterns.html#intro-patterns

```

---
-
  hosts: webservers
  remote_user: root  ← От имени кого выполним
  tasks:
    -
      name: ensure apache is at the latest version
      yum:
        name: httpd
        state: latest
    -
      name: ensure apache is running
      service:
        name: httpd
        state: started
  ...

```

ЧТО ВЫПОЛНИМ

```
---  
-  
  hosts: webserver  
  remote_user: root  
  tasks:  
    -  
      name: ensure apache is at the latest version  
      yum:  
        name: httpd  
        state: latest  
    -  
      name: ensure apache is running  
      service:  
        name: httpd  
        state: started  
...  

```

```

---
-
  hosts: webserver
  remote_user: root
  tasks:
    -
      name: ensure apache is at the latest version
      yum:
        name: httpd
        state: latest
    -
      name: ensure apache is running
      service:
        name: httpd
        state: started
    ...

```

Первый task {

{ Второй task

```

---
-
  hosts: webserver
  remote_user: root
  tasks:
    -
      name: ensure apache is at the latest version
      yum:
        name: httpd
        state: latest
    -
      name: ensure apache is running
      service:
        name: httpd
        state: started
  ...

```

Какой модуль → yum:

Комментарий ↓

Параметры модуля }



Привилегии

1. Срабатывает после подключения к хосту
2. Использует su, sudo и другие способы эскалации привилегий
3. Все директивы эскалации привилегий независимы

Краткий список:

- become
- become_user
- become_method

Ссылки по теме:

https://docs.ansible.com/ansible/latest/user_guide/become.html

```

---
-
  hosts: web
  become: true
  become_user: root
  become_method: su
  tasks:
    -
      name: install latest apache
      yum:
        name: httpd
        state: latest
    ...

```

```
---  
-  
  hosts: web  
  gather_facts: false  
  remote_user: remote_user  
  tasks:  
    -  
      name: install latest apache  
      yum:  
        name: httpd  
        state: latest  
    -  
      name: restart apache  
      become: yes  
      become_method: sudo  
      service:  
        name: httpd  
        state: restarted  
...  

```



Модули

- firewallld:
 - service: https
 - permanent: yes
 - state: enabled
- firewallld:
 - port: 8081/tcp
 - permanent: yes
 - state: disabled
- firewallld:
 - port: 161-162/udp
 - permanent: yes
 - state: enabled
- firewallld:
 - zone: dmz
 - service: http
 - permanent: yes
 - state: enabled

Ссылки по теме:

https://docs.ansible.com/ansible/latest/user_guide/modules.html



Подключение Работа в консоли

- По умолчанию используется OpenSSH
- Можно использовать как пароли, так и ключи
- По умолчанию Ansible использует `known_hosts` как источник доверенных ключей
- Подключение к Windows хостам реализуется через `winrm`

Ссылки по теме:

https://docs.ansible.com/ansible/latest/user_guide/connection_details.html

https://docs.ansible.com/ansible/latest/user_guide/windows_winrm.html

Вызов встроенного модуля/произвольной команды из командной строки

```
$ ansible host-pattern -m module [-a 'module arguments'] [-i inventory] [options]
```

```
[root@podman ~]# ansible ans1.test.local -m ping -i ./inventory -u remote_user -k
SSH password:
ans1.test.local | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
[root@podman ~]#
```

Ссылки по теме:

https://docs.ansible.com/ansible/latest/user_guide/intro_adhoc.html

Вызов справки по модулю из командной строки

```
$ ansible-doc ping
```

```
<...>
```

```
AUTHOR: Ansible Core Team, Michael DeHaan
```

```
  METADATA:
```

```
    status:
```

```
      - stableinterface
```

```
    supported_by: core
```

Ссылки по теме:

<https://docs.ansible.com/ansible/latest/cli/ansible-doc.html>

Проверка синтаксиса из командной строки

```
[root@podman ~]# ansible-playbook ./test_playbook.yaml -i ./inventory --syntax-check
playbook: ./test_playbook.yaml
[root@podman ~]#
```

```
[root@podman ~]# ansible-playbook ./test_playbook.yaml -i ./inventory --syntax-check
ERROR! Syntax Error while loading YAML.

The error appears to have been in '/root/test_playbook.yaml': line 3, column 9, but may
be elsewhere in the file depending on the exact syntax problem.

The offending line appears to be:

- hosts: all
  tasks:
    ^ here

exception type: <class 'yaml.scanner.ScannerError'>
exception: mapping values are not allowed in this context
in "<unicode string>", line 3, column 9
[root@podman ~]#
```

Ссылки по теме:

<https://docs.ansible.com/ansible/latest/cli/ansible-playbook.html>

Вызов сценариев из командной строки

```
[root@podman ~]# ansible-playbook ./test_playbook.yaml -i ./inventory -u remote_user -k
SSH password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [ans1.test.local]

TASK [install vim] *****
ok: [ans1.test.local]

TASK [Copy file with owner and permissions] *****
changed: [ans1.test.local]

PLAY RECAP *****
ans1.test.local      : ok=3    changed=1    unreachable=0    failed=0
```

Ссылки по теме:

<https://docs.ansible.com/ansible/latest/cli/ansible-playbook.html>



Файлы конфигурации

ansible.cfg может находиться в (ищется в этом же порядке):

- ANSIBLE_CONFIG (environment variable if set)
- ansible.cfg (in the current directory)
- ~/.ansible.cfg (in the home directory)
- /etc/ansible/ansible.cfg

В текущей директории ansible.cfg используется только при условии, что директория недоступна всем для записи

Ссылки по теме:

https://docs.ansible.com/ansible/latest/reference_appendices/config.html#ansible-configuration-settings

<https://docs.ansible.com/ansible/latest/cli/ansible-config.html#ansible-config>

```
[defaults]  
inventory=/root/inventory  
remote_user=remote_user  
ask_pass=true
```

```
[privilege_escalation]  
become=true  
become_method='sudo'  
become_ask_pass=true
```



Переменные

- Переменные могут быть заданы как в inventory, так и в playbook
- Имя переменной может содержать английские буквы верхнего и нижнего регистра, подчёркивания и цифры. Начинаться имя должно с буквы

Синтаксис:

foo:

field1: one

field2: two

Способ обращения:

"{{ var_name }}"

"{{ to_do['install'] }}"

Ссылки по теме:

https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html

https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html#adding-variables-to-inventory

```
[atlanta]
host1 http_port=80 ansible_user=user2
host2 http_port=303 ansible_user=user1

atlanta:
  host1:
    http_port: 80
    ansible_user: user2
  host2:
    http_port: 303
    ansible_user: user1
```

```
[atlanta]
host1
host2

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

atlanta:
  hosts:
    host1:
    host2:
  vars:
    ntp_server: ntp.atlanta.example.com
    proxy: proxy.atlanta.example.com
```

```

---
-
  hosts: web
  vars:
    to_install: 'httpd'
  gather_facts: false
  remote_user: remote_user
  tasks:
    -
      name: install latest apache
      yum:
        name: "{{ to_install }}"
        state: latest
    -
      name: restart apache
      become: yes
      become_method: sudo
      service:
        name: "{{ to_install }}"
        state: restarted
  ...

```

```

---
-
  hosts: web
  vars:
    to_do:
      install: 'httpd'
      restart: 'firewalld'
  gather_facts: false
  remote_user: remote_user
  tasks:
    - name: install latest apache
      yum:
        name: "{{ to_do['install'] }}"
        state: latest
    - name: restart apache
      become: yes
      become_method: sudo
      service:
        name: "{{ to_do['restart'] }}"
        state: stopped
...

```

Перед запуском ваших модулей Ansible собирает данные о хосте. Эти данные он помещает в переменную `ansible_facts`, к которой вы можете обращаться в вашем плейбуке.

Как мы уже знаем, сбор фактов о хосте можно отключать.

Ссылки по теме:

https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html#variables-discovered-from-systems-facts

Выдержка

```
...
"ansible_all_ipv4_addresses": [
    "REDACTED IP ADDRESS"
],
"ansible_all_ipv6_addresses": [
    "REDACTED IPV6 ADDRESS"
],
"ansible_apparmor": {
    "status": "disabled"
},
"ansible_architecture": "x86_64",
"ansible_bios_date": "11/28/2013",
"ansible_bios_version": "4.1.5",
...
"ansible_nodename":
"centos-7-rax-dfw-0003427354",
"ansible_os_family": "RedHat",
...
```

Способ обращения старый

```
"{{ ansible_facts['nodename'] }}"  
"{{ ansible_facts['apparmor']['status']  
}}"
```

Способ обращения новый

```
"{{ ansible_nodename }}"  
"{{ ansible_apparmor['status'] }}"
```



To be continued...