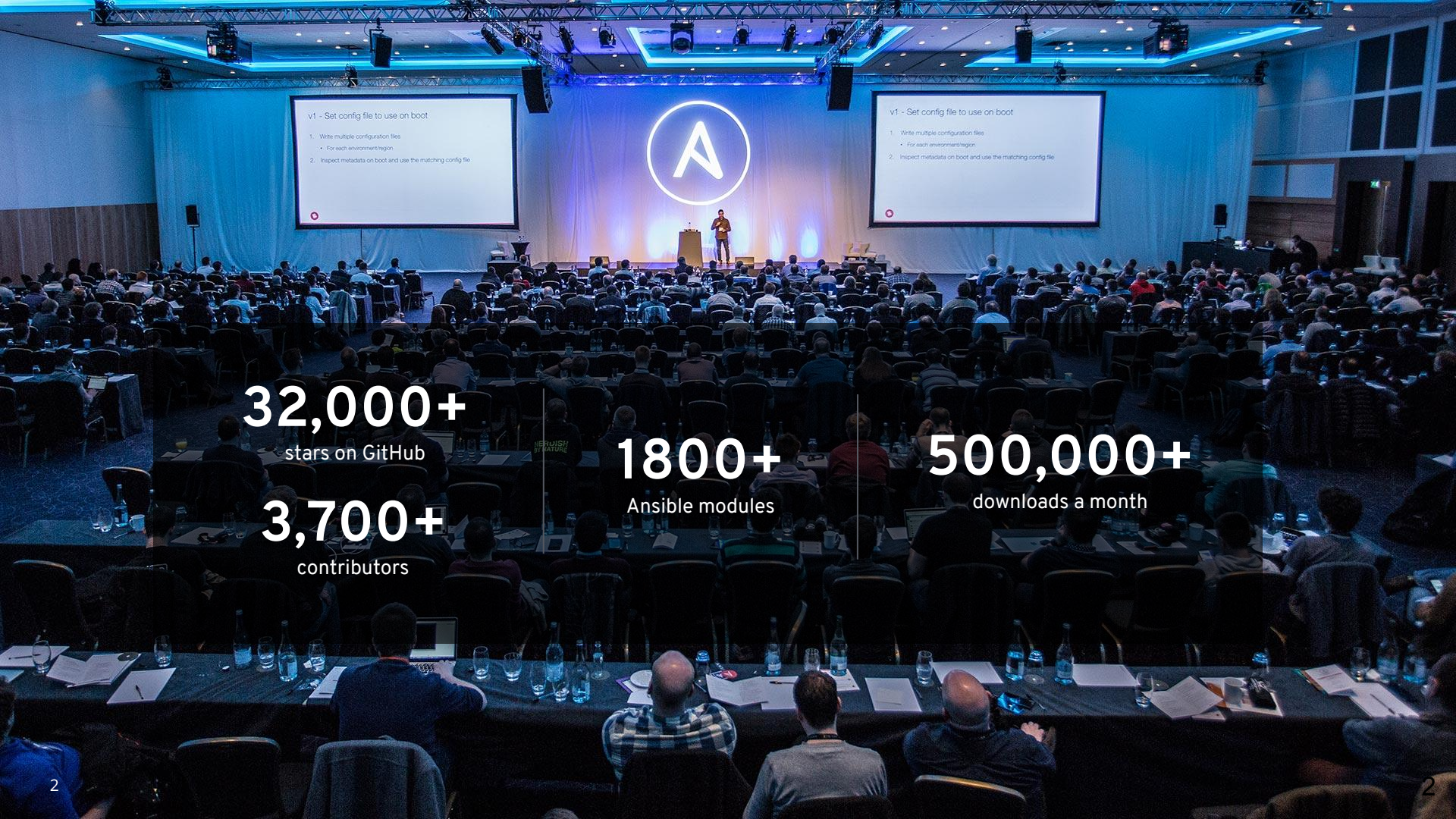




# Зачем мне Ansible, если я работаю в сетевом департаменте

Red Hat Ansible Automation (Networking)



v1 - Set config file to use on boot

1. Write multiple configuration files
  - For each environment/region
2. Inspect metadata on boot and use the matching config file

v1 - Set config file to use on boot

1. Write multiple configuration files
  - For each environment/region
2. Inspect metadata on boot and use the matching config file



**32,000+**

stars on GitHub

**3,700+**

contributors

**1800+**

Ansible modules

**500,000+**

downloads a month

# ОСНОВНЫЕ ПРЕИМУЩЕСТВА ANSIBLE

## ПРОСТОТА В ИСПОЛЬЗОВАНИИ

- Простой язык на основе YAML
- Не требует навыков программирования
- Последовательное выполнение задач
- Автоматизация для всех

**БЫСТРО ПОЛУЧИТЬ  
РЕЗУЛЬТАТ И ПОВЫСИТЬ  
ПРОИЗВОДИТЕЛЬНОСТЬ**

## ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ

- Развертывание приложений
- Управление изменениями
- Автоматизация рабочих процессов
- Оркестрация между различными платформами

**ОРКЕСТРАЦИЯ ПРИЛОЖЕНИЯ  
ИЛИ СЕРВИСА В ТЕЧЕНИЕ  
ЖИЗНЕННОГО ЦИКЛА**

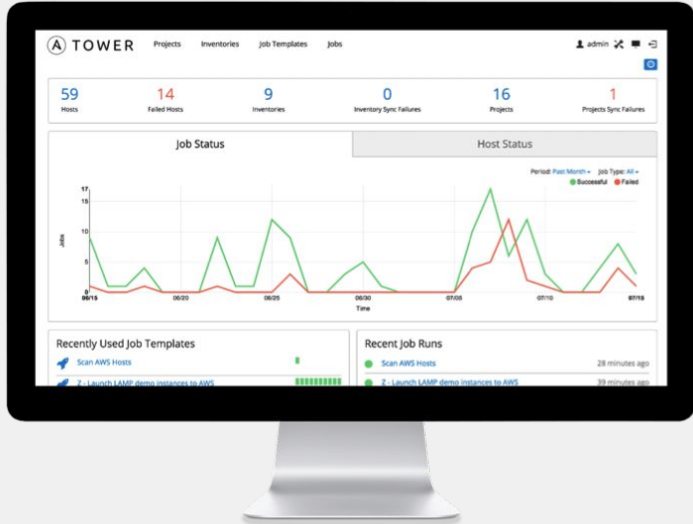
## БЕЗОПАСНОСТЬ

- Не требует агента на клиенте
- Меньше уязвимостей и сложностей с апдейтами
- OpenSSH & WinRM в качестве транспорта
- Эскалация привилегий

**РАБОТАТЬ БОЛЕЕ  
ЭФФЕКТИВНО И БОЛЕЕ  
БЕЗОПАСНО**



# RED HAT ANSIBLE AUTOMATION



- **Ansible** - Язык для автоматизации ИТ-инфраструктуры и приложений с помощью сценариев
- **Ansible Engine** - Движок для запуска и работы сценариев
- **Ansible Tower** - платформа/система управления Ansible-автоматизацией **корпоративного класса**, обеспечивающая:
  - 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

# RED HAT ANSIBLE AUTOMATION



## RED HAT ANSIBLE TOWER

Scale + operationalize your automation

CONTROL

KNOWLEDGE

DELEGATION

## RED HAT ANSIBLE ENGINE

Support for your Ansible automation

SIMPLE

POWERFUL

AGENTLESS

FUELED BY AN INNOVATIVE **OPEN SOURCE** COMMUNITY

# ANSIBLE - PROJECT & PRODUCT

Open Source  
(Communities)



Red Hat Ansible Automation  
(Enterprise)

OPS - IT Managers, "Teams"



Bottom-Up  
Influence

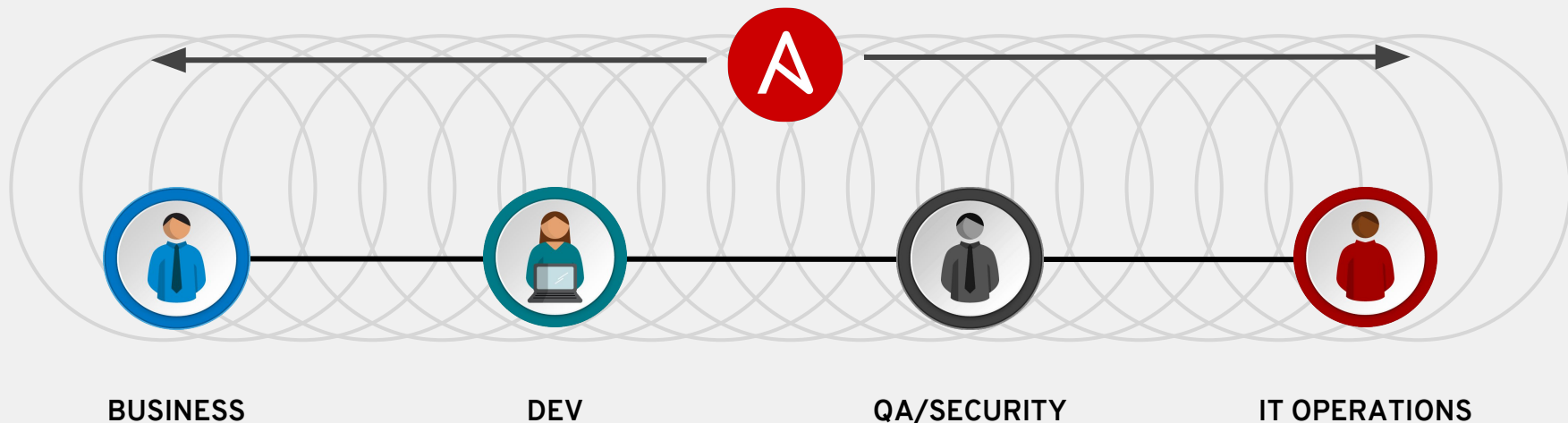
Top-Down  
Strategy

NETOPS - Network Operations



DEV - Playbook Authors, "Individuals"

# ANSIBLE - ЭТО УНИВЕРСАЛЬНЫЙ ЯЗЫК



Ansible is the first **automation language** that can be read and written across IT.

Ansible is the only **automation engine** that can automate the entire application lifecycle and continuous delivery pipeline.

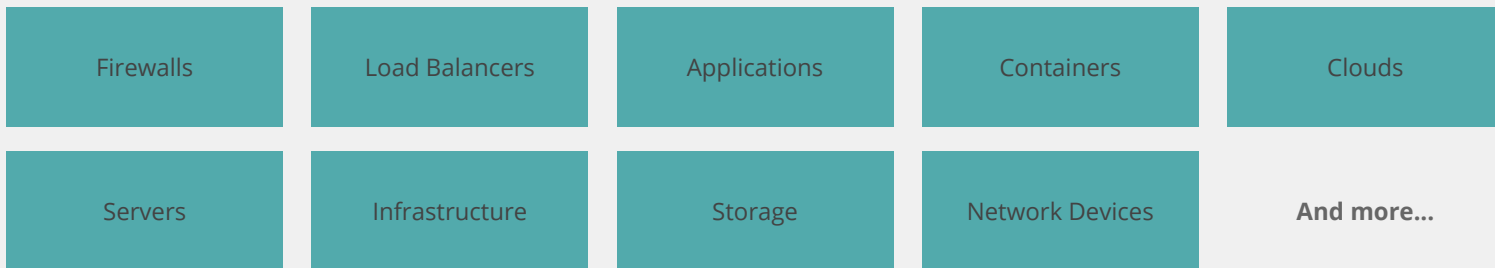
# ЧТО МОЖНО СДЕЛАТЬ, ИСПОЛЬЗУЯ ANSIBLE

Automate the deployment and management of your entire IT footprint.

## Do this...



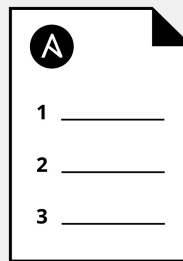
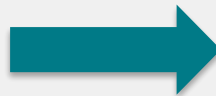
## On these...





# КОНВЕРТИРУЕМ ПРОЦЕДУРЫ В ANSIBLE PLAYBOOKS

1. Create VLAN
2. Add port to VLAN
3. Address Interface

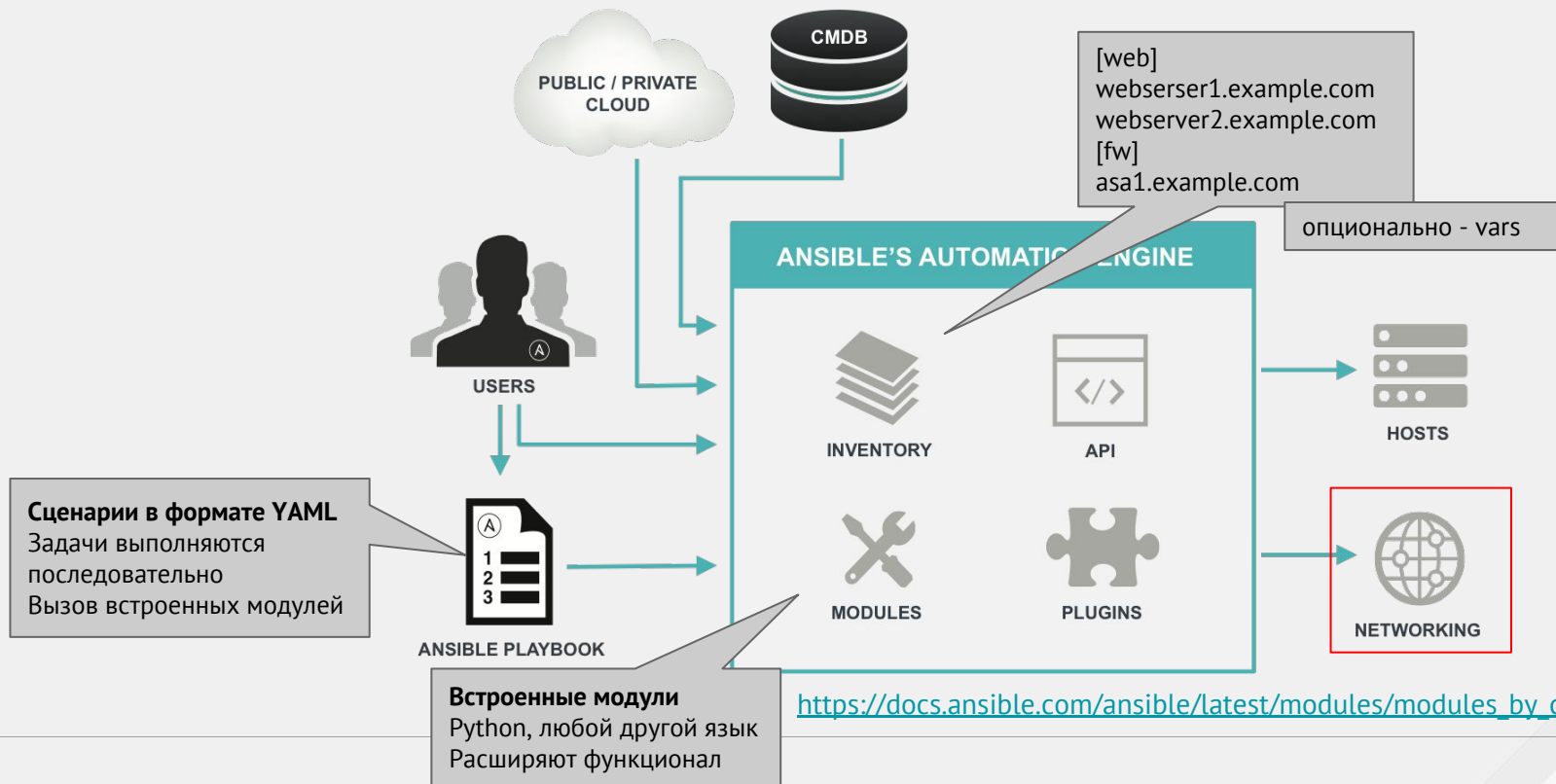


Method of Procedure

Playbook

- Define Intent, Policy, Architecture
- Apply across device type, vendor

# ANSIBLE AUTOMATION ARCHITECTURE



# ANSIBLE PLAYBOOK (СЦЕНАРИЙ)

```
- hosts: network
```

Inventory: The devices to configure

```
vars:
```

```
  site_domain_name: 'example.net'
```

```
  network_name_servers:
```

```
    - 8.8.8.8
```

```
    - 8.8.4.4
```

```
  log_host: 10.2.2.3
```

Variables: The key/value pairs that change from device to device

```
tasks:
```

```
- name: Configure the hostname and domain name
```

```
  net_system:
```

```
    hostname: "{{ inventory_hostname }}"
```

```
    domain_name: "{{ site_domain_name }}"
```

```
    name_servers: "{{ network_name_servers }}"
```

```
- name: configure host logging
```

```
  net_logging:
```

```
    dest: host
```

```
    name: "{{ log_host }}"
```

Tasks: The tasks to perform on those devices

# ANSIBLE MODULE EXAMPLE - 'COMMAND'

- Executes command on device
- Provides output for further processing

```
- hosts: network
gather_facts: no
connection: local
tasks:
  - name: show version
    ios_command:
      commands:
        - show version
      wait_for:
        - result[0] contains
          Version
    register: results

  - set_fact:
    ver: "{{ results.stdout[0]|regex_search('Version ([0-9.]+)', '\\1') }}"

  - debug: var=ver
```

```
PLAY [network]
*****
TASK [show version and show interfaces]
*****
ok: [rtr1]

TASK [set_fact]
*****
ok: [rtr1]

TASK [debug]
*****
ok: [rtr1] => {
  "ver": [
    "16.06.01"
  ]
}
PLAY RECAP
*****
rtr1 : ok=3 changed=0 unreachable=0 failed=0
```

# ANSIBLE MODULE EXAMPLE - 'CONFIG'

```
First Run:
PLAY [network]
*****
TASK [configure hostname]
*****
changed: [rtr1]

PLAY RECAP
*****
rtr1                                : ok=1    changed=1    unreachable=0    failed=0

Second Run:
PLAY [network]
*****
TASK [configure hostname]
*****
ok: [rtr1]

PLAY RECAP
*****
rtr1                                : ok=1    changed=0    unreachable=0    failed=0
```

```
- hosts: network
gather_facts: no
connection: local
tasks:
  - name: configure hostname
    ios_config:
      lines:
        - "hostname {{ inventory_hostname }}"
```

# ANSIBLE MODULE EXAMPLE - 'FACTS'

```
- hosts: network
  connection: local
  gather_facts: False
  tasks:
    - name: Get facts
      ios_facts:
        gather_subset: all

    - debug: msg="Serial Number is {{ ansible_net_serialnum }}"
```

```
PLAY [network]
*****

TASK [Get facts]
*****
ok: [rtr1]

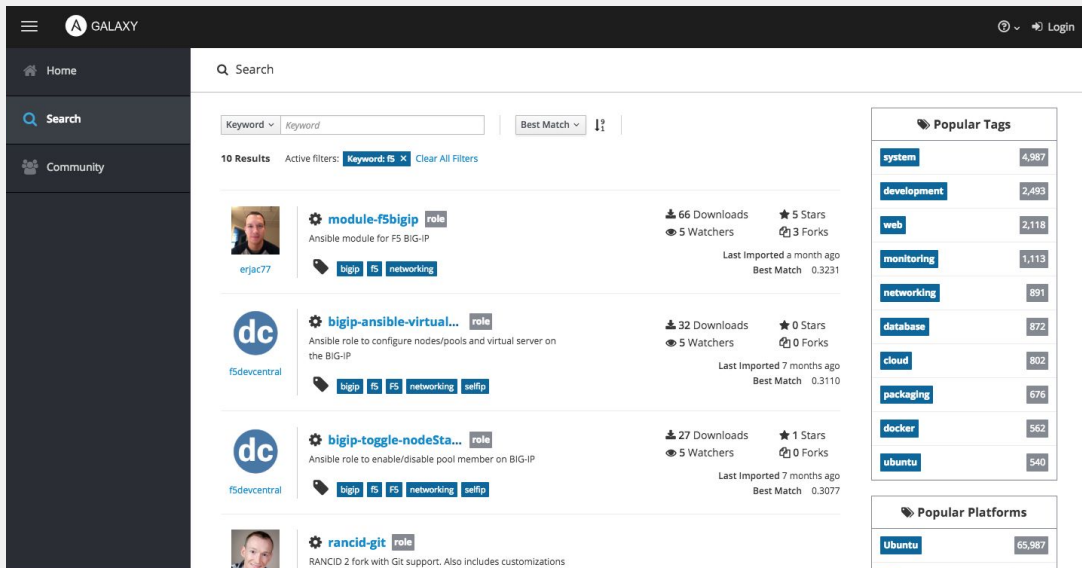
TASK [debug]
*****
ok: [rtr1] => {
  "msg": "Serial Number is 9G20X4MKLVP"
}

PLAY RECAP
*****
rtr1                    : ok=2    changed=0    unreachable=0    failed=0
```



# ANSIBLE GALAXY. РОЛИ

Роли позволяют объединить **сценарии** совместно с **задачами**, переменными, шаблонами



The screenshot shows the Ansible Galaxy search interface. The search bar contains the keyword 'f5'. The results list several roles:

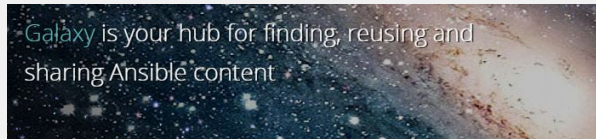
- module-f5bigip** (role) by erja77: 66 Downloads, 5 Stars, 5 Watchers, 3 Forks. Last imported a month ago. Best Match: 0.3231. Tags: bigip, f5, networking.
- bigip-ansible-virtual...** (role) by f5decentral: 32 Downloads, 0 Stars, 5 Watchers, 0 Forks. Last imported 7 months ago. Best Match: 0.3110. Tags: bigip, f5, networking, selfip.
- bigip-toggle-nodeSta...** (role) by f5decentral: 27 Downloads, 1 Star, 5 Watchers, 0 Forks. Last imported 7 months ago. Best Match: 0.3077. Tags: bigip, f5, networking, selfip.
- rancid-git** (role) by RANCID 2 fork with Git support. Also includes customizations.

On the right side, there are two sections:

- Popular Tags:**

system	4,987
development	2,493
web	2,118
monitoring	1,113
networking	891
database	872
cloud	802
packaging	676
docker	562
ubuntu	540
- Popular Platforms:**

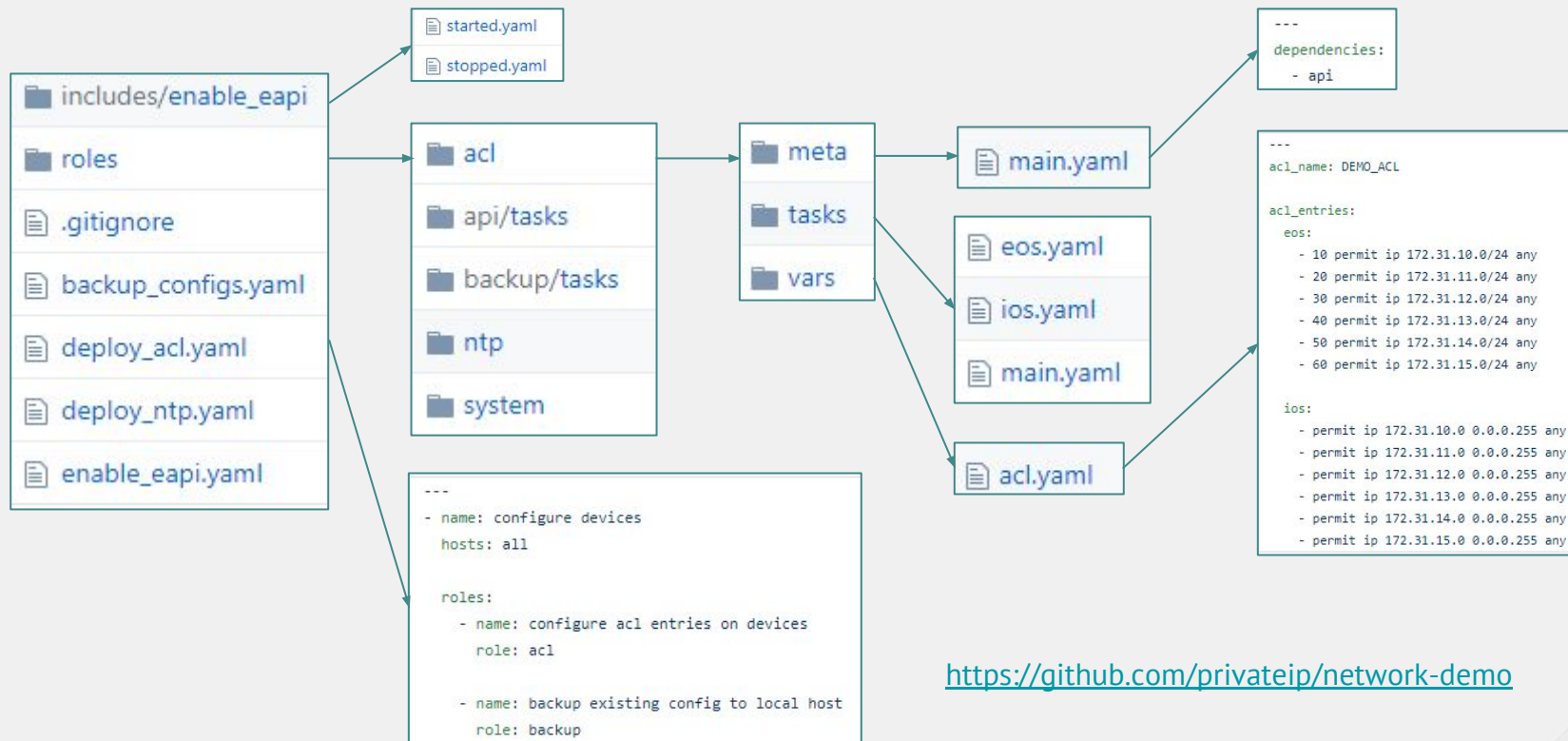
Ubuntu	65,987
--------	--------



<https://galaxy.ansible.com/>

Ansible Galaxy - репозиторий, содержащий более 15000 ролей, открытых и готовых к использованию

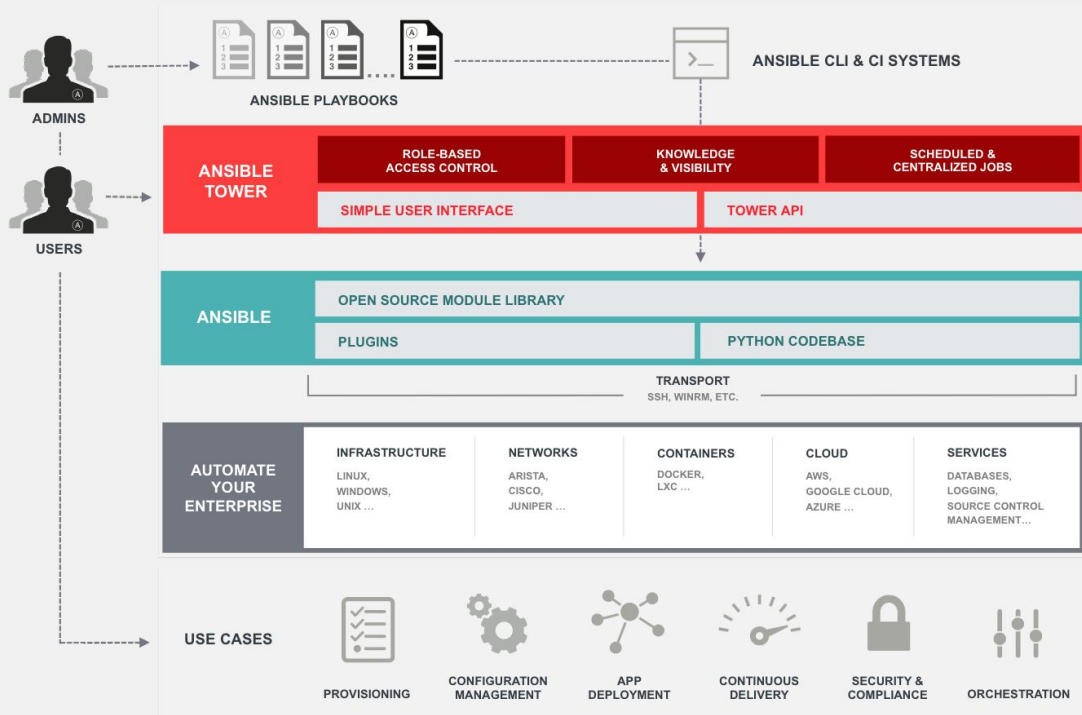
# ПРИМЕР - РОЛИ ANSIBLE ДЛЯ ACL



<https://github.com/privateip/network-demo>

RED HAT ANSIBLE TOWER

# ANSIBLE TOWER ARCHITECTURE



<https://www.ansible.com/products/tower>



# ANSIBLE TOWER - КОНТРОЛЬ И RBAC

The screenshot displays the Ansible Tower interface. At the top, there are navigation tabs: TOWER, PROJECTS, INVENTORIES, TEMPLATES, and JOBS. Below this, the breadcrumb path is 'JOBS / 248 - APPLY STANDARD CONFIGURATION'. The main content area is titled 'REMOVE VMWARE HOST' and shows a progress bar with a green indicator and a '3' in a green circle. Below the progress bar, there are statistics: PLAYS 1, TASKS 8, HOSTS 15, and ELAPSED 00:00:05. A search bar is present. The main execution log shows a list of tasks with their status and duration. The tasks are: 1. PLAY [Remove VMWare Host] (00:00:05), 2. GATHERING FACTS (00:00:01), 16. TASK: [ansiblelicense | install required packages via yum] (00:00:01), 20. TASK: [ansiblelicense | update setuputils] (00:00:01), 29. TASK: [ansiblelicense | update pip] (00:00:01), 36. TASK: [ansiblelicense | create unprivileged user for ansiblelicense] (00:00:01), 40. TASK: [ansiblelicense | configure ansiblelicense directory permissions] (00:00:01), 44. TASK: [ansiblelicense | enable maintenance page] (00:00:01), 66. TASK: [ansiblelicense | check ssh connection to github] (00:00:01), and 69. PLAY RECAP. On the left, a 'DETAILS' sidebar shows job status as 'Successful', started at 1:00:07 AM, finished at 1:00:12 AM, template 'Update License Server', job type 'Playbook Run', launched by 'User', inventory 'License Server', project 'License', revision '0000000', playbook 'store.yml', credential 'License Server Deployment', limit 'Store', verbosity 'Update License Server', and extra variables including 'expire\_time: 1453164676' and 'vmware\_host: cent7issue'.

The screenshot shows the 'CLOUDFORMS' interface with two tabs: 'DETAILS' and 'PERMISSIONS'. The 'DETAILS' tab is active. It displays fields for '\*NAME' (CloudForms), 'DESCRIPTION' (Inventory Script), and 'ORGANIZATION' (Bit63). Below this, there is a '\*TYPE' dropdown menu set to 'Red Hat CloudForms'. Under 'TYPE DETAILS', there are fields for '\*CLOUDFORMS HOST' (https://cloudforms07.bit63.net), '\*USERNAME' (admin), and '\*PASSWORD' (SHOW). The interface is clean and modern with a light blue and white color scheme.

The screenshot shows the 'Help Desk' interface with a sub-tab 'Add Permissions'. A message says 'Please select resources from the lists below.' Below this, there are several filter buttons: 'JOB TEMPLATES', 'WORKFLOW TEMPLATES', 'PROJECTS', 'INVENTORIES', and 'CREDENTIALS'. A search bar is present. Below the search bar, there is a table with columns 'NAME' and 'DESCRIPTION'. A single entry is visible: 'Demo Job Template'.

- Регистрация всех действия в БД
- Статус выполнения сценариев (Jobs)
- История выполненных задач
- Управление на уровне организаций, рабочих групп
- Хранение пользовательских данных в зашифрованном виде
- Не надо иметь учетки на клиентах



# ANSIBLE TOWER - УПРАВЛЕНИЕ

INVENTORIES / MANAGE CLOUD STAGING SERVERS / EDIT

### CLOUD SERVERS

DETAILS NOTIFICATIONS

\* NAME DESCRIPTION SOURCE

Cloud servers [ ] Amazon EC2

CLOUD CREDENTIAL REGIONS INSTANCE FILTERS

Amazon keys [x] US East (Northern Virginia) tag:Name=\*staging\*

ONLY GROUP BY UPDATE OPTIONS

Overwrite  
 Overwrite Variables  
 Update on Launch

VARIABLES [YAML] [JSON]

1 ---

JOB TEMPLATES SCHEDULES / JOB TEMPLATE SCHEDULES.EDIT

### DAILY REMEDIATION

\* NAME \* START DATE (MM/DD/YYYY) \* START TIME (HH:MM:SS)

Daily remediation 10/03/2016 01:23:45

\* LOCAL TIME ZONE \* REPEAT FREQUENCY

America/New\_York Day

FREQUENCY DETAILS

Demo Job Template | SURVEY ON

### ADD SURVEY PROMPT

\* PROMPT

How many instances you need to deploy?

DESCRIPTION

JBOSS EAP instances to deploy

\* ANSWER VARIABLE NAME

eap\_count

\* ANSWER TYPE

Integer

MINIMUM MAXIMUM

1 100

DEFAULT ANSWER

10

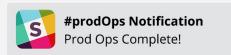
REQUIRED

CANCEL + ADD

- Динамическая инвентаризация клиентов
- Запуск сценариев по расписанию



- Использование опросника для задания доп. параметров
- Уведомления в режиме реального времени



# ANSIBLE TOWER - УПРАВЛЕНИЕ 'WORKFLOW'

EDIT WORKFLOW

KEY: — On Success — On Fail — Always P Project Sync I Inventory Sync TOTAL TEMPLATES 6

START → Demo Project (P) → (On Success) → Demo Job Template → (On Success) → Demo Job Template

START → Demo Project (P) → (On Fail) → Demo Job Template

START → Demo Project (P) → (Always) → Demo Job Template → (Always) → Demo Job Template → (Always) → Demo Job Template

DEMO JOB TEMPLATE

JOBS PROJECT SYNC INVENTORY SYNC

SEARCH Q KEY

NAME ^

Demo Job Template INFO

ITEMS 1 - 1 OF 1

\* TYPE

On Success

On Failure

CANCEL SELECT

**Provision → Configure → Deploy → Scale**

**Build → Test → Promote → Verify → Deploy**

Встроенный редактор рабочих процессов / workflow:

- Возможность группировать сценарии в зависимости от условий и результатов выполнения предыдущего
- Ролевая модель и результат выполнения задач

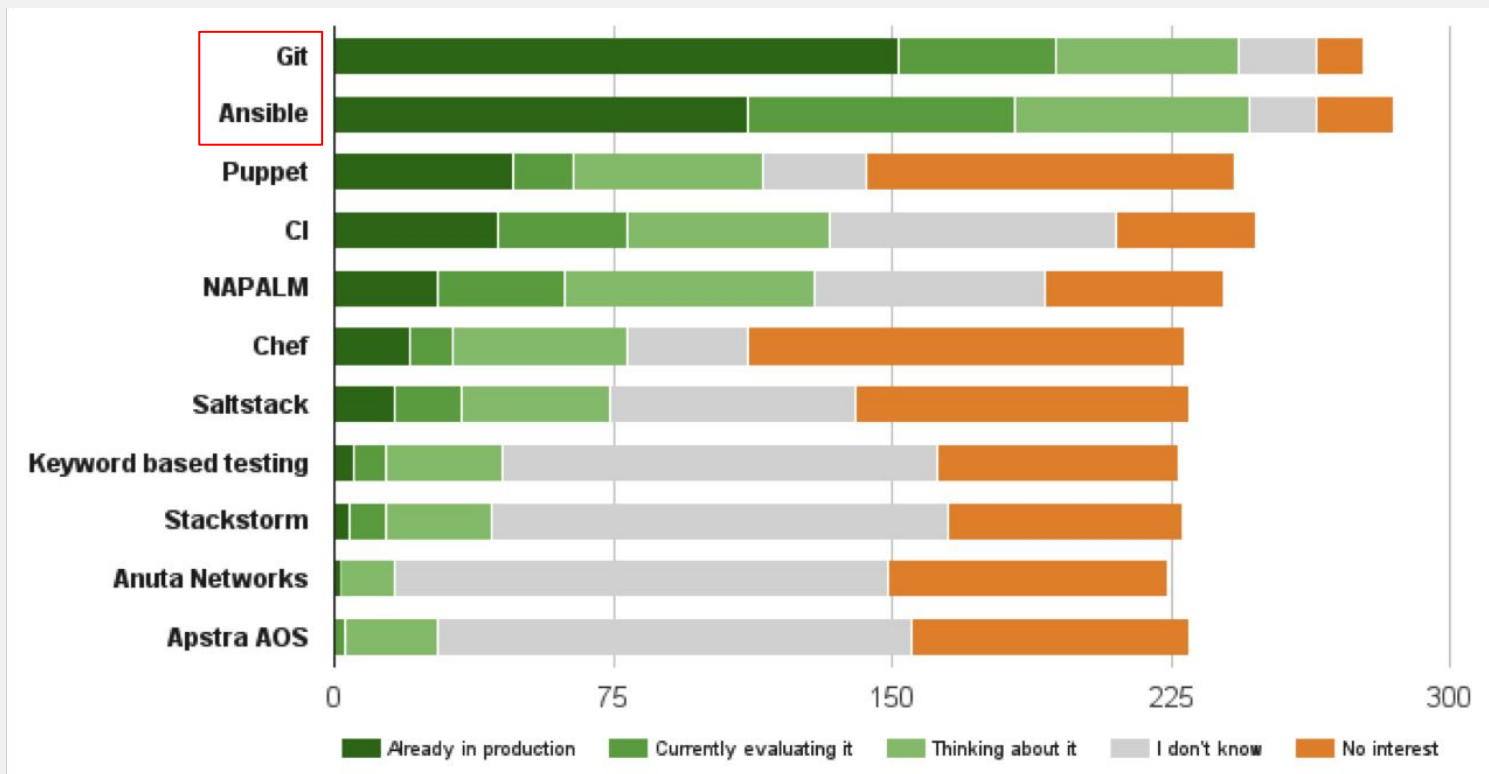
# ANSIBLE AUTOMATION - NETWORKING



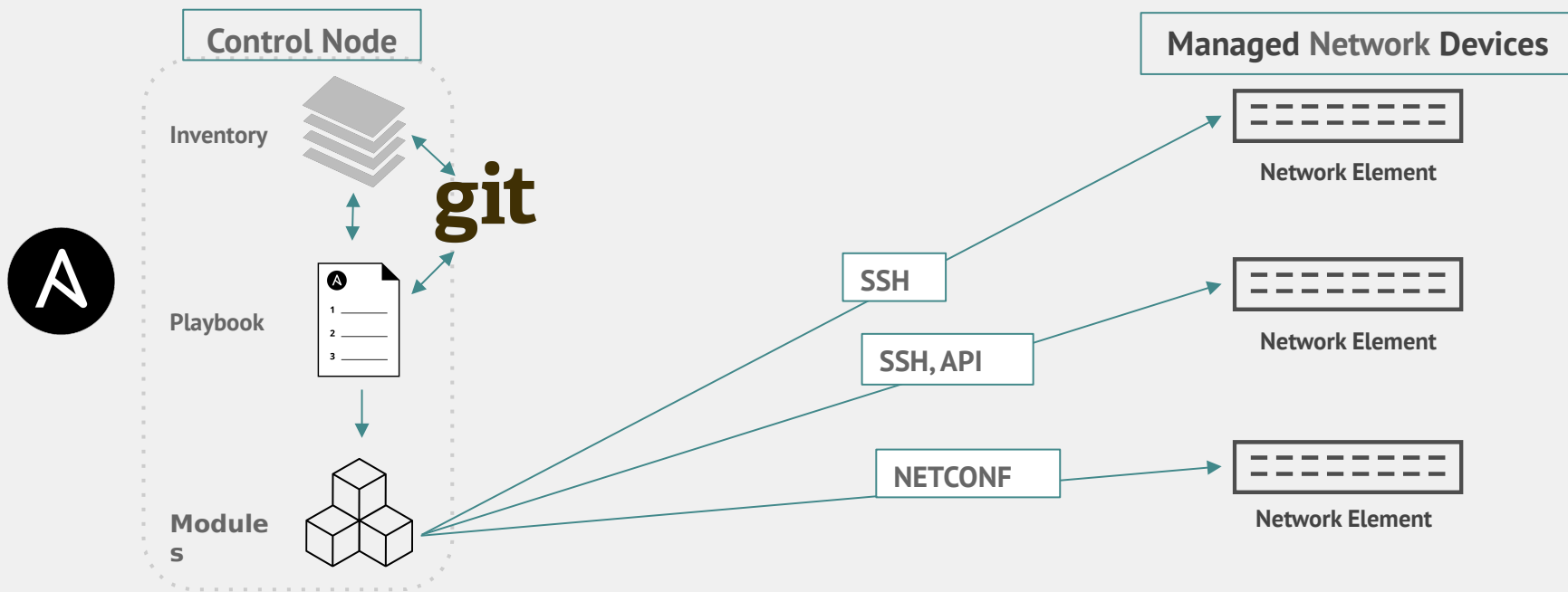
# MANAGING NETWORKS HASN'T CHANGED IN 30 YEARS.

- Networks are mission critical
- Every network is a unique snowflake
- Ad-hoc changes that proliferate
- Vendor specific implementations
- Testing is expensive/impossible

# NETWORKTOCODE – NETDEVOPS SURVEY (NOV. 2016)



# КАК РАБОТАЕТ ANSIBLE -> NETWORK



## Modules:

Handles execution of remote system commands

## Control Node:

Any client system (server, laptop, VM) running Linux or Mac OSX

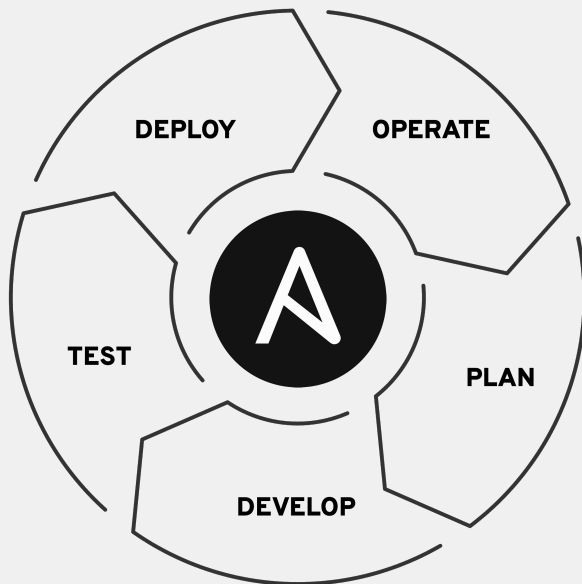
## Managed Nodes (Inventory):

A collection of endpoints being managed via SSH or API.



# ANSIBLE - START SMALL, THINK BIG

Three high-level benefits for successful network operations



## **INFRASTRUCTURE AS YAML**

- Automate backup & restores
- Manage “golden” versions of configurations

## **CONFIGURATION MANAGEMENT**

- Changes can be incremental or wholesale
- Make it part of the process: agile, waterfall, etc.

## **ENSURE AN ONGOING STEADY STATE**

- Schedule tasks daily, weekly, or monthly
- Perform regular state checking and validation

# TRANSFORMING THE NETWORKING

Use Ansible to manage, validate, and continuously track heterogeneous network device configurations and deployments.



**PHYSICAL NETWORK  
STATE VALIDATION**



**DATA CENTER  
FABRIC AUTOMATION**



**HYBRID CLOUD & CONTAINER  
NETWORK INTERCONNECT**

**45**

Networking  
platforms

**630+**

Networking  
modules

<https://www.ansible.com/overview/networking>

# NETWORK AUTOMATION PROGRESS

[https://docs.ansible.com/ansible/devel/modules/list\\_of\\_network\\_modules.html](https://docs.ansible.com/ansible/devel/modules/list_of_network_modules.html)

Полный список поддерживаемых Red Hat Ansible Engine Networking Add-On платформ и модулей:

<https://access.redhat.com/solutions/3184741>

[https://docs.ansible.com/ansible/latest/modules/network\\_maintained.html](https://docs.ansible.com/ansible/latest/modules/network_maintained.html)

7 Platforms  
28 Modules

**2.1**

May 2016

17 Platforms  
141 Modules

**2.2**

Oct 2016

29 Platforms  
267 Modules

Persistent  
Connections

NETCONF  
Support

**2.3**

Apr 2017

33 Platforms  
463 Modules

Declarative  
Intent

Aggregate  
Resources

Platform  
Agnostic

**2.4**

Sep 2017

40 Platforms  
572 Modules

network\_cli  
connection  
plug-in

NETCONF  
connection  
plug-in

Better  
Logging

XML  
Filters

More  
Docs

**2.5**

Mar 2018

45 Platforms  
639 Modules

net\_get

net\_put

netconf\_get

netconf\_rpc

httpapi  
connection  
plug-in

**2.6**

Jun 2018

# NETWORK MODULES: BUILT-IN DEVICE ENABLEMENT

A10

Dell OS6, OS9, OS10

Mellanox ONYX

Apstra AOS

Exoscale

Ordnance

Arista EOS, CVP

Extreme EX-OS, SLX-OS

NETCONF

Aruba Networks

F5 BIG-IP, BIG-IQ

Netvisor

AVI Networks

Fortinet FortiOS, FMGR

OpenSwitch

Big Switch Networks

Huawei CloudEngine

Open vSwitch (OVS)

Brocade Ironware

Illumos

Palo Alto PAN-OS

Cisco ACI, AireOS, ASA, IOS,  
IOS-XR, Meraki, NSO, NX-OS

Infoblox NIOS

Nokia NetAct, SR OS

Citrix Netscaler

Juniper JunOS

Ubiquiti EdgeOS

Cumulus Linux

Lenovo CNOS, ENOS

VyOS

# PLAYBOOK EXAMPLE: NETWORK AUTOMATION

```
---
- name: configure ios interface
  hosts: ios01
  tasks:
    - name: collect device running-config
      ios_command:
        commands: show running-config interface GigabitEthernet0/2
        provider: "{{ cli }}"
      register: config

    - name: administratively enable interface
      ios_config:
        lines: no shutdown
        parents: interface GigabitEthernet0/2
        provider: "{{ cli }}"
      when: "'shutdown' in config.stdout[0]`

    - name: verify operational status
      ios_command:
        commands:
          - show interfaces GigabitEthernet0/2
          - show cdp neighbors GigabitEthernet0/2 detail
      waitfor:
        - result[0] contains 'line protocol is up'
        - result[1] contains 'iosxr03'
        - result[1] contains '10.0.0.42'
      provider: "{{ cli }}"
```

```
(ansible) [network]$ ansible-playbook ios_interface.yaml
```

```
PLAY [configure ios interface] *****

TASK [collect device running-config] *****
ok: [ios01]

TASK [administratively enable interface] *****
changed: [ios01]

TASK [verify operational status] *****
ok: [ios01]

PLAY RECAP *****
ios01                : ok=3    changed=1    unreachable=0    failed=0
```

# EXTRAS FOR NETWORK AUTOMATION

## Network Resource Modules

```
tasks:
  - name: configure eos system properties
    eos system:
      domain_name: ansible.com
      vrf: management
      when: ansible_network_os == 'eos'
  - name: configure nxos system properties
    nxos system:
      domain_name: ansible.com
      vrf: management
      when: ansible_network_os == 'nxos'
```

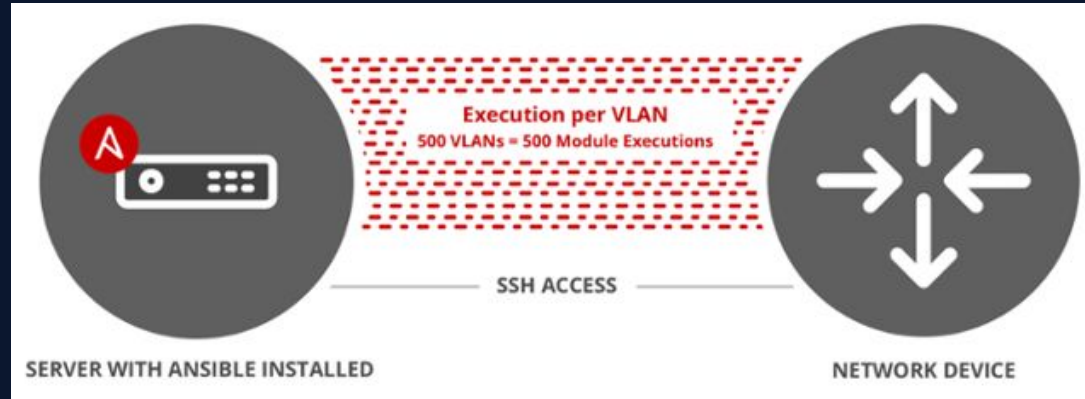
- Per Platform Implementation
- Focused on managing a resource
- Declarative by design
- Handles complexity

## Network Facts: For non-Linux systems there are vendor specific modules for fact collection

```
$ ansible -m ios facts routers -c network_cli
student1-rtr1 | SUCCESS => {
  "ansible_facts": {
    "ansible_net_all_ipv4_addresses": [
      "172.17.1.238"
    ],
    "ansible_net_all_ipv6_addresses": [],
    "ansible_net_filesystems": [
      "bootflash:"
    ],
    "ansible_net_gather_subset": [
      "hardware",
      "default",
      "interfaces"
    ],
    "ansible_net_hostname": "ip-172-17-1-238",
    "ansible_net_image": "bootflash:csr1000v-universalk9.16.05.01b.SPA.bin",
```

# EXTRAS FOR NETWORK AUTOMATION, #2

Loop Method  
("with\_items")



Aggregate  
Method  
(new -  
Ansible 2.4)



<https://www.ansible.com/blog/accelerate-ansible-networking-aggregate-resources>

# DECLARATIVE INTENT WITH ANSIBLE

N.B. Not All  
Modules &  
Parameters

Declared  
Configuration

Intended  
State

```
- name: configure interface
  net_interface:
    name: GigabitEthernet0/2
    description: public interface configuration
    enabled: yes
    state: connected
    neighbors:
      - host: core-01
        port: Ethernet5/2/6
```

```
- name: configure bgp neighbor
  net_bgp_neighbor:
    peer: 1.1.1.1
    remote_as: 65000
    enabled: yes
```

## CONFIGURATION

Only perform configuration. Ignore resource state on the device

## VALIDATE STATE

Only perform state validation. Ignore configuration of the resource

```
- name: validate bgp neighbor
  net_bgp_neighbor:
    peer: 1.1.1.1
    nbr_state: established
    pfx_rx: 16593
    pfx_tx: 132
```

<https://www.ansible.com/blog/networking-features-in-ansible-2-4>

<https://www.ansible.com/blog/coming-soon-networking-features-in-ansible-2.5#Module>



# GIT FOR NETWORK AUTOMATION

- 1) Store everything in Git with version control:
  - playbooks, roles, inventories,
  - configs, other files
- 2) Track drift of config files with diff in Git
- 3) Use Git's Pull (Merge) Request functionality to include approval workflows
  - modification of inventory
  - modification of playbooks & roles
  - modification of 'master' configs

```
7 ■■■■■ nxosconfig.txt
...      ...      @@ -1,9 +1,11 @@
1        1        vlan 2
2        2        name ESXi_MGMT
3        3        vlan 3
4        4        name ESXi_vMotion
5        5        vlan 10
6        6        name ServerNetwork
7        7        +vlan 11
8        8        + name ServerNetwork2
```

Merge request approvals ⓘ

**Approvers**

Search for users or groups Add

Add users or groups who are allowed to approve every merge request

Approvers 9

- Sean McGivern ✕
- Victor Wu ✕
- Group: GitLab Product Team 8 ✕

Approvals required

5 ⌵

Set number of approvers required before open merge requests can be merged

Can override approvers and approvals required per merge request ⓘ

Remove all approvals in a merge request when new commits are pushed to its source branch

# NETWORK SERVICE AUTOMATION - UNNAMED REF

Data Center and WAN Network Devices. > 10,000 Multi-vendor Devices – NXOS, JunOS, IOS-XR, EOS

## Before Red Hat Automation Tools:

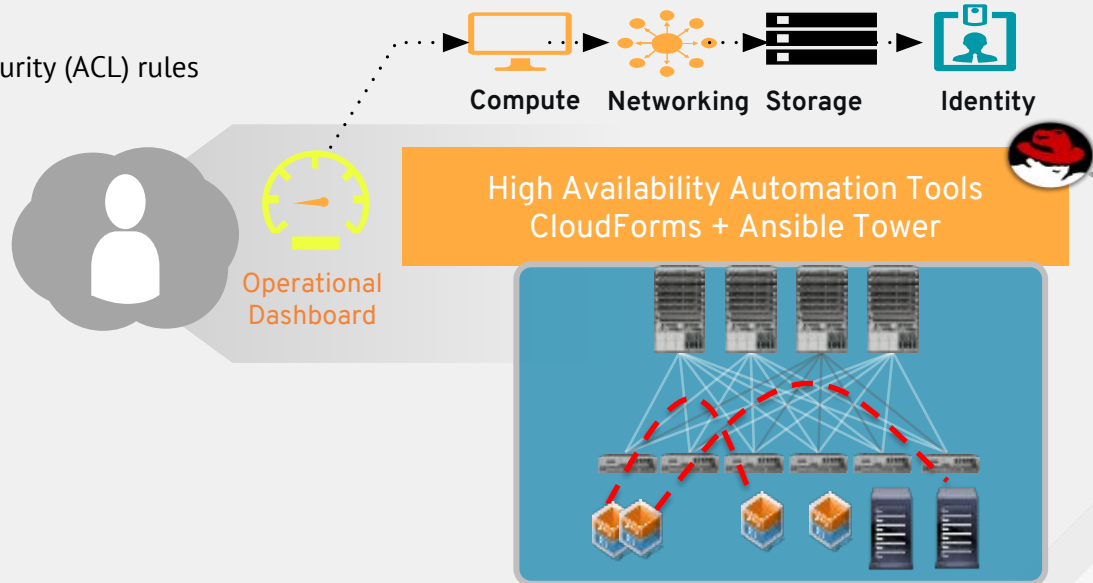
- Manual operator translation of daily changes to security (ACL) rules
- Not Compliant with Operational Business Rules

## After:

- Ansible Tower + CloudForms
- Automated Rule Playbooks
- Daily Remediation of Business Rules
- Hours to Provision One Device Reduced to Minutes

## Key Benefits:

- Fully Automated, Open, Secure
- Saves Time – More Productive Teams
- Fewer Mistakes / Errors
- Significant Reductions in Complexity
- Apply Across Other Service Deployments



**Faster Deployments -> Lower Costs**

# NETWORK VENDORS CAN'T AVOID TREND



**AUTOMATE CISCO ENVIRONMENTS  
WITH RED HAT ANSIBLE AUTOMATION**

Optimize workflows, aggregate IT platforms, and unify infrastructure teams

PARTNER SOLUTION OVERVIEW

## NEW NETWORK SOLUTIONS FOR A NEW ERA

Enterprise networking and server connectivity have come a long way in the last few decades. From traditional on-premise, appliance-based networks to new applications programming interface (API)-focused, virtualized environments, the need for flexibility in management has never been greater.

Cisco is a global leader in compute, network, and cloud solutions for enterprises, data centers, and any infrastructure in between. But as businesses expand and needs evolve, companies are searching for new approaches to meet increased end-user demand while continuing to deliver value to the business—all with limited resources. Automation will become an ever-increasing strategy for enterprises, because as they are finding out, automation is no longer about getting ahead—it is more about keeping up.<sup>1</sup>

Red Hat can help. As the global, trusted leader in open source solutions for today's enterprises, Red Hat has been helping organizations of all shapes and sizes adapt to continually changing IT requirements. With Red Hat® Ansible® Automation as part of the Red Hat management portfolio, organizations are now able to gain greater efficiency, deliver value faster, and solve IT and business workflow challenges across the entire IT organization.

## MANAGING MORE WITH LESS

<https://www.redhat.com/en/resources/automate-cisco-environments-ansible-automation-partner-solution-overview>



# Swisscom Selects Red Hat Ansible Tower to Improve Efficiency and Reduce Deployment Time

Red Hat Ansible Automation helps Swisscom stay competitive and free resources for strategic tasks and innovation

*Bojan Nikolic, Head of I.T. Service Delivery, Swisscom*

“We’re very pleased with the progress of the project with Ansible Tower and we’re ecstatic to have a central solution for automation. We chose Ansible Tower partly because it’s easy to use and learn. You can make very fast progress compared to other configuration management tools. With our current deployment, we can save many operation hours and invest the time of our engineers in more valuable tasks.”

To continue to successfully operate in the highly competitive network infrastructure market, Swisscom is constantly seeking ways to reduce costs, work more efficiently, and speed up response times. As part of these ongoing efforts, the company sought to automate the management of its IT processes, systems, and components.

Swisscom deployed Red Hat Ansible Tower for automation and orchestration across multiple environments with different network configurations. After production implementation is complete, five teams of 80 users will use Red Hat Ansible Tower to automate the management of about 15,000 components, including servers, firewalls, network devices, and storage devices. For example, the team responsible for Swisscom TV infrastructure services uses Red Hat Ansible Tower to automate and orchestrate radio stations and monitoring its OpenStack platform.

Key benefits of Red Hat Ansible Tower for Swisscom include:

- Role-based access: this allows employees in a broad range of jobs complete tasks and actions, instead of waiting for access permission to be granted manually.
- More effective collaboration on shared projects: teams can combine, centralize, and share Red Hat Ansible Playbooks with the application and database teams, improving quality and standardization.
- Defining specific inventories for running playbooks and giving access to other teams or employees to run processes on servers without requiring local server access. As a result, teams can focus on their core work, and tasks can be completed faster and more accurately.

ZURICH/BERN – September 11, 2018 – Red Hat, Inc. (NYSE: RHT), the world’s leading provider of open source solutions, today announced that Swisscom, Switzerland’s largest telecommunications company and one of its leading IT companies, has deployed Red Hat Ansible Tower as its enterprise-wide IT and network automation platform. By using Red Hat Ansible Tower, Swisscom is able to focus on more valuable development projects and accelerate response times for resource requests.

*“We’re very pleased with the progress of the project with Ansible Tower and we’re ecstatic to have a central solution for automation. We chose Ansible Tower partly because it’s easy to use and learn. You can make very fast progress compared to other configuration management tools. With our current deployment, we can save many operation hours and invest the time of our engineers in more valuable tasks.”*

BOJAN NIKOLIC, HEAD OF I.T. SERVICE DELIVERY, SWISSCOM

# SDN VS AUTOMATION

BENEFIT	SDN	AUTOMATION
Reconfigure the network from a central point	✓	✓
Reduced vendor lock in with commodity hardware	?	✓
Leverage existing infrastructure	✗	✓
Decouple control plane from data plane	✓	✓
Reduced opex/capex costs	?	✓

# NETCONF & OPENCONFIG

## Ansible 2.4 and older

local connection method using provider

```
1 ---
2 - hosts: rtr1
3   connection: local
4   gather_facts: no
5
6   vars:
7     login:
8       username: admin
9       password: ansible
10      transport: netconf
11
12   tasks:
13     - name: Backup configuration
14       junos_config:
15         backup: yes
16         provider: "{{login}}"
17
```



## Ansible 2.5

netconf connection method

```
1 ---
2 - hosts: rtr1
3   connection: netconf
4   remote_user: admin
5
6   tasks:
7     - name: Backup configuration
8       junos_config:
9         backup: yes
```

<https://www.ansible.com/blog/coming-soon-networking-features-in-ansible-2.5>

<http://karneliuk.com/2018/08/new-netconf-modules-in-ansible-2-6-examples-for-arista-eos-cisco-ios-xr-and-nokia-sr-os/>

# ANSIBLE TOWER UI

NAME ^	DESCRIPTION ▾	ACTIVITY	LABELS	ACTIONS
Bug Check				
Continuous Compliance	Validate ephemeral state			
Daily Switch Report	Run basic checks against all devices			
Deploy Arista Pod	deploy spine and leaf			
Ops Playbook				
Run Ad-hoc Commands	Provide a list of commands to run			
Sanitize Demo	Reset Lab to Initial State			
Troubleshoot Virtual M...	Gather network related info for any VM			
Upgrade EOS	Upgrade EOS to a specific version			

# ANSIBLE TOWER UI

The screenshot shows the 'SOURCES' tab for the organization 'darkbulb-nxos'. At the top, there are navigation tabs: DETAILS, PERMISSIONS, GROUPS, HOSTS, SOURCES (selected), and COMPLETED JOBS. Below these is a search bar with a 'KEY' button and a 'SEARCH' button. To the right, there are buttons for 'SYNC ALL' and '+ ADD SOURCE'. A table lists the sources:

SOURCES	TYPE	ACTIONS
darkbulb-nxos: autodiscovery_inventory	Sourced from a Project	Start sync process, Refresh, Calendar, Edit, Delete

At the bottom right, it says 'ITEMS 1 - 1'.

The screenshot shows the 'INVENTORIES' tab. At the top, there are navigation tabs: INVENTORIES (selected) and HOSTS. Below these is a search bar with a 'KEY' button and a 'SEARCH' button. To the right, there is a '+ ADD' button. A table lists the inventories:

NAME	TYPE	ORGANIZATION	ACTIONS
darkbulb-nxos	Inventory	DEMO	Edit, Delete
Demo Inventory	Inventory	Default	Edit, Delete

Playbook for devices discovery (sweep address range; facts + snmp) -> upload to Git -> Use as Inventory in Ansible Tower

[https://github.com/victorrock/autodiscovery\\_inventory/blob/darkbulb-nxos/hosts.yaml](https://github.com/victorrock/autodiscovery_inventory/blob/darkbulb-nxos/hosts.yaml)

[https://github.com/victorrock/autodiscovery\\_inventory/blob/clean/main.yaml](https://github.com/victorrock/autodiscovery_inventory/blob/clean/main.yaml)



# ANSIBLE ДЛЯ СЕТЕВОЙ ИНФРАСТРУКТУРЫ

## Самые привлекательные сценарии применения

- Часть DevOps и Трансформационной истории
  - Кроме контейнеров актуальна и автоматизация legacy, а Network - наиболее legacy
  - Можно дополнять Ansible для ИТ, реализовывать параллельно, или, наоборот, первым
- Альтернатива/дополнение к вендорским EMS/NMS
- Альтернатива вендорским SDN
  - Вместо закупки контроллера и доп.лицензий

## Самые привлекательные сценарии применения

- Новые дата-центры и проекты модернизации существующих
- Территориально-распределенная инфраструктура с типовыми блоками устройств

## Важно

- В конкурентах только 'Do Nothing' и DIY. Популярный для VM Puppet несопоставим, различные SNMP средства удачно дополняются Ansible, начиная с настройки SNMP и агентов на устройстве
- Стоимость мала на фоне затрат на сетевое оборудование и обслуживающих его людей

# RED HAT ANSIBLE AUTOMATION (NETWORK)

## NETWORK MODULES

- **Developed, maintained, tested, and supported** by Red Hat
- **140+ supported modules** and growing\*
- Red Hat **reports and fixes problems**
- **Networking modules included** with Ansible Engine offering, but **the Ansible Engine Networking Add-On SKU purchase is required** for full support

\*take special note of the specific supported platforms

### NETWORKING ADD-ON

#### INCLUDED SUPPORT:

Arista EOS

Cisco IOS

Cisco IOS XR

Cisco NX-OS

Juniper Junos

Open vSwitch

VyOS

Net (Platform-Agnostic)

# ANSIBLE - ПОЛЕЗНЫЕ РЕСУРСЫ

- Ansible Networking Homepage: <https://www.ansible.com/overview/networking>
- Ansible & Ansible Tower <https://www.ansible.com/resources/get-started> <https://docs.ansible.com/>
- Educational webinars <https://www.ansible.com/resources/webinars-training>
- Videos. incl. Ansible Fest presentations <https://www.ansible.com/resources/videos>
- Join the Community  
Users list: [ansible-project](#)      Development list: [ansible-devel](#)      Announce list: [ansible-announce](#) (read only)  
[irc.freenode.net: #ansible](#)      [slack.networktocode.com: #ansible](#)      [ansiblenetwork.slack.com](#)
- The Ansible Networking Linklight project is intended for effectively demonstrating Ansible's capabilities through instructor-led workshops or self-paced exercises. <https://github.com/network-automation/linklight>
- Вебинары @ru про Ansible <https://www.facebook.com/RedHatRussia/videos/1819400778131240/>  
Ansible для Network: <https://www.facebook.com/RedHatRussia/videos/2050866404944882/>

