NFE Networking For Everyone

DevNet

Темы

- YAML
- Docker
- Git, CI/CD
- Ansible
- Terraform
- Jenkins
- …

- NETCONF
- Netmiko
- Nornir
- gNMI
- Ansible
- …

- Kubernetes

Ресурсы

- Linux/Unix workstation (WSL для Windows)
- EVE-NG: 4vCPU + 8Gb
- Google/Yandex cloud
- github.com

Драйверы NetDevOps

- Рост бизнеса
  - Виртуализация приложений и постоянно уменьшающийся time to market
  - Внедрение новых сервисов
  - Сервисы в облаках
- Programmability
  - Python: низкий порог входа и разнообразие библиотек
  - Решение комплексных задач по обслуживанию сети
- Современные программные продукты
  - Device-to-device управление
  - Возможность собирать много информации о состоянии сети и проводить корреляции между событиями

Потребности бизнеса

Потребитель

- Есть потребность
- Готов платить за решение

Конкуренты

- Делают решение лучше/за меньшие деньги

Предприниматель

- Есть возможность и средства предоставить решение

Эволюция

- Изменения потребностей клиента
- Глобальная экономика
- Ситуация на рынке

Сложные рутинные задачи

Для планирования оценить количество неактивных за последние 3 месяца портов

- для каждого коммутатора моей сети
  - для каждого интерфейса коммутатора
    - если статус интерфейса down больше 3 месяцев
      - добавляем в результирующий список

```
result = []
for switch in network:
    for interface in switch:
        if interface.is_down() and interface.last_change() > three_months:
            result.append(interface)
```

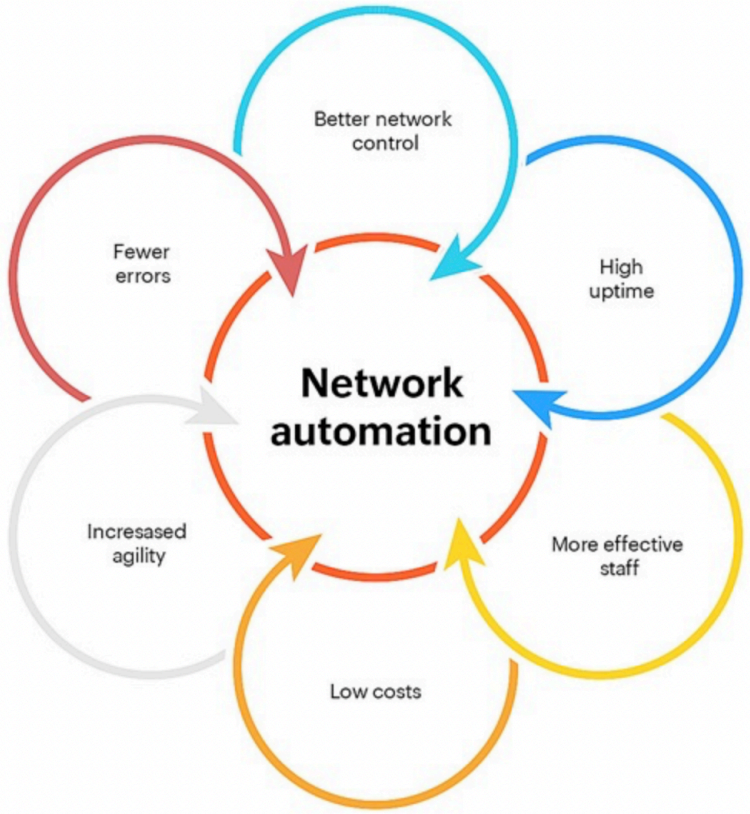
```
>>> import requests
>>> requests.get("http://10.211.55.21/api/status/").json().get("netbox-version")
'3.6.0'
```

Network Automation

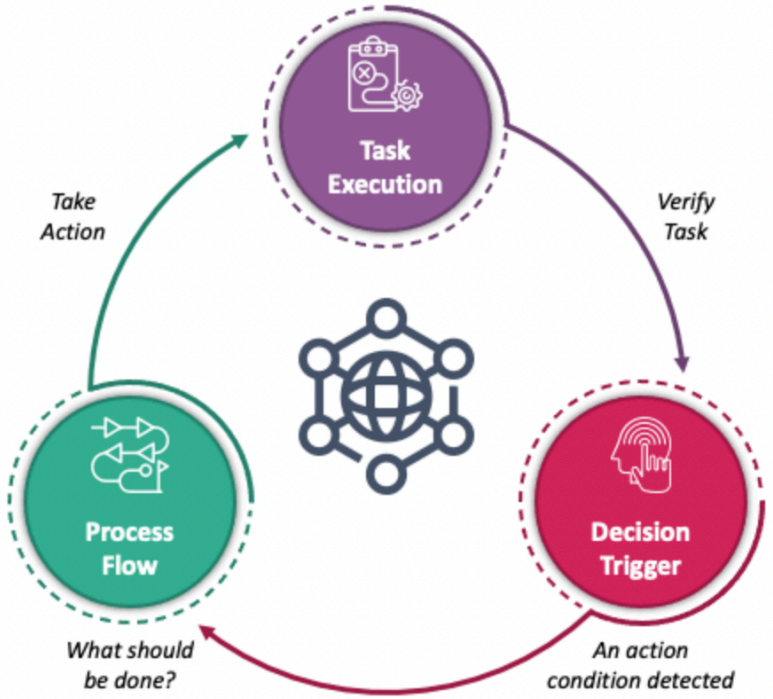
Практика автоматизации подготовки/применения конфигурации, тестирования, операционной деятельности, обнаружения новых устройств и прочее на сети.

- увеличение операционной эффективности
- уменьшение вероятности ошибок
- уменьшение затрат на обслуживание сети

Самостоятельное выполнение одной задачи без вмешательства оператора.



Network Orchestration



Эволюция подхода “Network Automation”

Выполнение множества связанных задач без вмешательства оператора.

Automation	Orchestration
Одна low-level задача	Множество последовательных задач
Ограничена типом устройств/производилем	Мультивендорное решение
Не учитывает состояние сети	Состояние/статус/конфигурация учитывается при выполнении задач

Network Orchestration

- Сбор device inventory (конфигурация, модель, ОС и т.д.)
- Логирование, хранение, восстановление
- Network state aware
- Device/Vendor agnostic
- GUI-based интерфейс управления

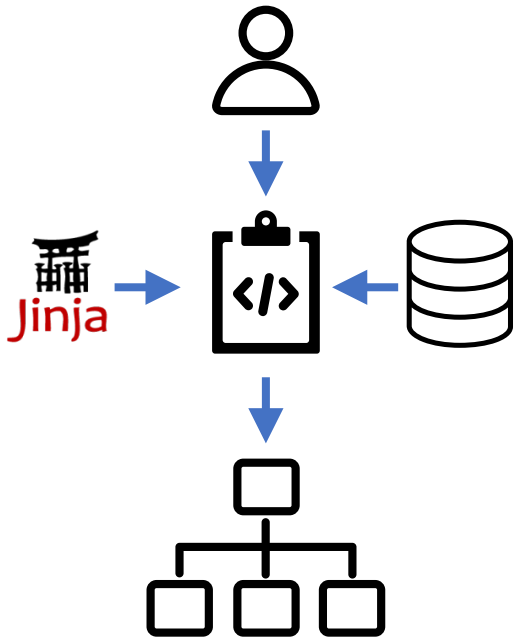
Network Orchestration

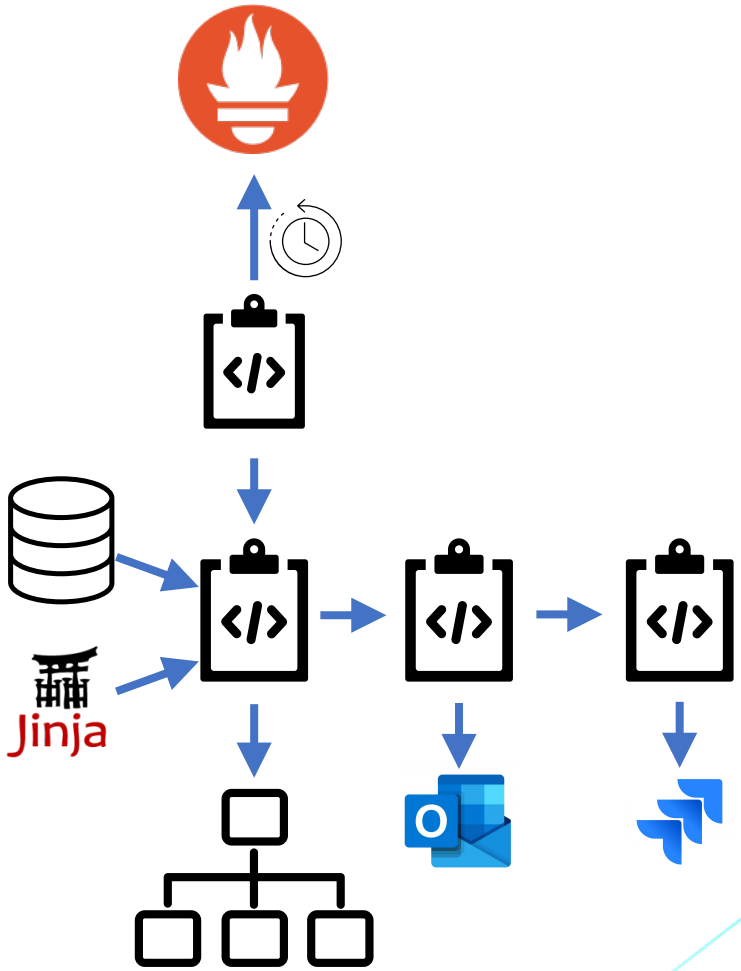
- Policy-Based Automation (PBA): самый простой тип. Конфигурационные политики в виде шаблонов прописаны в системе и система предоставляет интерфейс управления/загрузки/сверки и пр.

- Software-Defined Networking (SDN): изначально идея была разделить control-plane и data-plane, но широкого распространения она не получила (пока), но другие концепции, например: отдельный контроллер для управления/мониторинга, или REST интерфейс для интеграции с другими системами, сейчас успешно применяются

- Intent-Based Networking Systems (IBNS): самый комплексный и сложный вид. Абстрагируется до потребностей бизнеса и оперирует “их” терминами. Использует ML/AI для прогнозов, поиска неисправностей, корреляции событий. Применяет декларативный подход.

Network Automation & Orchestration





Декларативный и императивный подходы

императивный - как делать?

```
#!/bin/bash
if [ $(id -u) -eq 0 ]; then
    username="tony"
    egrep "^$username:" /etc/passwd >/dev/null
    if [ $? -eq 0 ]; then
        echo "user '$username' already exists"
        exit 1
    else
        useradd -m $username
        if [ $? -eq 0 ]; then
            echo "user '$username' has been added to system"
        else
            echo "failed to add a user '$username'"
        fi
    fi
else
    echo "should be executed under the root"
fi
```

декларативный - что делать?

```
---
- name: "add user to the host"
  hosts: host01
  become: true
  tasks:
    - name: "add 'mark' user"
      ansible.builtin.user:
        name: "mark"
        state: "present"
```

Challenge

Приложения	Сеть
За последние годы сервисы виртуализировались и time-to-market значительно сократился	Сетевая инфраструктура статичная, это все еще специализированные аппаратные устройства
Сервисы легко обновить/мигрировать	Много LDoS устройств
Высокая и быстрая масштабируемость	Планирование -> закупка -> инсталляция
Декларативный подход	Императивный подход
Преобладают m2m интерфейсы	В основном human интерфейсы

Преимущества NetDevOps

- Оптимизация операционной деятельности (уход от использования CLI, автоматизация day-to-day активностей, сбор диагностической информации и прочее)
- Быстрое развертывание нового функционала/узлов (замена оборудования, открытие нового сайта, добавление нового DMVPN хаба)
- Уменьшение рисков наступления сбоя в сети (применение шаблонов, частые и маленькие изменения)
- Улучшение коммуникации с бизнесом (разнообразные отчеты, планирование)

Инструменты

- Infrastructure as Code (IaC)
  - Ansible
  - Terraform
- CI/CD
  - GitHub Actions
  - GitLab Pipelines
  - Jenkins
- Testing
  - EVE-NG
  - GNS3
  - Cisco CML
- Source Control
  - GitHub
  - GitLab
  - Bitbucket
- Monitoring
  - ELK stack (Elasticsearch, Logstash, Kibana)
  - Splunk
  - Prometheus

Характеристики решений

- cloud или on-premises
- managed или нет
- open Source, vendor, in-house
- интеграция в существующую экосистему

Характеристики инфраструктуры

- воспроизводима

- Консистентно

- взаимозаменяема

- не имеет конечного состояния

IaC - описание состояния инфраструктуры через machine-readable файлы.

Адаптация

- Automation и Operations это одна команда или разные
- “изменение мышления”: сначала шаблоны, затем изменения
- проблема поколений
- существующая экосистема
- решение отдельных задач

Legacy?

TCL на маршрутизаторах Cisco

```
conf t
alias exec set_active tclsh flash:/active.tcl
alias exec set_backup tclsh flash:/backup.tcl
end


tclsh
puts [open "flash:/active.tcl" w+] {
ios_config "route-map rm_to_RR permit 10" "no set community" "set community 64512:12345 additive" "no set as-path prepend 64512"
exec "clear bgp vpnv4 unicast peer-group pg_RR soft out”
ios_config "route-map rm_BGP_to_OSPF permit 10" "set metric 10"
}
tclquit


tclsh
puts [open "flash:/backup.tcl" w+] {
ios_config "route-map rm_to_RR permit 10" "no set community" " set community 64512:12345 additive" "set as-path prepend 64512"
exec "clear bgp vpnv4 unicast peer-group pg_RR soft out"
ios_config "route-map rm_BGP_to_OSPF permit 10" "set metric 20"
}
tclquit
```

Конфигурационный дрифт

- Внесение недокументированных изменений:
  - Приходит задача: “срочно закрыть порты”
  - Изменения вносятся руками
  - Система отклоняется от шаблона конфигураций (желаемое состояние)

- Повторяем процедуру несколько раз

- Система накапливает конфигурационный дрифт и становится неуправляемой через программные комплексы

Изменение инфраструктуры

Задача: автоматизация изменений инфраструктуры

Контекст: baremetal / VM

Метод: config sync / immutable inrfa

Инструменты: PXE? Ansible? Terraform?

Immutable infra:

- Минимальный дрифт конфигурации: все образы одинаковые
- Масштабируемость: быстрое и легкое горизонтальное масшатбируемость
- Безопасность: перекатывание образов уменьшает эффект от взлома
- Стоимость: aws vs gcp vs yandex cloud

Изменение инфраструктуры

Задача: автоматизация изменений инфраструктуры

Контекст: k8s / containers

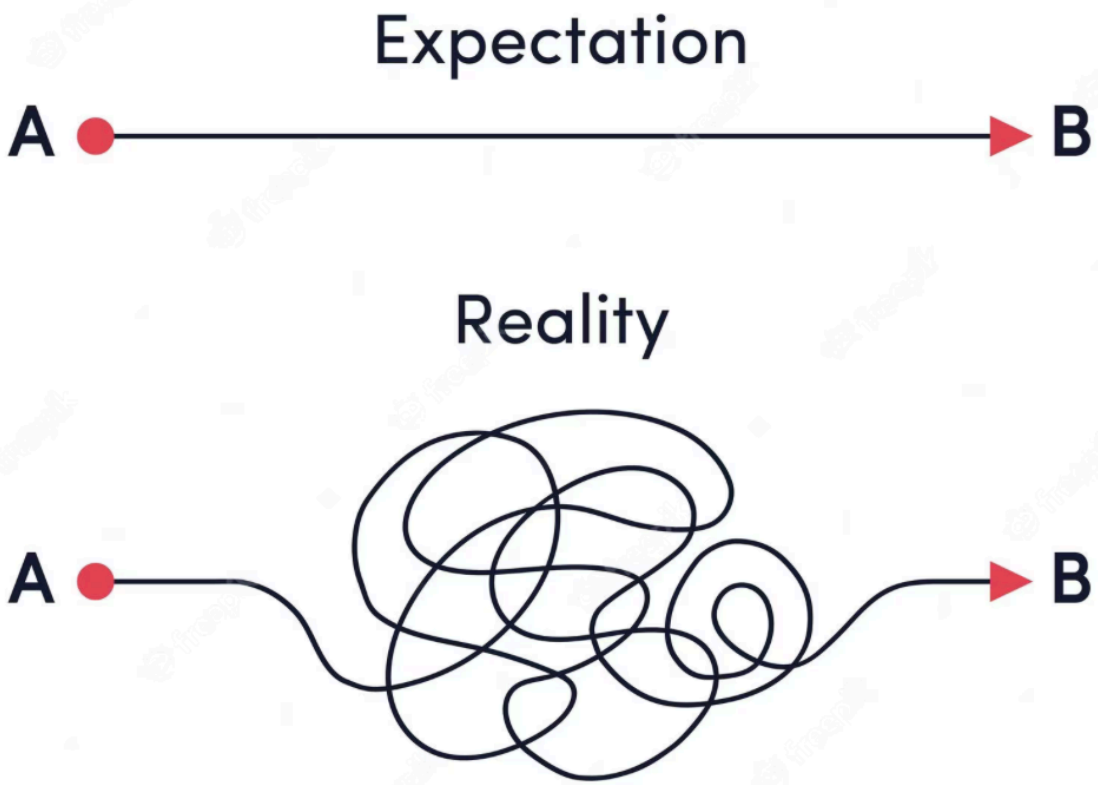
Метод: immutable inrfa / GitOps

Инструменты: Terraform? Helm?

GitOps:

- Все плюсы immutable inrfa
- Повторяемость
- Приложения доставляются сразу в контейнер при сборке

Реальность



NFE Networking For Everyone