99% fulfillment

- hear beat stops every 85s
- 30 typos pro DIN A4 page

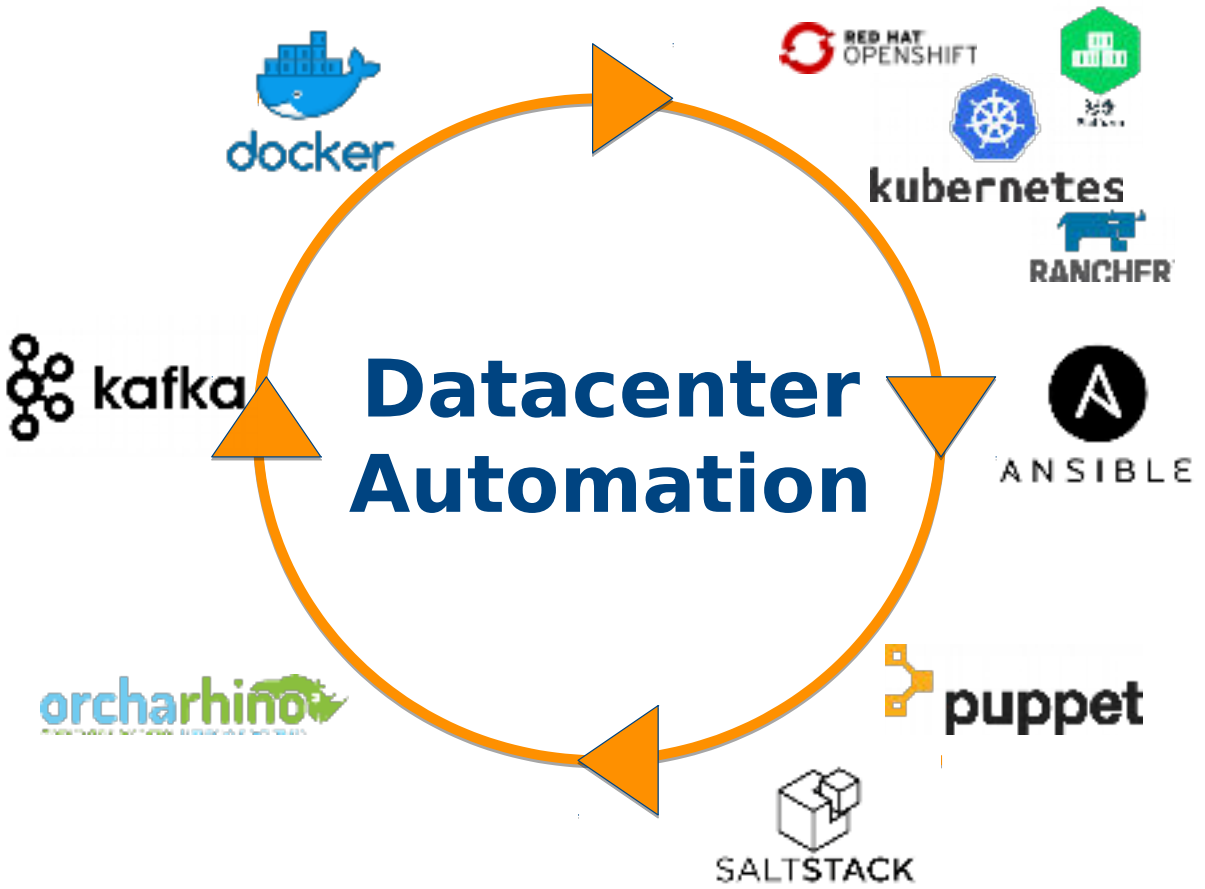
99.9% fulfillment

- 2 crashes at Frankfurt Airport per day
- every hour 22000 wrong bank bookings
- 32000 missed heart beats per year

IT needs automation!

Automation helps to solve problems whose existence one is often not aware of.

Atix AG - all about Automation



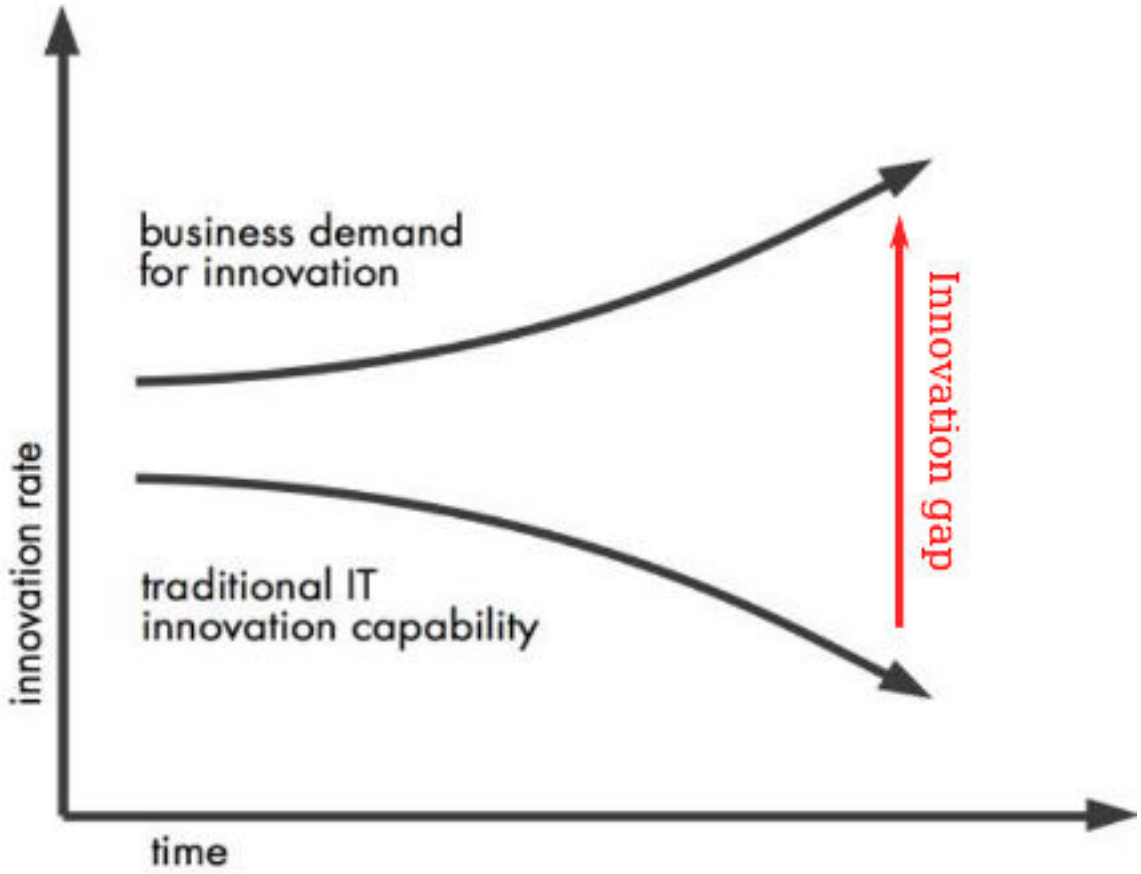
Automating Kafka...

...with Ansible

Dr. Bernhard Hopfenmüller

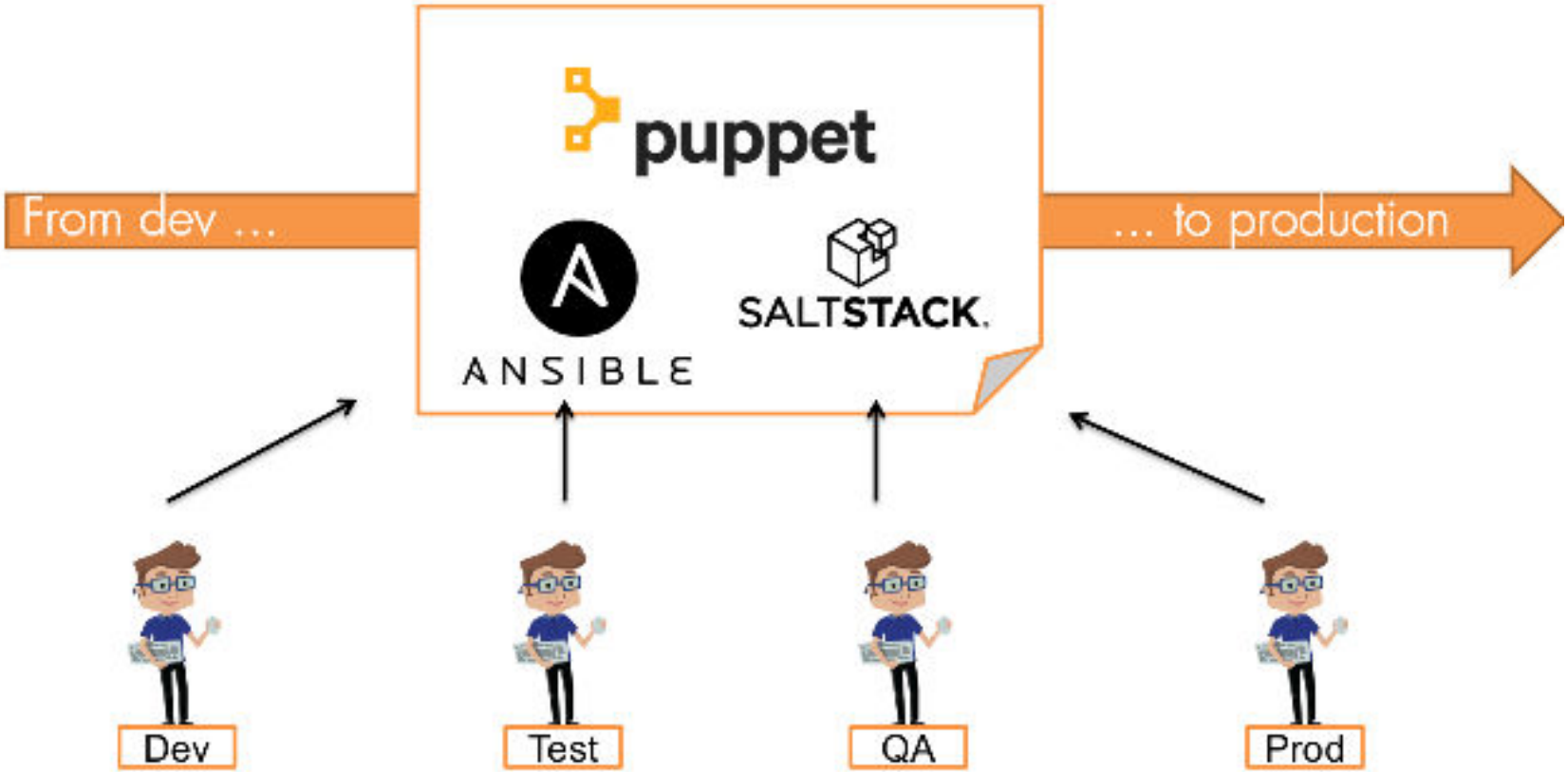
28. Mai 2019

The innovation gap

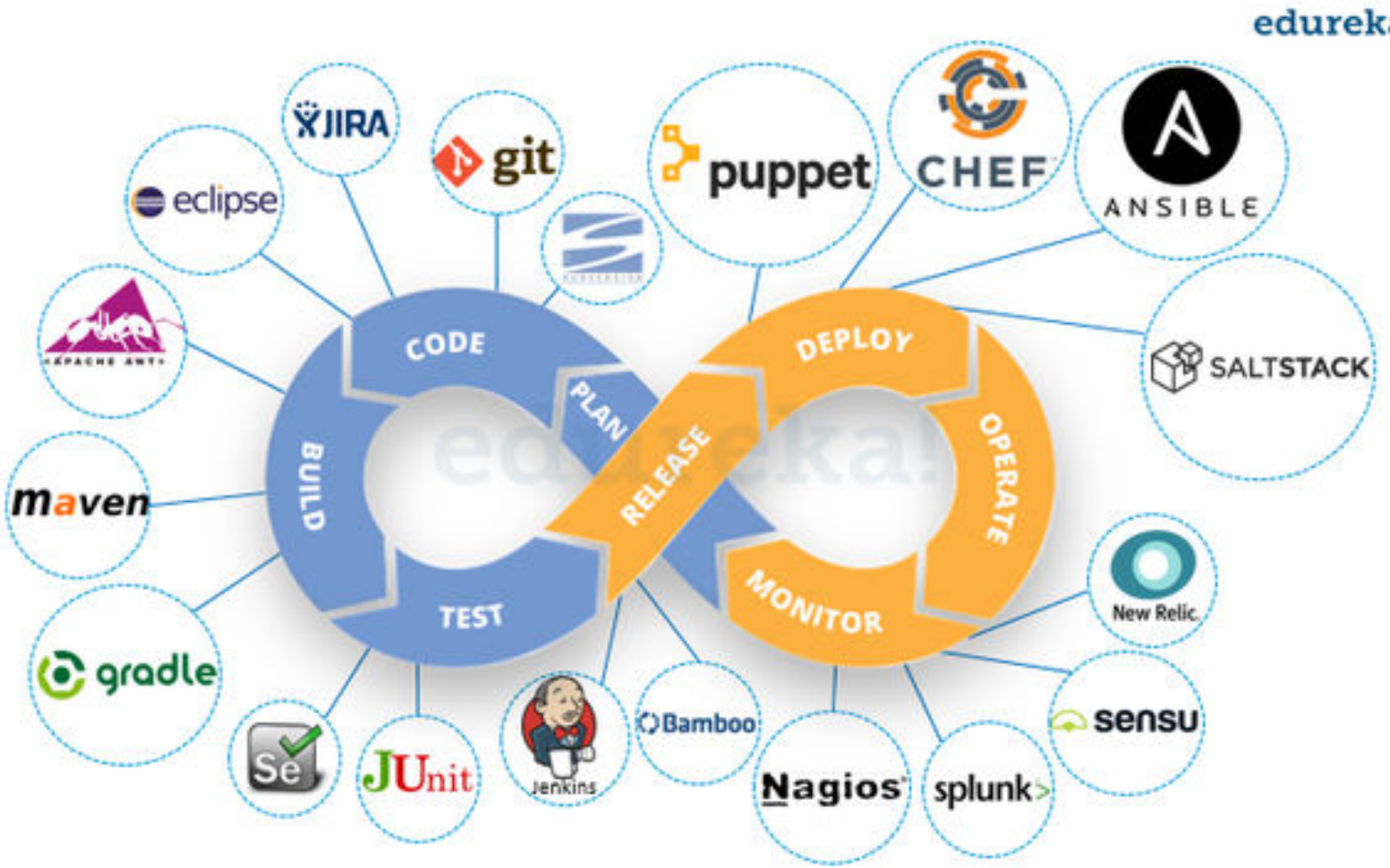


Innovation Gap is closed by Automation and Standardization

Standardization means common IT language



DevOps


edureka!
JIRA
git
puppet
CHEF
ANSIBLE
eclipse
SALTSTACK
maven
gradle
CODE
PLAN
BUILD
TEST
RELEASE
DEPLOY
OPERATE
MONITOR
Se
JUnit
Jenkins
Bamboo
Nagios
splunk>
New Relic.
sensu

Configuration Management

- Keep config up to date (Git)
- Separation of data and code
- Infrastructure as code
  - Code is reusable
  - Data is configuration
- Idempotency

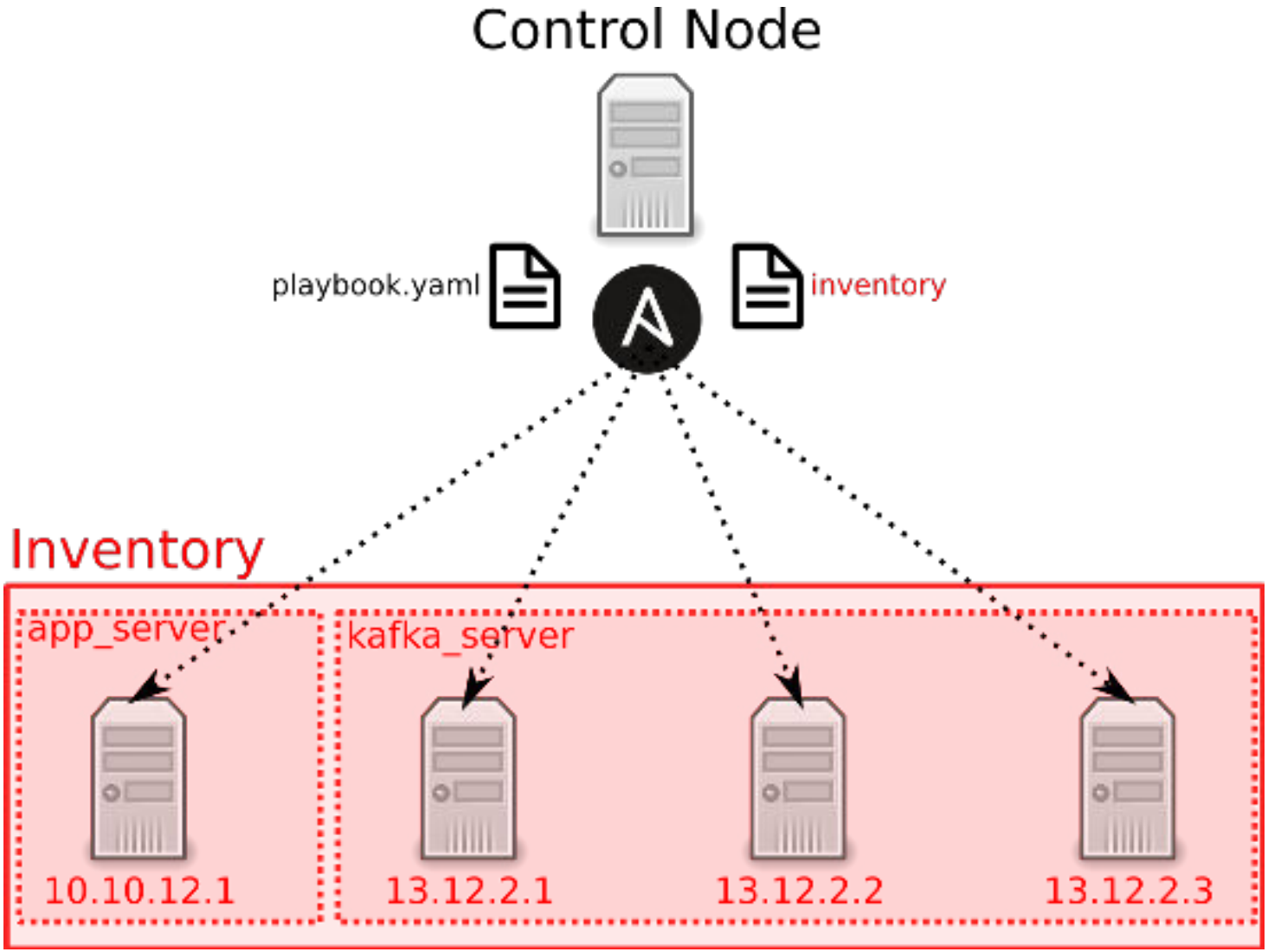
Ansible in a nutshell

Ansible …

- … is „radically simple“ Puppet -
- … has low requirements Puppet -
- … needs no agent Puppet Salt -
- … is human readable Puppet -

→ Ansible is the best choice for automated deployments

Run Ansible


Control Node
playbook.yaml
inventory
Inventory
app_server
kafka_server
10.10.12.1
13.12.2.1
13.12.2.2
13.12.2.3

Inventory

```
[kafkaserver]
web1.server.com ansible_host=13.12.2.1
web2.server.com ansible_host=13.12.2.2
web3.server.com ansible_host=13.12.2.3

[appserver]
app1.server.com ansible_host=10.10.12.1
```

- defining the infrastructure
- ini or yaml syntax
- combine servers into groups
- define specific vars

Playbook

```
---
- hosts: kafkaserver # mapping role to host
  become: true # run as root
  roles:
    - preflight # role1
    - zookeeper   # role2
    - kafkabroker # role3
```

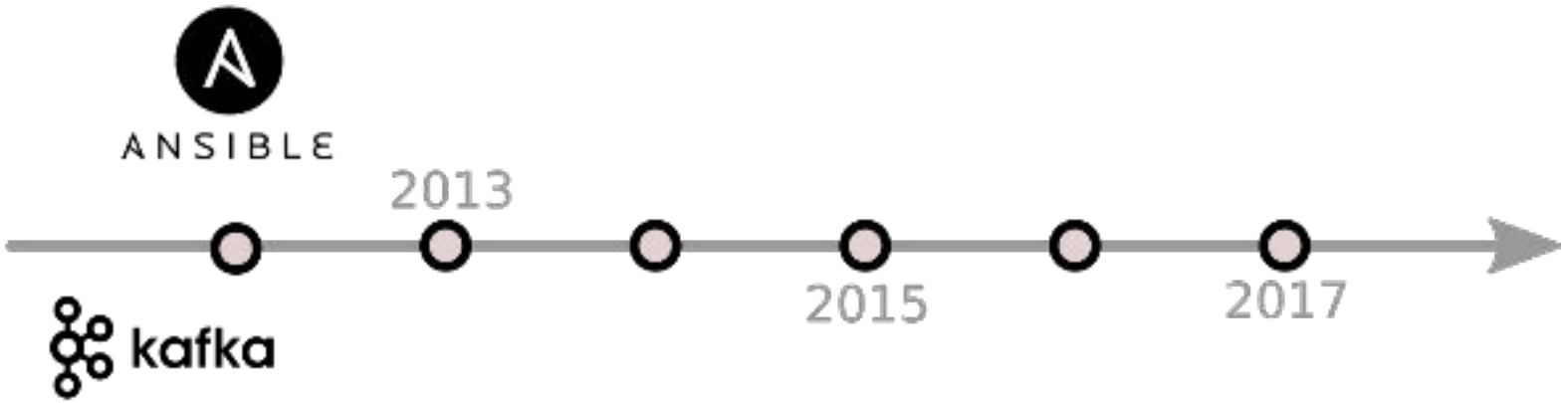
- maps roles to hosts - standardization
- written in YAML - human readable
- ideally in combination with VCS

Role

- combination of tasks
- featuring idempotency
- ideally in combination with VCS

```
---
#preflight confluent role
- name: Add open JDK repo
  apt_repository:
    repo: ppa:openjdk-r/ppa

- name: Install Java
  apt:
    name: "openjdk-8-jdk"
    update_cache: yes
...
```

A history of Application Orchestration



Automate Kafka deployment

- standardized
- idempotency and rolling updates
- infrastructure-as-code

Confluent Kafka Playbooks

Scope:

- Installs Confluent Platform packages
- Starts services using systemd scripts
- config options for plaintext, SSL, SASL_SSL

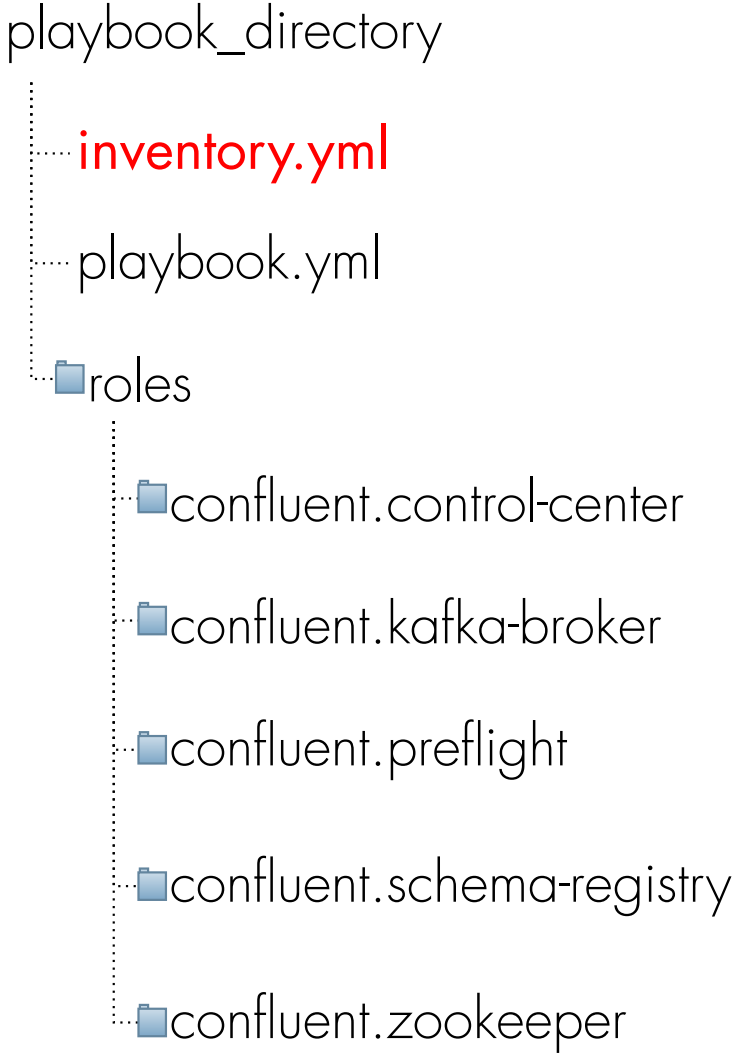
Services:

- ZooKeeper
- Kafka
- Schema Registry
- REST Proxy
- Confluent Control Center
- Kafka Connect (distributed mode)

https://docs.confluent.io/current/tutorials/cp-ansible/docs/index.html

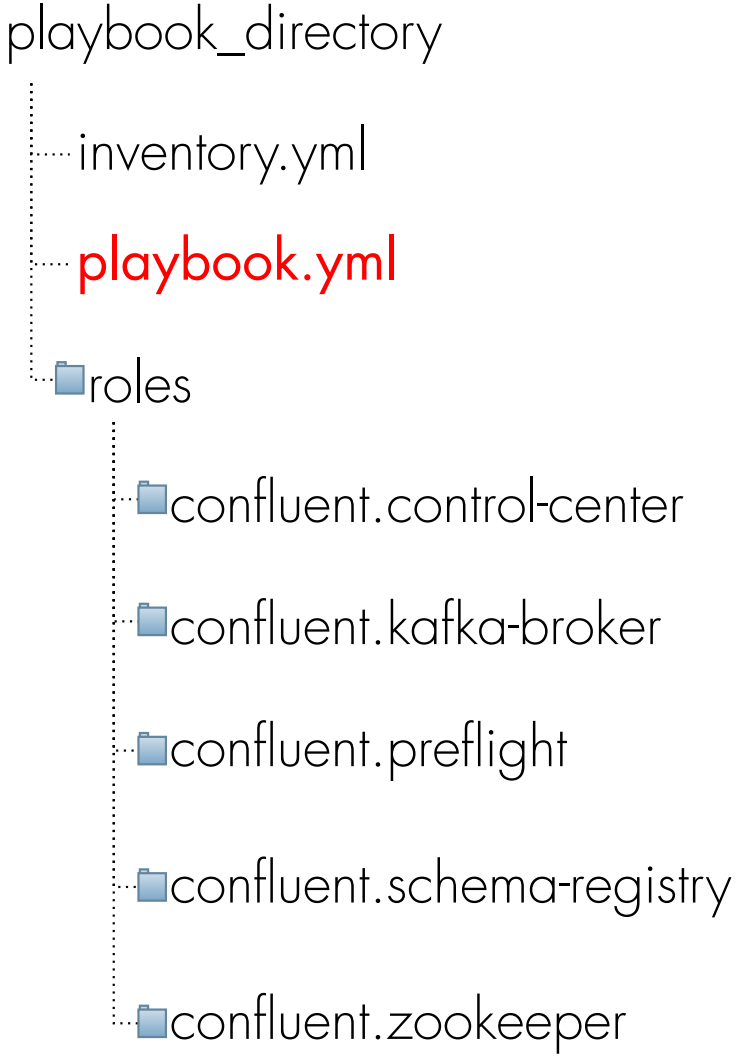
Confluent Kafka Playbooks - Setup

```
---
  zookeeper:
    hosts:
      host1:
        ansible_host: 13.12.2.1
      host2:
        ansible_host: 13.12.2.2
      host3:
        ansible_host: 13.12.2.3
  broker:
    hosts:
      host1:
        ansible_host: 13.12.2.1
      host2:
        ansible_host: 13.12.2.2
      host3:
        ansible_host: 13.12.2.3
```



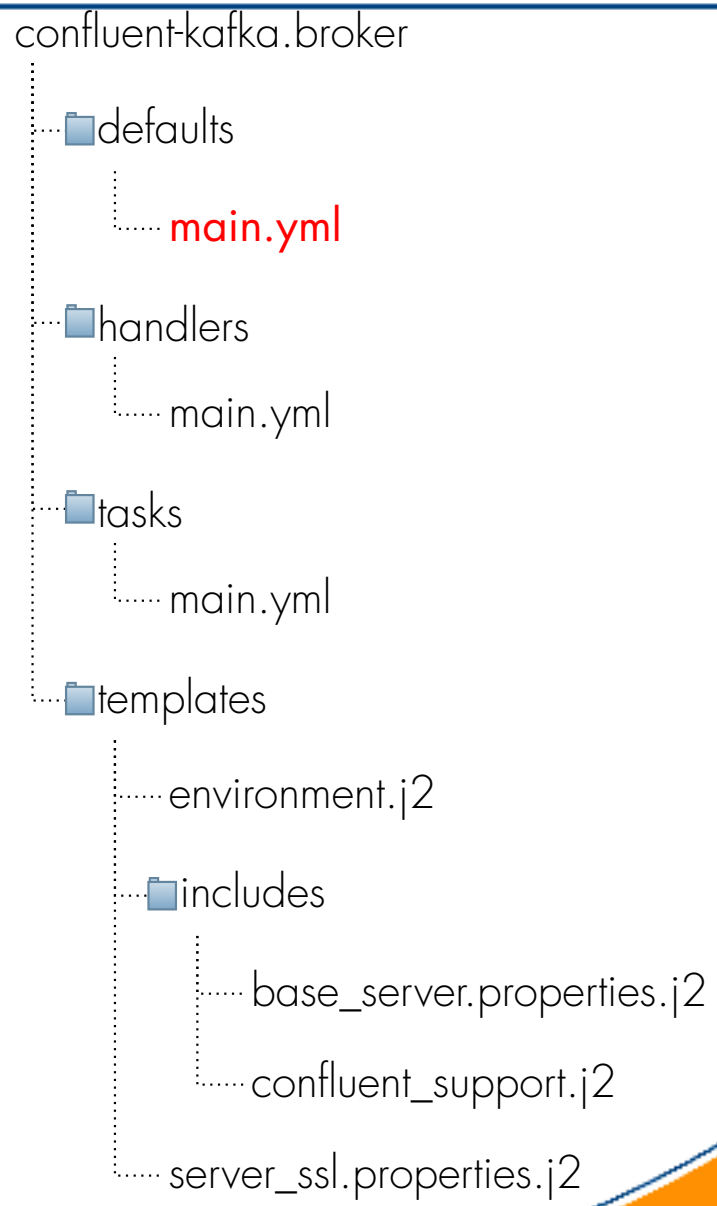
Map roles to servers

```
---
- hosts: preflight
  tasks:
  - import_role:
      name: confluent.preflight

- hosts: ssl_CA
  tasks:
  - import_role:
      name: confluent.ssl_CA

- hosts: zookeeper
  tasks:
  - import_role:
      name: confluent.zookeeper

- hosts: broker
  tasks:
  - import_role:
      name: confluent.kafka-broker

- hosts: schema-registry
  tasks:
  - import_role:
      name: confluent.schema-registry
```



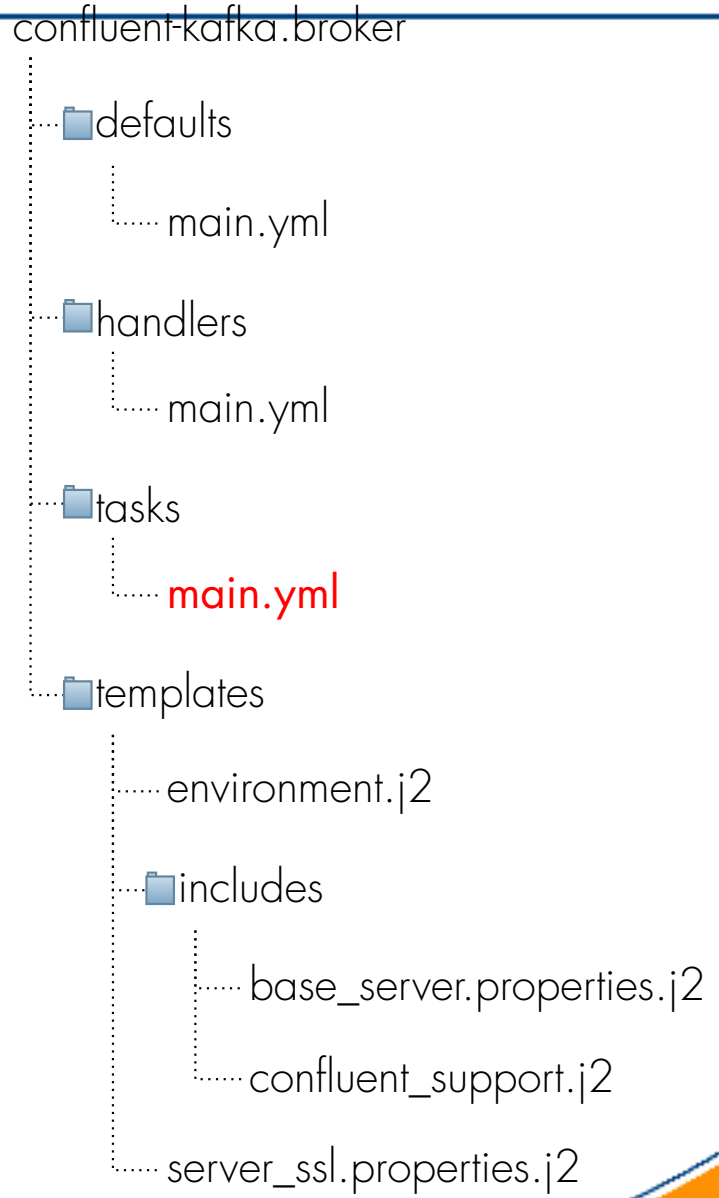
Separation of Data ...

```yaml
---
kafka:
  broker:
    user: "cp-kafka"
    group: "confluent"
    config_file: "/etc/kafka/server.properties"
    systemd_file: |
     "/usr/lib/systemd/system/kafka.service"
    service_name: "kafka"
    datadir:
      - "/var/lib/kafka/data"
    systemd:
      enabled: yes
      state: "started"
    environment:
      KAFKA_HEAP_OPTS: "-Xmx1g"
    config:
      group.initial.rebalance.delay.ms: 0
      log.retention.check.interval.ms: 300000
      num.partitions: 1
      num.recovery.threads.per.data.dir: 2
      offsets.topic.replication.factor: 3
      transaction.state.log.min.isr: 2
      zookeeper.connection.timeout.ms: 6000
      # [...] many more
```



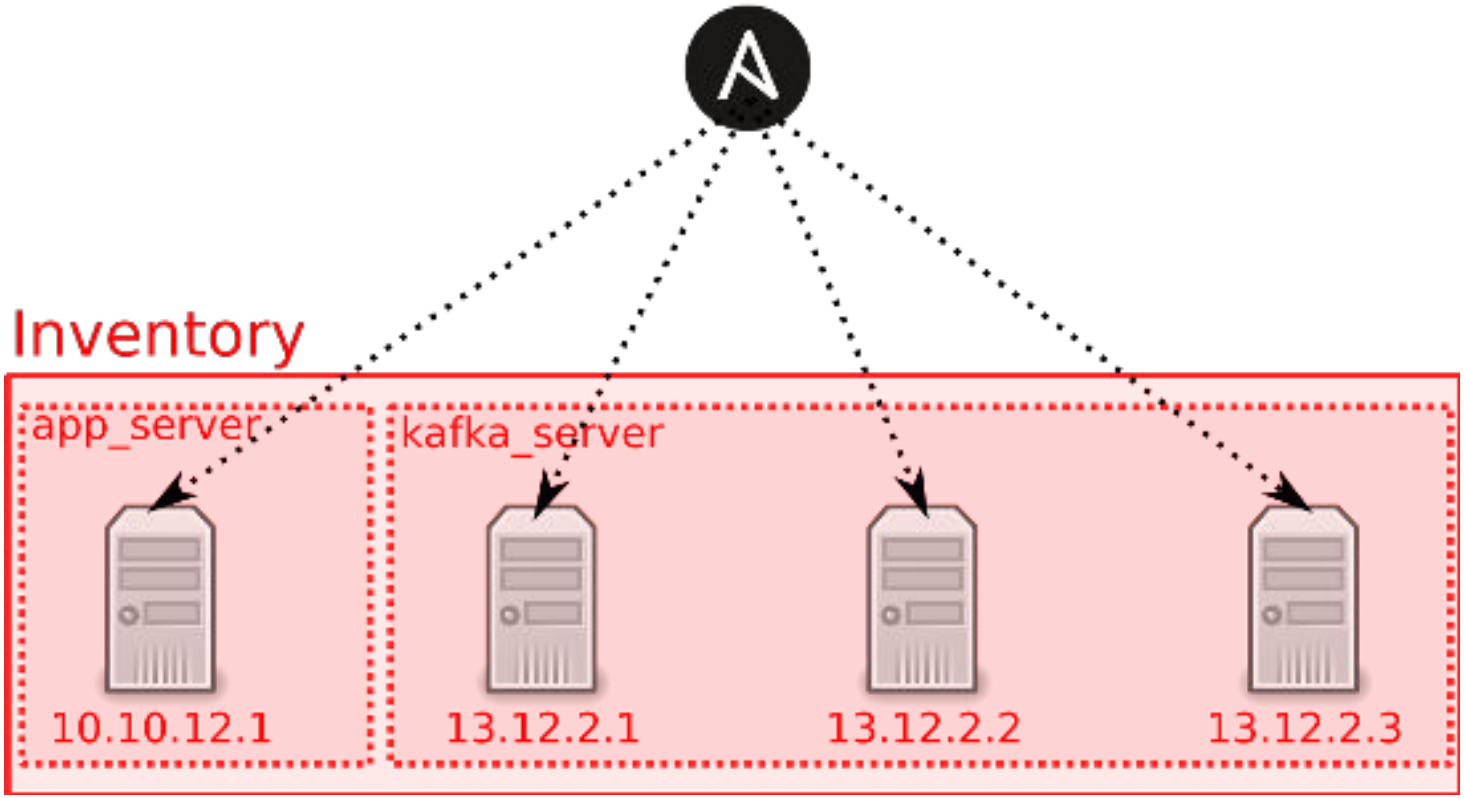
… and Code

```yaml
---
# [...] tasks to create user,group and dirs
- name: broker ssl config
  template:
    src: server_ssl.properties.j2
    dest: "{{kafka.broker.config_file}}"
    mode: 0640
    owner: "{{kafka.broker.user}}"
    group: "{{kafka.broker.group}}"
  when: security_mode == "ssl"
  notify:
    - restart kafka

- name: create systemd override file
  file:
    path: "{{kafka.broker.systemd_override}}"
    owner: "{{kafka.broker.user}}"
    group: "{{kafka.broker.group}}"
    state: directory
    mode: 0640

- name: broker configure service
  systemd:
    name: "{{kafka.broker.service_name}}"
    enabled: "{{kafka.broker.systemd.enabled}}"
    state: "{{kafka.broker.systemd.state}}"
```

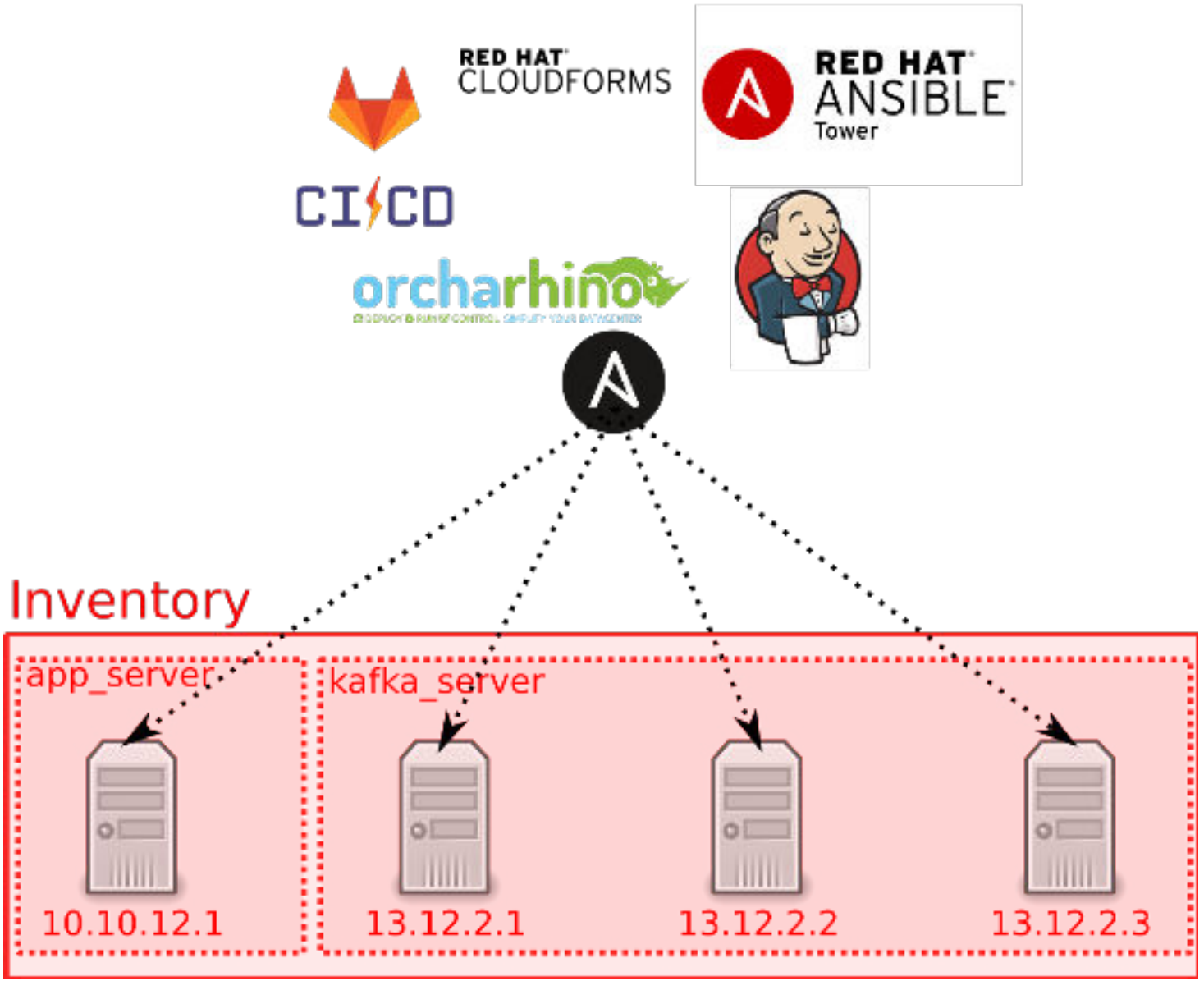


Use it by hand ...

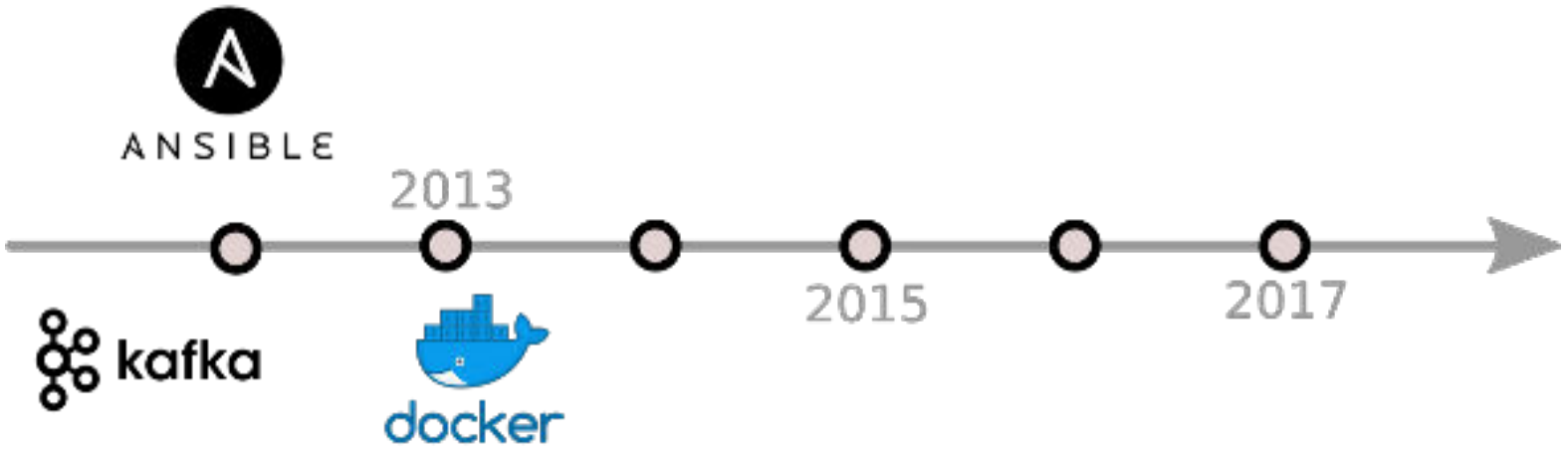
```
# Check and See Changes
ansible-playbook -i inventory.yml all.yml --check --diff

# Run Initial Setup/Update
ansible-playbook -i inventory.yml all.yml
```



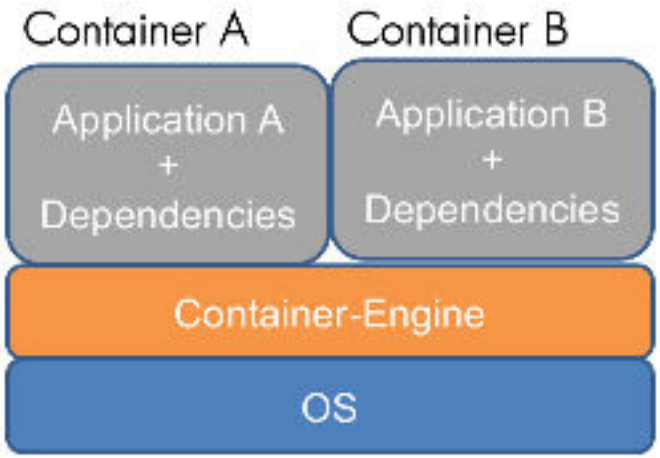
… or by CI/CD operator



A history of Application Orchestration



Dockerization of applications



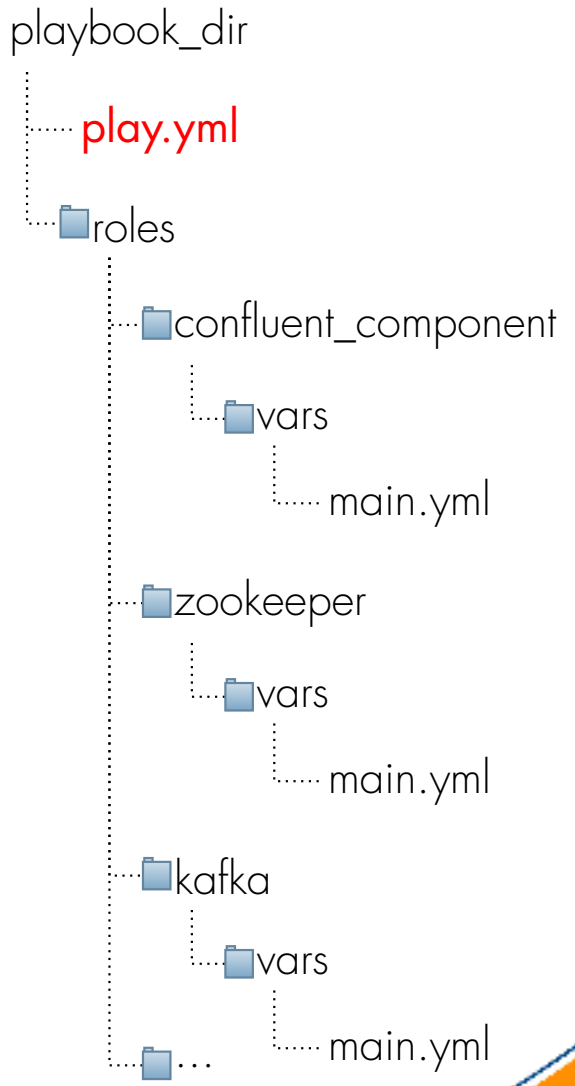
- portability
- scalability
- (security)
- efficiency
- reproducibility

Ansible and Docker

Ansible module for managing Docker container
→ even more generic roles

Generic role!

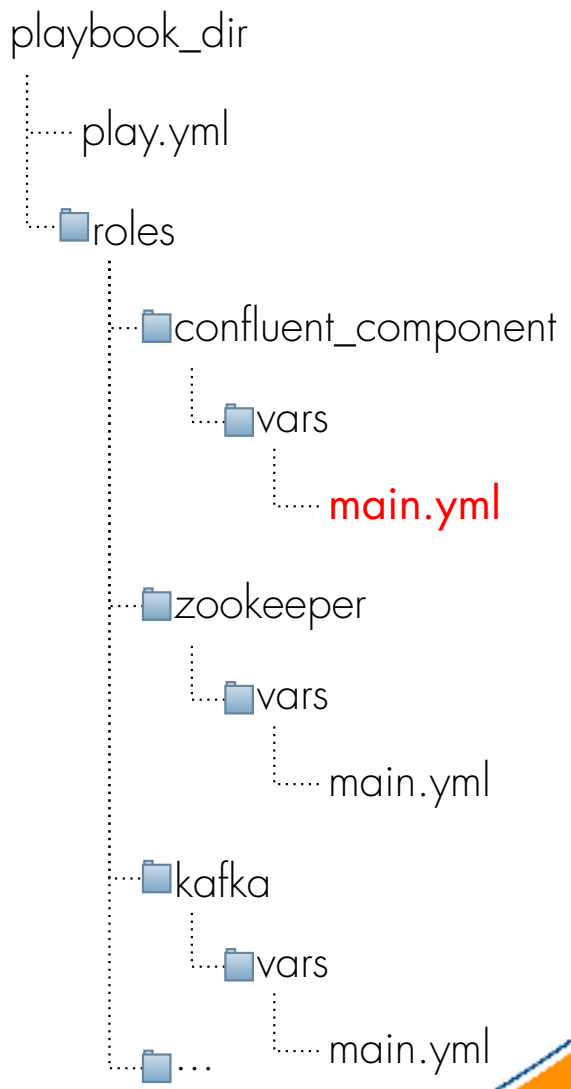
```
---
- hosts: zookeeper
  tasks:
    - name: Deploy Zookeeper
      include_role:
        name: confluent_component
        vars_from: zookeeper

- hosts: kafka
    - name: Deploy Kafka
      include_role:
        name: confluent_component
        vars_from: kafka

- hosts: control-center
    - name: Deploy Control-Center
      include_role:
        name: confluent_component
        vars_from: control-center
```



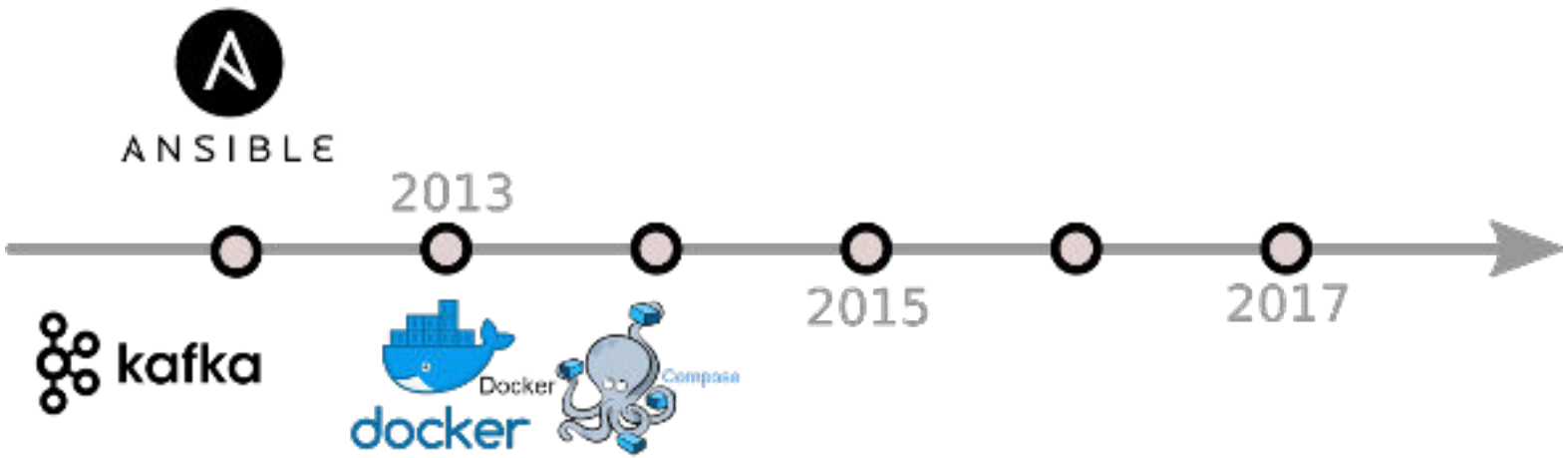
Generic docker role, separate data and code

```yaml
---
- name: "Start Docker-Container"
  docker_container:
    name: "{{ kafka_component_name }}"
    image: "{{ kafka_component_container_image }}"
    state: "{{ kafka_component_container_state }}"
    restart: "{{ config_changed.changed }}"
    published_ports: "{{ published_ports }}"
    restart_policy: "{{ container_restart_policy }}"
    env_file: "{{ kafka_component_env_file }}"
    volumes: "{{ kafka_component_volumes }}"
```

```yaml
---
kafka_component_name: "zookeeper"
image: "confluentinc/cp-kafka"
published_ports:
  - 12888:2888
  - 13888:3888
```



A history of Application Orchestration



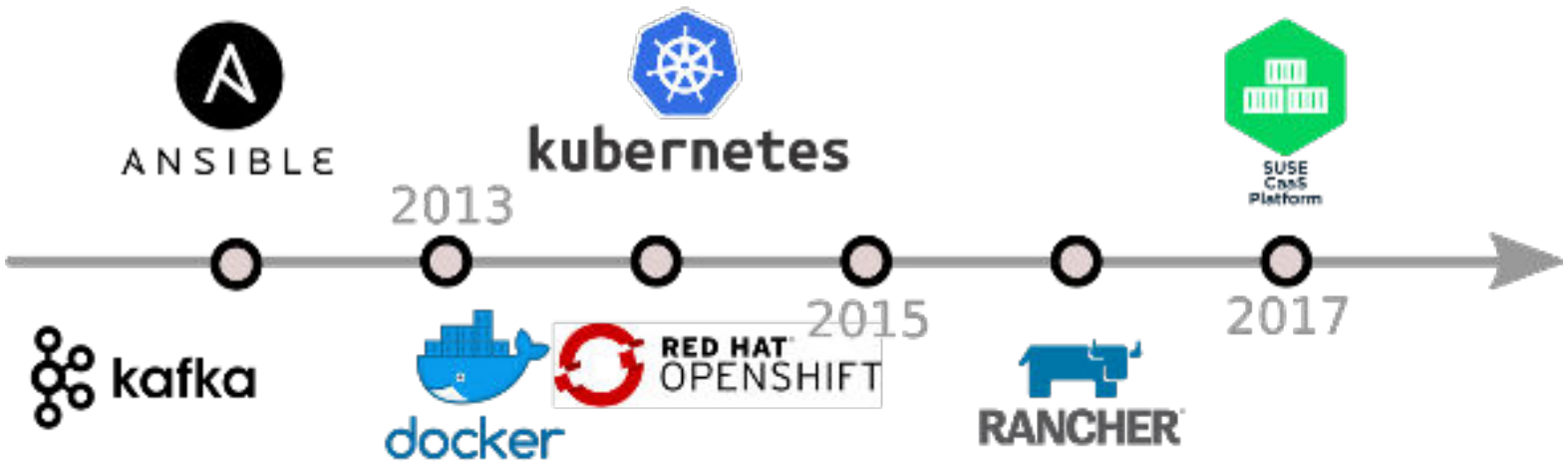
Docker Compose - Multi container deployment

```yaml
---
version: '2'
services:
  zookeeper:
    image: "confluentinc/cp-zookeeper:latest"
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
      ZOOKEEPER_TICK_TIME: 2000
  kafka:
    image: "confluentinc/cp-kafka:latest"
    depends_on:
      - "zookeeper"
    ports:
      - 9092:9092
    environment:
      KAFKA_BROKER_ID: 1
      KAFKA_ZOOKEEPER_CONNECT: "zookeeper:2181"
    #  [...] more kafka broker settings
```

Use Docker Compose

```yaml
---
- hosts: kafka-server
  tasks:
    - name: "Spin up Kakfa-Cluster"
      docker_compose:
        project_src: "cp_kafka"
        state: absent
      register: output

    - name: "Ensure Stack is running"
      assert:
        that:
          - kafka.cp_kafka_kafka_1.state.running
          - zookeeper.cp_kafka_zookeeper_1.state.running
```

A history of Application Orchestration



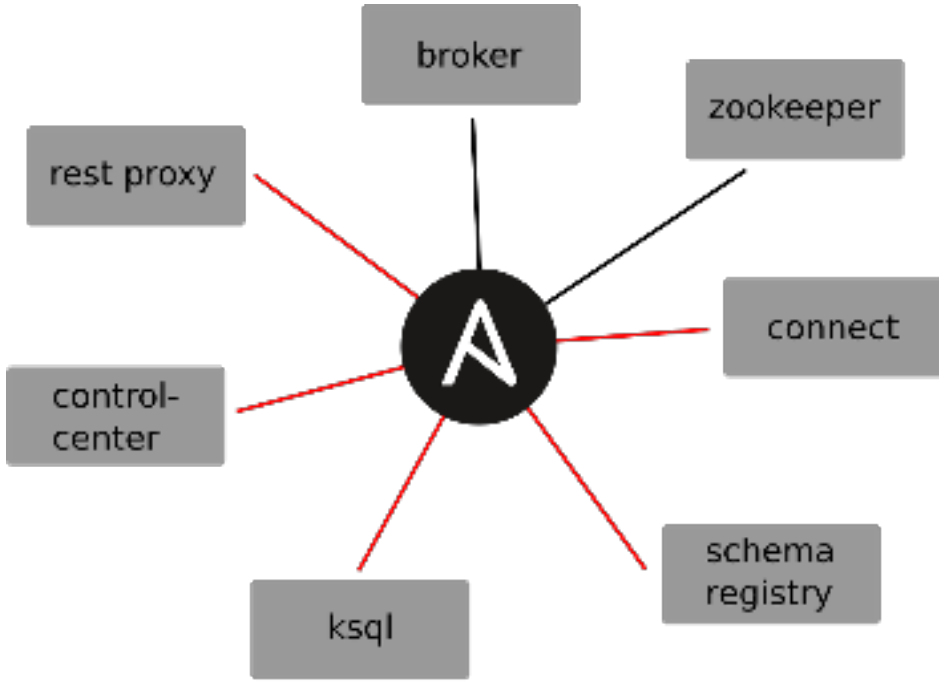
Use Helm

```
---
- hosts: kafka-server
  tasks:
    - name: "Install Confluent-Operator with Helm"
      command: >
        helm install \
        -f ./providers/private.yaml \
        --name kafka-operator \
        --namespace kafka-space \
        --set operator.enabled=true \
        --set rbac.Enabled=true \
        ./confluent-operator
      when:  "'kafka-operator' not in helm_installed"
```

What else?

- Ansible Helm Module
- Ansible Operator (Openshift)

Manage Kafka with Ansible



Manage Kafka - Topics

```yaml
---
- name: "create topic"
  kafka_lib:
    resource: 'topic'
    name: 'test'
    partitions: 2
    replica_factor: 1
    options:
      retention.ms: 574930
      flush.ms: 12345
    state: 'present'
    zookeeper: >
      "{{ zookeeper_ip }}:2181"
    bootstrap_servers: >
      "{{ kafka_ip_1 }}:9092, {{ kafka_ip2 }}:9092"
    security_protocol: 'SASL_SSL'
    sasl_plain_username: 'username'
    sasl_plain_password: 'password'
    ssl_cafile: '{{ content_of_ca_cert_file_or_path_to_ca_cert_file }}'
```

github.com/StephenSorriaux/
ansible-kafka-admin

→ Ansible Module for Topics and ACls
→ No SSH Connection to remote broker needed

Manage Kafka - Topics

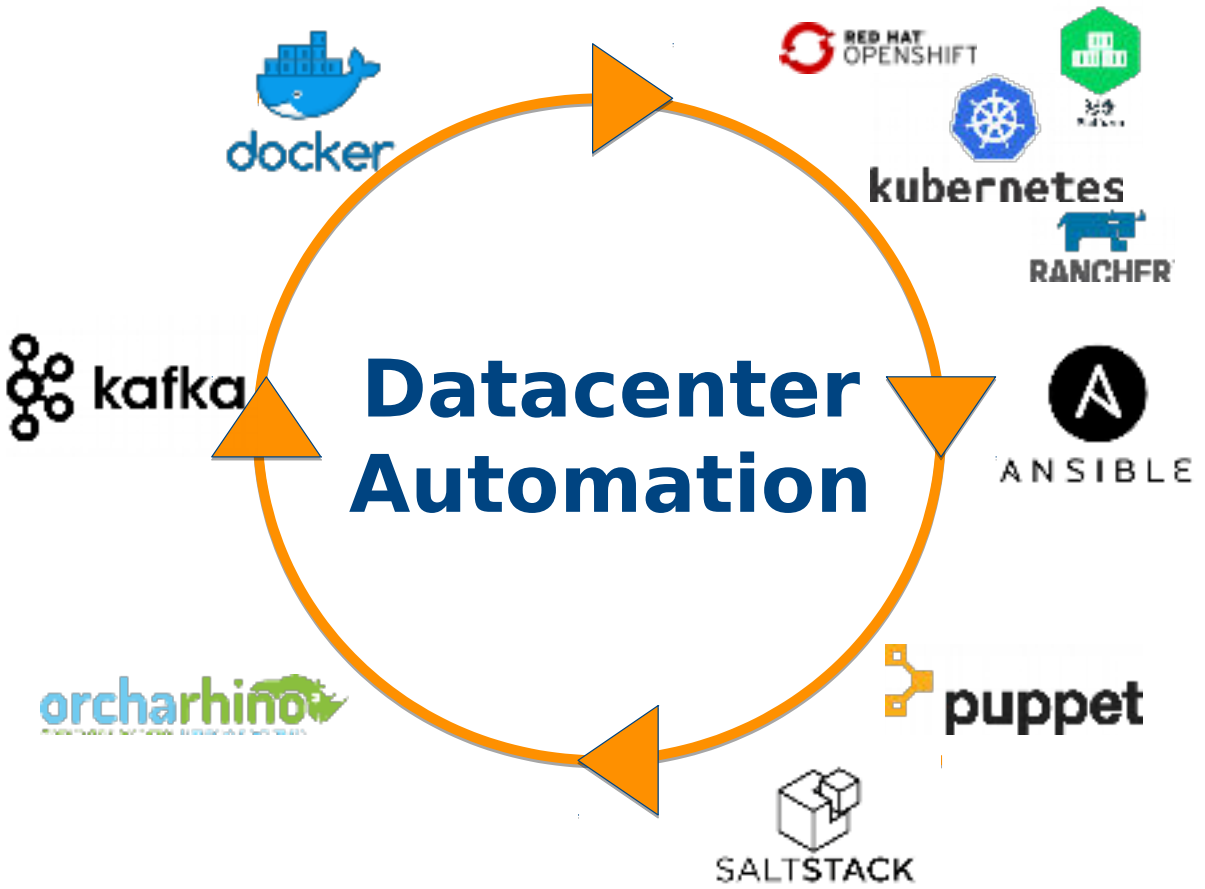
```
---
# Definition of topic
topic:
 name: "test"
 partitions: 2
 replica_factor: 1
 configuration:
  retention.ms: 574930
  flush.ms: 12345

---
- name: "Get topic information"
 uri:
  url: "{{ 'kafka_rest_proxy_url' + ':8082/topics/' + topic.name }}"
 register: result

- name: "Create new topic"
 command: "{{ 'kafka-topics --zookeeper ' + zookeeper +
                ' --create' +
                ' --topic ' + topic.name +
                ' --partitions ' + topic.partitions +
                ' --replication-factor ' + topic.replica_factor +
               topic.configuration }}"
```

- Rest Proxy cannot create topics
- manual idempotency
- access to one kafka broker

whoami

Bernhard Hopfenmüller
Senior IT Consultant @ ATIX AG

hopfenmueller@atix.de
github.com/Fobhep
Twitter: @fobhep
IRC: Fobhep


docker
RED HAT OPENSHIFT
kubernetes
RANCHER
kafka
Datacenter Automation
ANSIBLE
orcharhino
puppet
SALTSTACK