What is Ansible?

- ☐ Configuration Management Tool
- ☐ Orchestration Tool
- ☐ Automation Tool

Performance Testing Kafka

Automating CLI Tools with Ansible

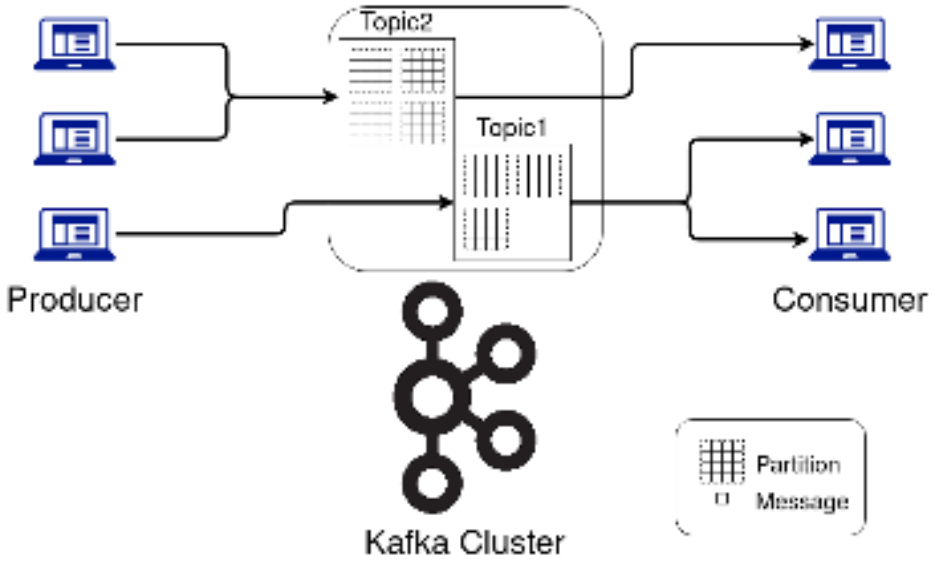
Bernhard Hopfenmüller

11. Mai 2021

whoami

```
ok: [localhost] => {
    "me": {
        "name": "Bernhard Hopfenmüller",
        "occupation": "Senior Consultant",
        "working with": "Ansible, Kafka,..",
          }
        }
```

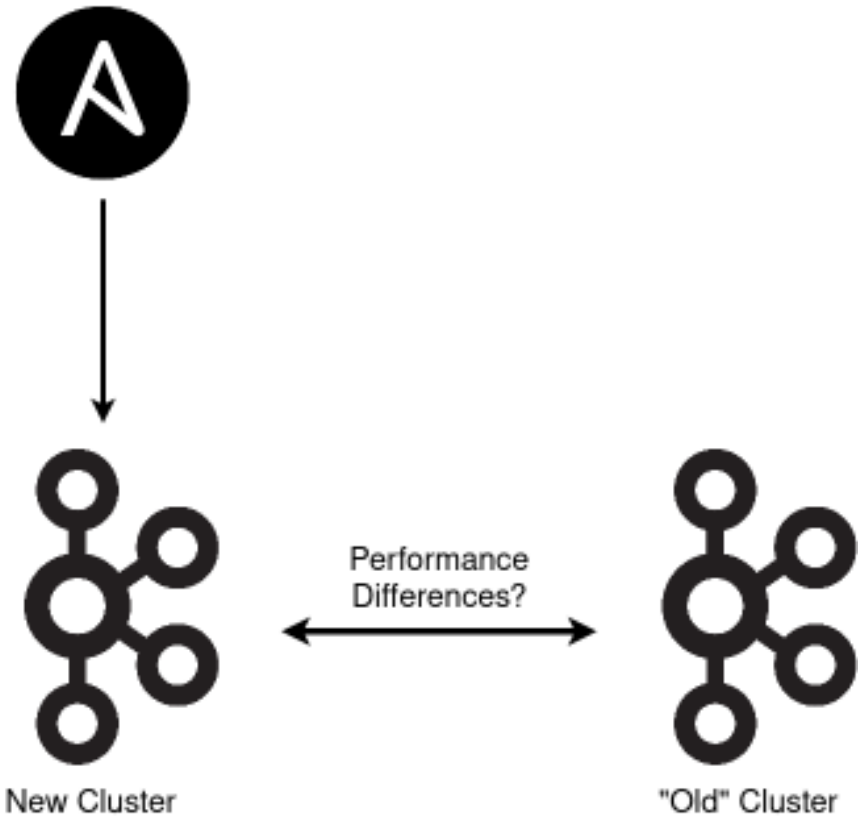
Kafka Cluster



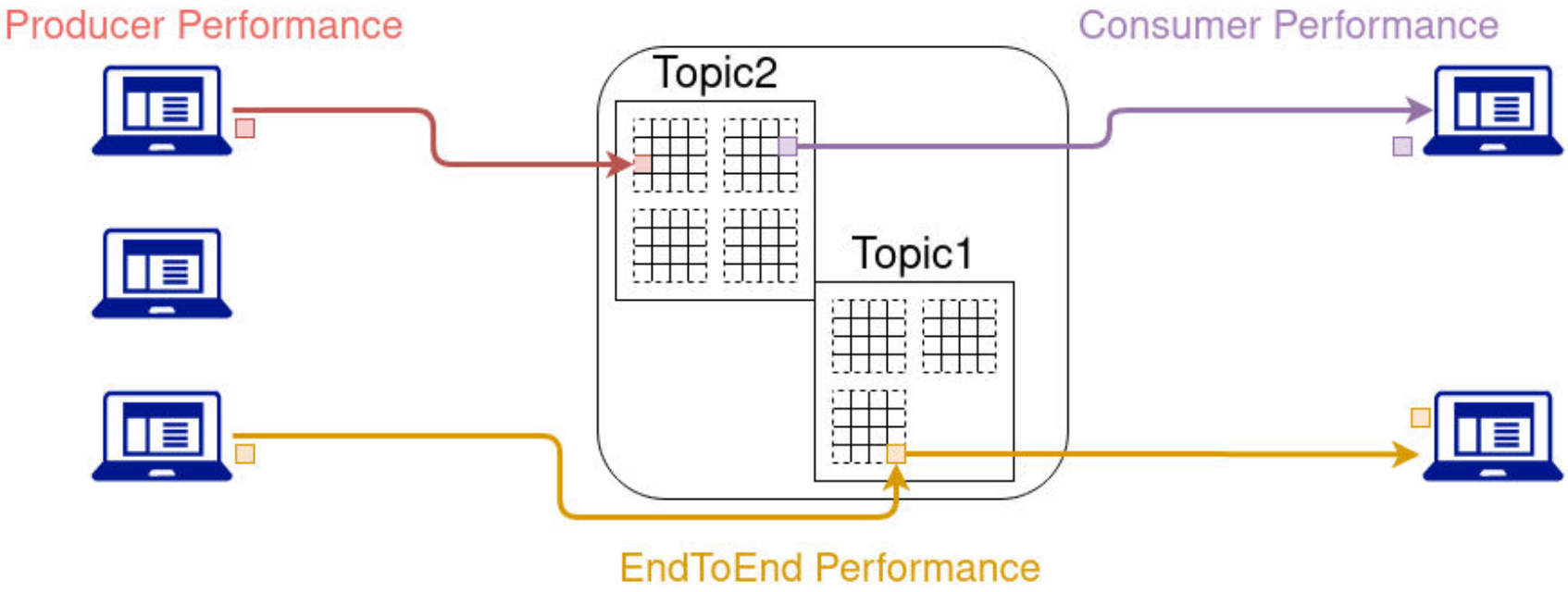
- distributed log
- messages stored in topics
- topics consist of partitions
- partitions are replicated

https://www.heise.de/select/ix/2019/4/1553935521978043

Starting point



How do we measure cluster perfomance?



CLI Based Clients

kafka-producer-perf-test --producer.config --num-records ...

kafka-consumer-perf-test --consumer.config --num-records ...

kafka-run-class kafka.tools.EndToEndLatency <num_of_records> <config-file> ...

Test conditions

→ Tests must run from “external” clients

→ Tests must be easily reproducible

→ Tests must run parallel

→ Results should be easily comparable

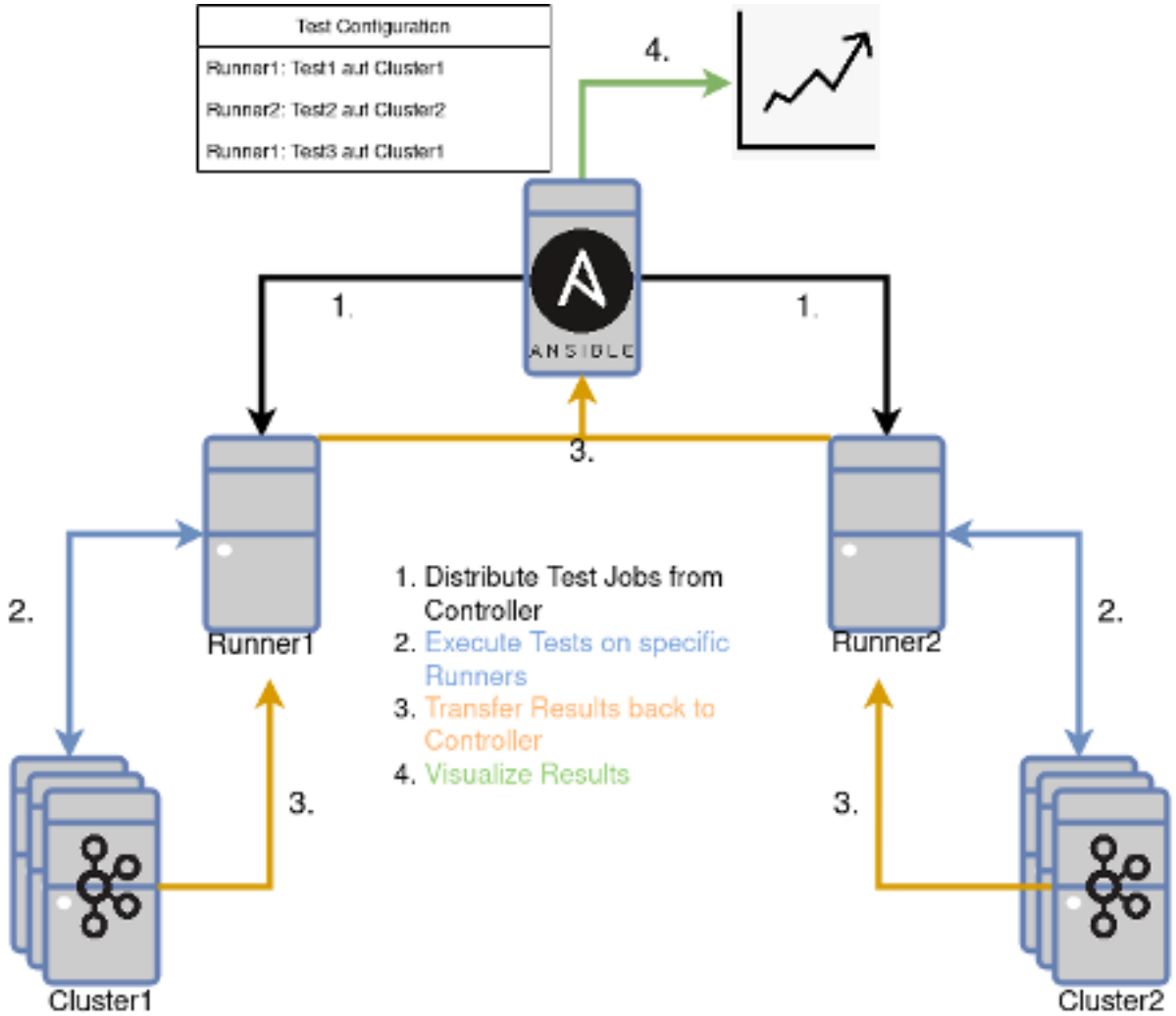
Why not use Ansible + CLI Tools

→ less effort than writing “proper” Kafka Clients

→ Ansible already “set up”

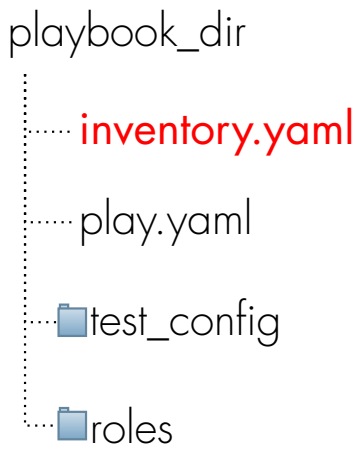
→ Ansible can execute tasks parallel on different hosts

Test Workflow



Inventory

```
---
all:
  children:
    runner:
      hosts:
        runner_cluster1:
        runner_cluster2:
```

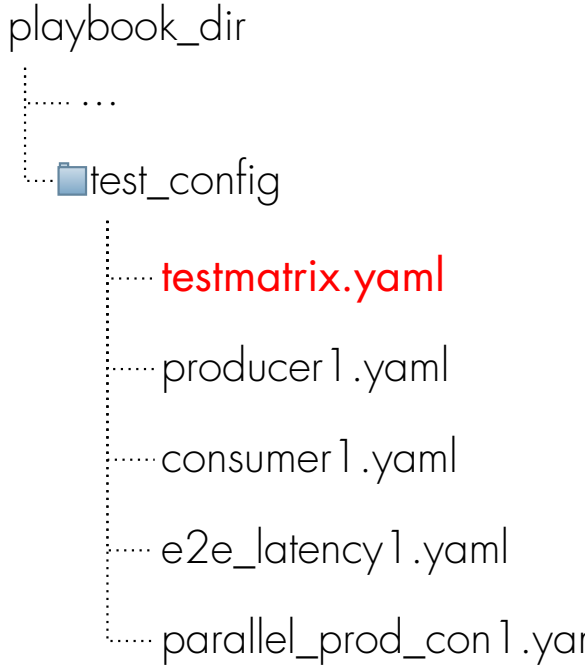


Playbook

```yaml
---
- hosts: runner
  gather_facts: false # facts of runner nodes don't matter
  strategy: free  # each host as fast as possible
  tags:
    - run_tests
  pre_tasks:  # use time of controller node for timestamps
    - name: Get epoch
      setup:
        filter: ansible_date_time
      delegate_facts: true
      delegate_to: localhost
  roles:
    - perf_test_kafka    # run actual performance tests

- hosts: runner
  gather_facts: false
  strategy: free
  tags:
    - pull_data
  roles:
    - pull_data      # move data back to controller node

- hosts: localhost
  tags:
    - visualize
  roles:
    - visualize_results # visualize data on controller node
```

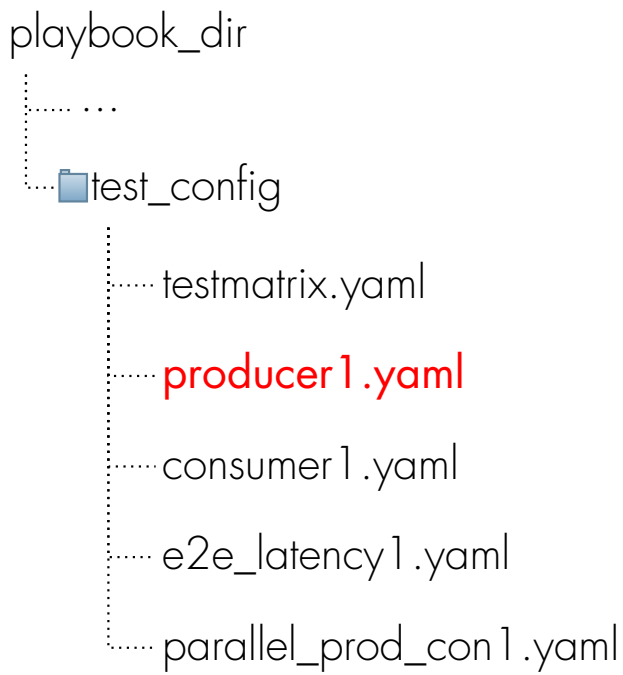
Test Matrix

```
---
runner_cluster1:
  client: client_cluster1.config
  testsequence:
    - test: producer1.txt
    - test: consumer1.txt
    - test: e2e_latency.txt
    - test: parallel_prod_con.txt

runner_cluster2:
  client: client_cluster2.config
  testsequence:
    - test: producer1.txt
    - test: consumer1.txt
    - test: e2e_latency.txt
    - test: parallel_prod_con.txt

visualize:
  full:
    - runner_cluster1 & runner_cluster2
```



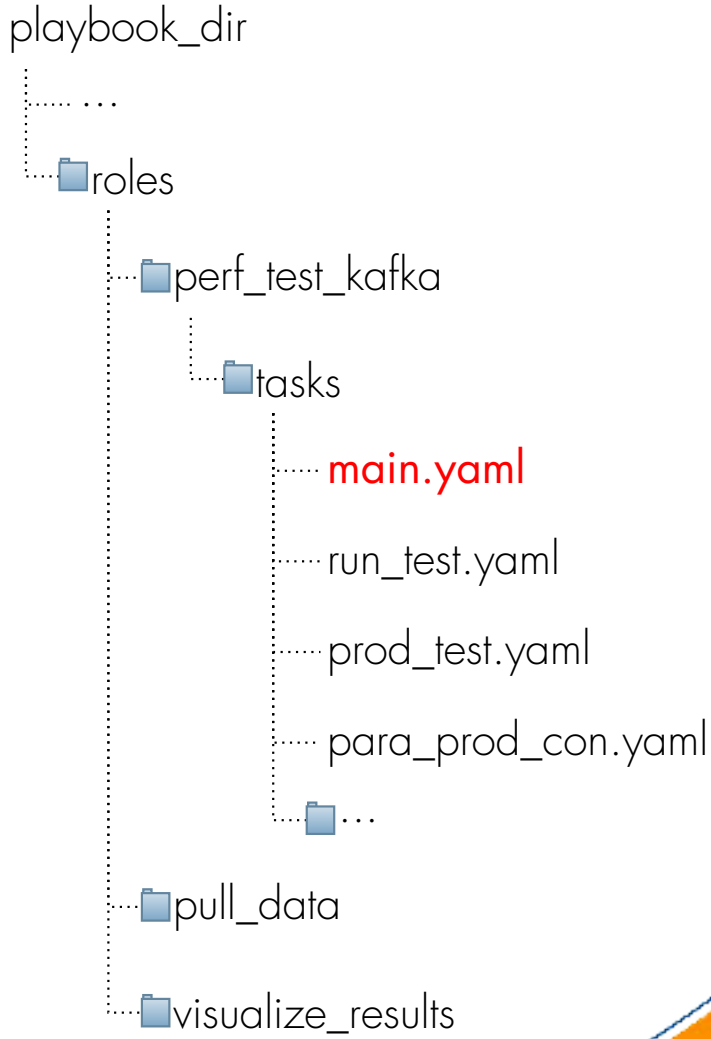
Specific Test Definition

```
---
#NOT THE WHOLE FILE
test_type: producer
num_records: 1000000
record_size: 15360
topic_name: producer.test.topic

#Will create roughly 1Mio x 15KiB = 15GB of data
```



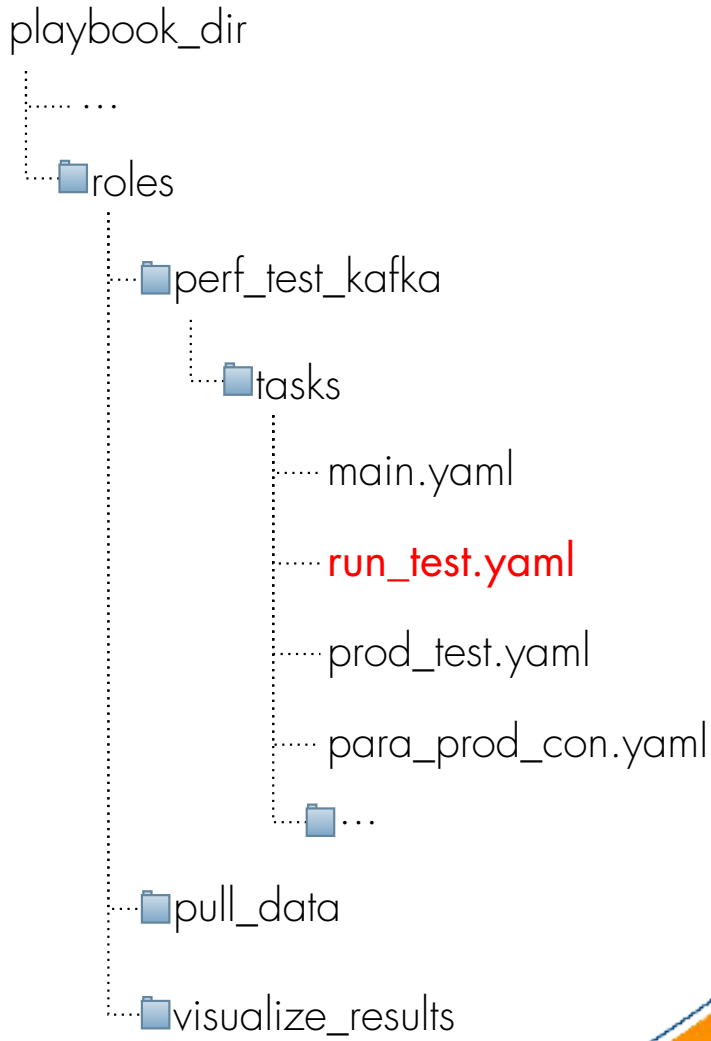
Performance Test Role

```
---
#NOT THE WHOLE FILE
- name: Collect test matrix
  set_fact:
    test_matrix: |
    "{{ lookup('template', 'test.matrix') | from_yaml }}"

- name: Run test matrix
  include_tasks: "run_test.yaml"
  loop: "{{ test_matrix[inventory_hostname]['testsequence'] }}"
  loop_control:
    loop_var: curr_test
  when: inventory_hostname in test_matrix
  register: matrix_run

- name: Show failed tests
  debug:
    msg: |
    "The following tests failed and
    need to be rerun {{ failed_tests }}!"
  when: |
  matrix_run is not skipped and failed_tests | length > 0
```



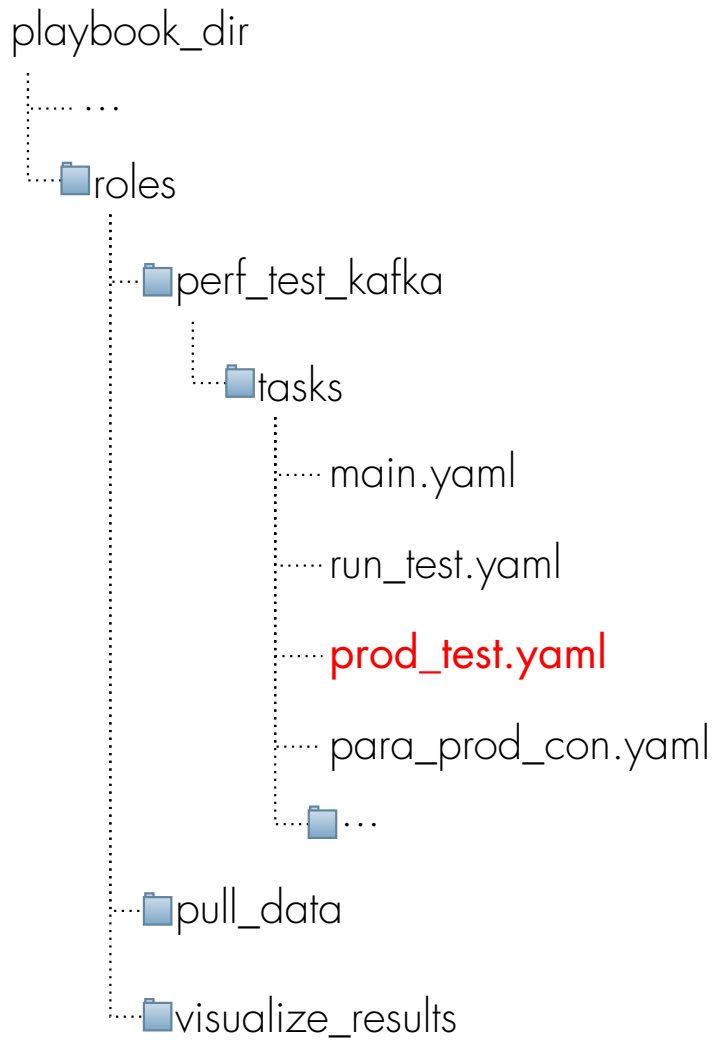
Performance Test Role

```
---
#NOT THE WHOLE FILE
- name: Collect current test
  set_fact:
    curr_test_config: |
    "{{ lookup('template', curr_test.test~'.yaml') | from_yaml }}"

- name: Copy client config to runner
  copy:
    src: "{{ test_config_dir ~ client }}"
    dest: "{{ test_folder }}/{{ client }}"

- name: Include appropriate test
  include_tasks: "{{ curr_test_config.test_type }}_test.yaml"
```



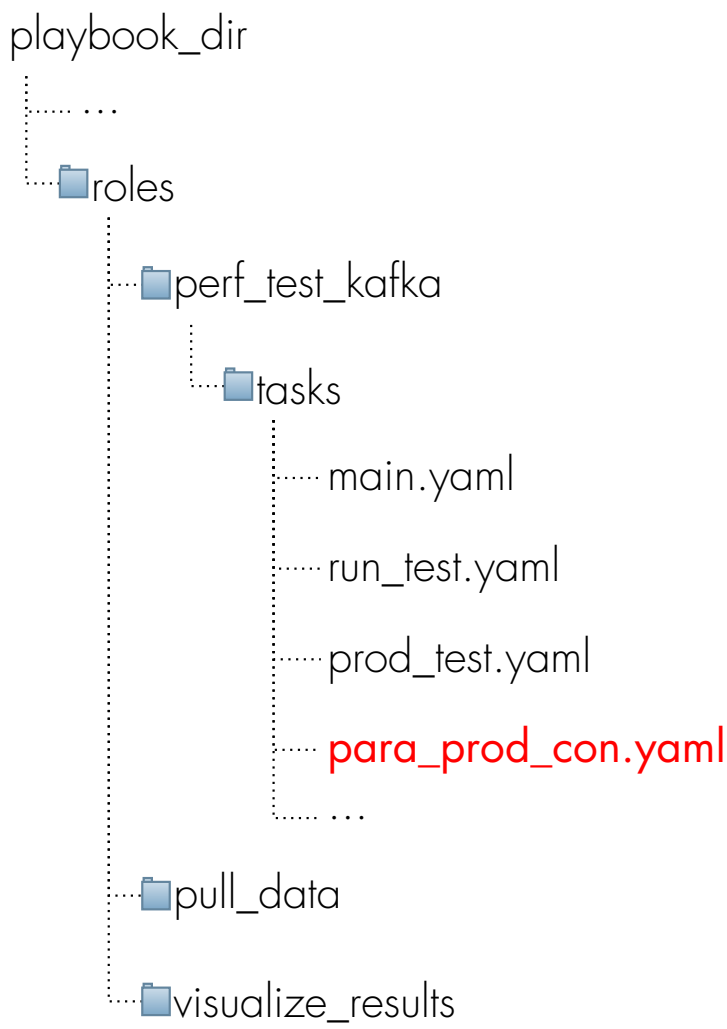
Producer Test

```
---
#NOT THE WHOLE FILE
- name: Create topic before producing
  include_tasks: create_topics.yaml

- block:
    - name: Start Producer Test
      shell: |
      "( time kafka-producer-perf-test \
        --topic {{ topic_prefix }}{{ curr_test_config.topic_name }} \
        --producer.config {{ test_folder }}/{{ client }} \
        --num-records {{ curr_test_config.num_records }} \
        --record-size {{ curr_test_config.record_size }} ) |
        & tee {{ testoutput }}.txt"
      args:
        creates: "{{ testoutput }}.txt"
      register: command_result
      failed_when: "'Exception' in command_result.stdout"
  rescue:
    - name: Copy failed test to fail directory
      copy:
        remote_src: true
        src: "{{ testoutput }}.txt"
        dest: "/tmp/failed_tests/{{ testoutput }}.txt"

    - name: Add Test to failed test array
      set_fact:
        failed_tests: "{{ failed_tests + [ curr_test.test ] }}"
```



Consumer and Producer parallel on Runner

```
---
#NOT THE WHOLE FILE
- block:
    - name: Start Producer Test
      shell: "( time kafka-producer-perf-test ...")
      args:
        creates: "{{ testoutput }}_prod.txt"
      async: 5000
      register: producer_task
      poll: 0

    - name: Run Consumer Test
      shell: "( time kafka-consumer-perf-test ...")
      args:
        creates: "{{ testoutput }}_con.txt"
      async: 5000
      poll: 0
      register: consumer_task

    - name: Check the run status for producer
      async_status:
        jid: "{{ producer_task.ansible_job_id }}"
      register: job_result
      until: job_result.finished
      retries: 5500

    - name: Check the run status for consumer
      async_status:
        jid: "{{ consumer_task.ansible_job_id }}"
      register: job_result
      until: job_result.finished
      retries: 5500
```



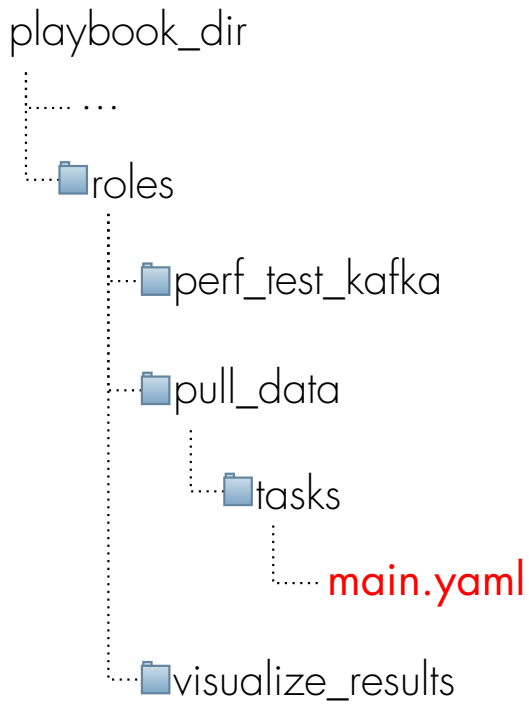
Output Files

Example: Producer Performance Test

```
#NOT WHOLE FILE
501 records sent, 100.0 records/sec (0.10 MB/sec), 9.3 ms avg latency, 240.0 ms max latency.
500 records sent, 99.8 records/sec (0.10 MB/sec), 13.7 ms avg latency, 218.0 ms max latency.
500 records sent, 99.9 records/sec (0.10 MB/sec), 5.4 ms avg latency, 132.0 ms max latency.

90000 records sent, 99.999444 records/sec (0.10 MB/sec), 8.33 ms avg latency,
579.00 ms max latency, 3 ms 50th, 37 ms 95th, 117 ms 99th, 194 ms 99.9th.

real  15m1.212s
user  0m45.827s
sys 0m14.397s
```

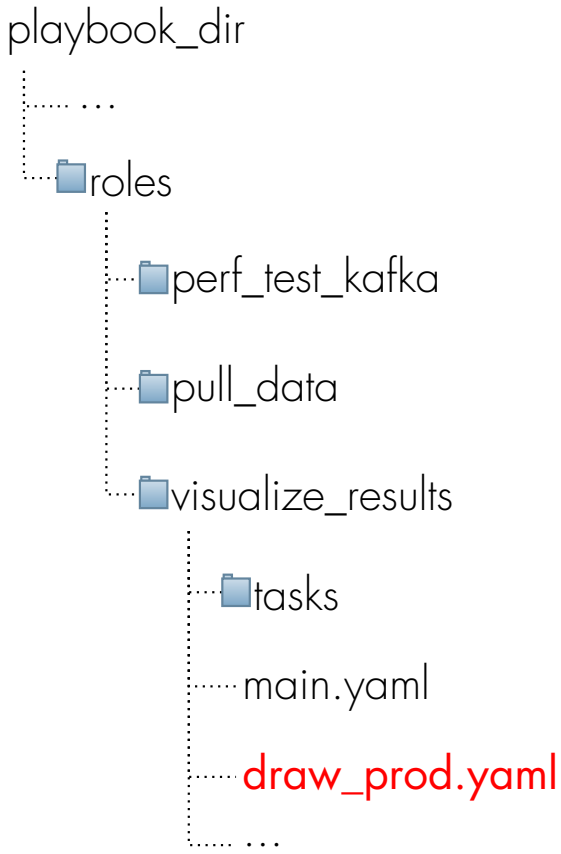
Pull Data

```
---
- name: Copy results back to ansible controller
  synchronize:
    recursive: true
    mode: pull
    src: "{{ results_folder }}"
    dest: results/
    rysnc_path: /bin/rsync

- name: Copy failed back to ansible controller
  synchronize:
    recursive: true
    mode: pull
    src: "{{ failed_folder }}"
    dest: results/
    rysnc_path: /bin/rsync
```

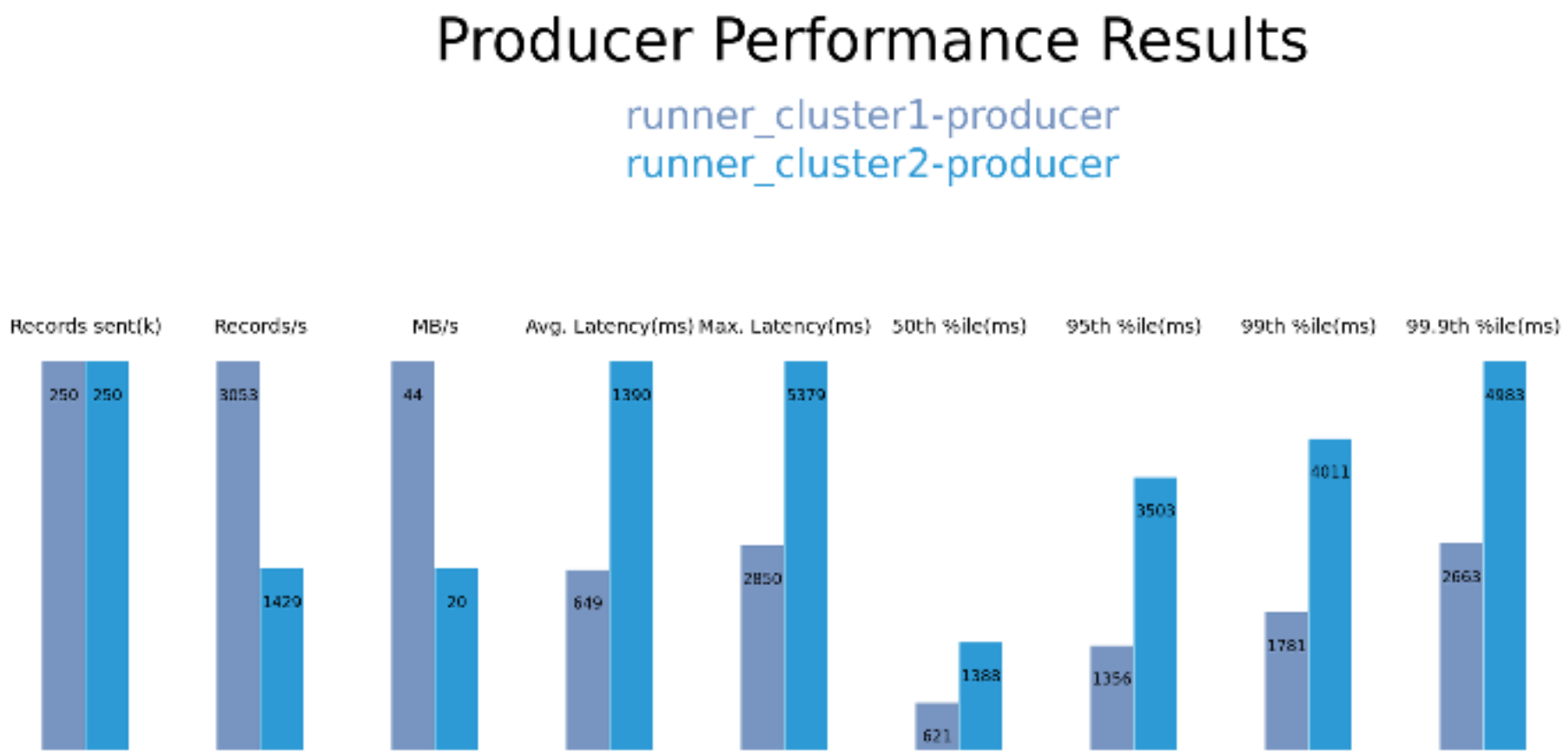


Visualize Data

```
---
- name: Sanitize command for split
  set_fact:
    first_data: "{{ item.split('&')[0].strip() ~ '.txt' }}"
    second_data: "{{ item.split('&')[1].strip() ~ '.txt' }}"

- name: Run producer visualization
  command: "./dvp.sh {{ first_data }} {{ second_data }}"
  args:
    chdir: results/results/
```



Example Visualization



What is Ansible?

- ✓ Automation Tool
- ✓ Configuration Management Tool - but not primarily
- ✓ Orchestration Tool - in combination with orcharhino,Foreman, Satellite, Tower, etc

ping me

@fobhep
hopfenmueller@atix.de
XING