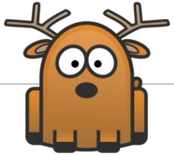
Automating Distributed Tasks with Ansible in the Cloud

Büsra Köken
Cloud System Developer



Hello!

*I am **Büsra Köken***

Cloud System Developer at Ericsson

You can find me at @busrakoken

Outline

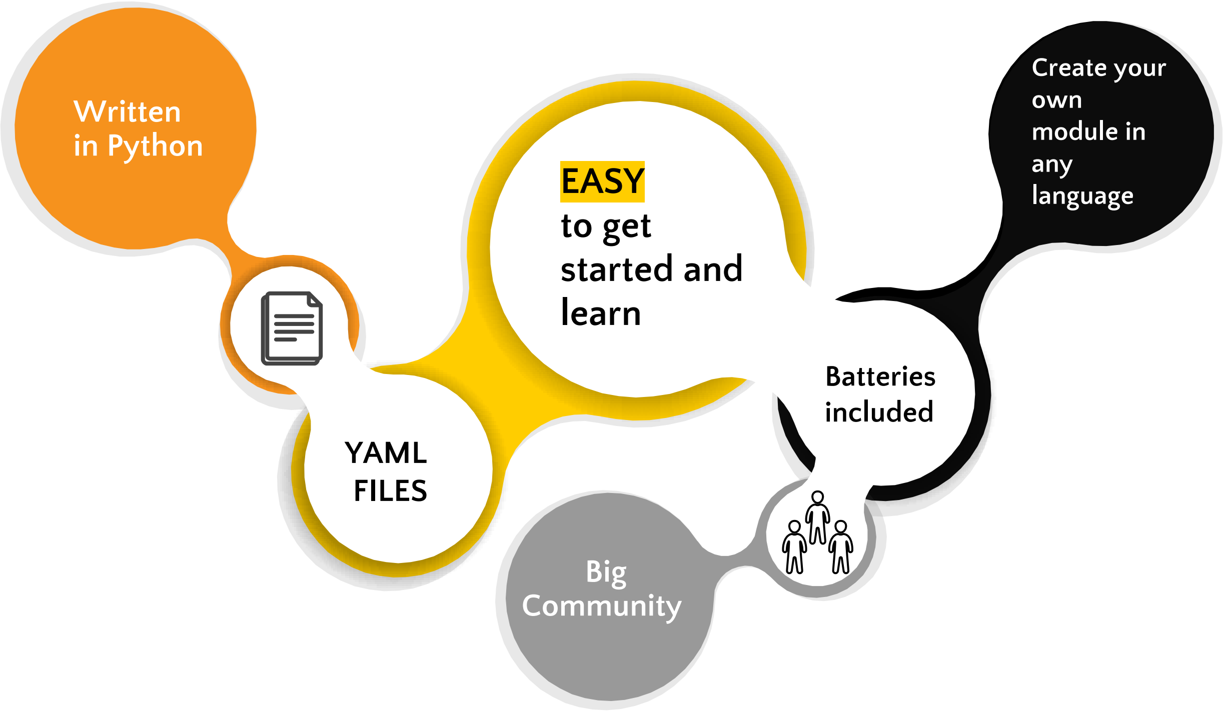
- Automation tools
- Ansible, how it works
- Case study explanation
- DEMO
- Summary and Q&A

Automation

IT Automation Tools

CFEngine	Puppet	Chef	Ansible
1993	2005	2009	2012

Why Ansible?



Written in Python

EASY to get started and learn

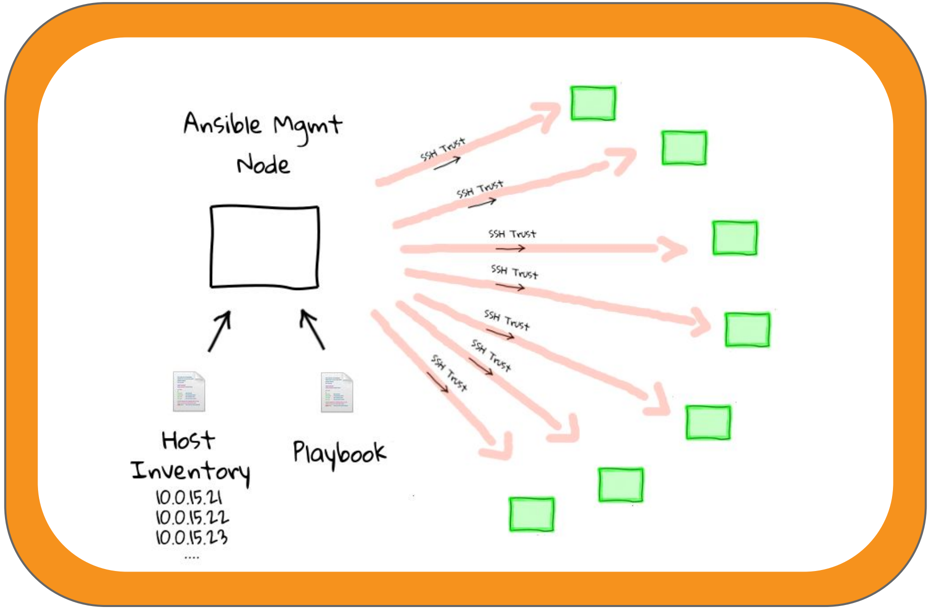
Create your own module in any language

YAML FILES

Batteries included

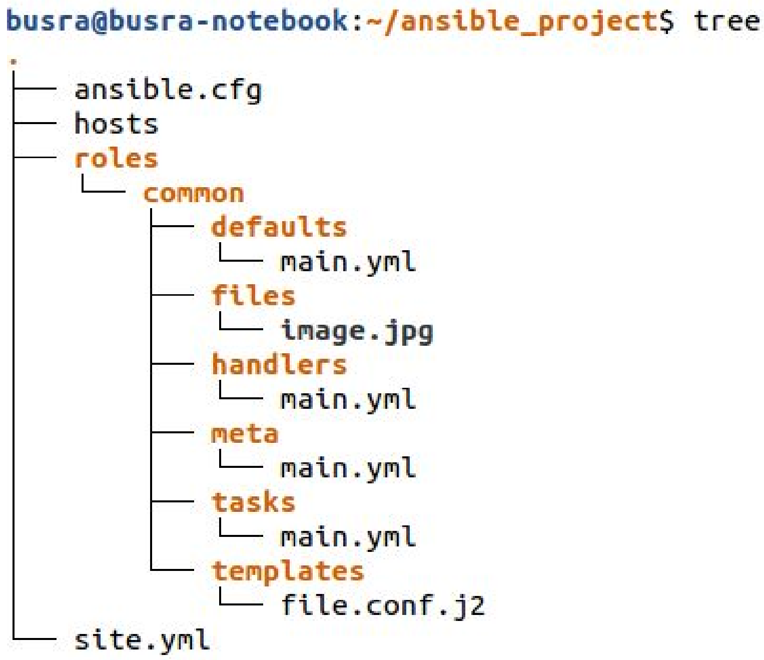
Big Community

How Ansible Works

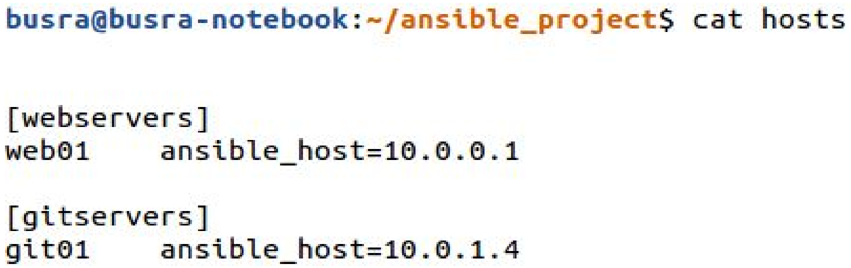


- Over ssh
- Inventory
- Playbooks
- Tasks
- Modules
- Directory structure

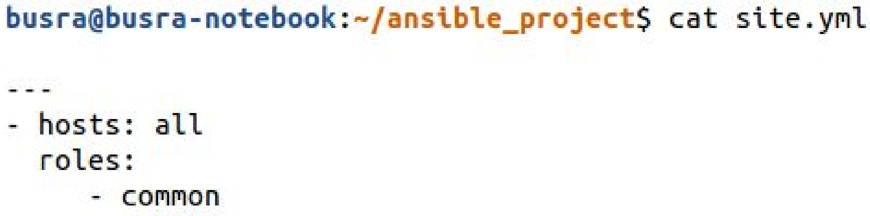
Directory Structure



```
busra@busra-notebook:~/ansible_project$ tree
.
├── ansible.cfg
├── hosts
├── roles
│   └── common
│       ├── defaults
│       │   └── main.yml
│       ├── files
│       │   └── image.jpg
│       ├── handlers
│       │   └── main.yml
│       ├── meta
│       │   └── main.yml
│       ├── tasks
│       │   └── main.yml
│       └── templates
│           └── file.conf.j2
└── site.yml
```

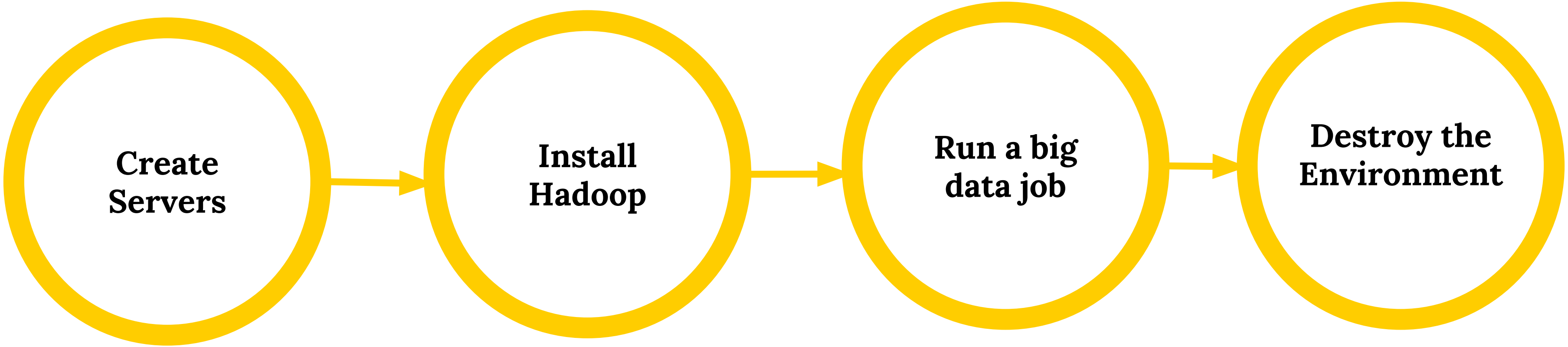


```
busra@busra-notebook:~/ansible_project$ cat hosts

[webservers]
web01     ansible_host=10.0.0.1

[gitservers]
git01     ansible_host=10.0.1.4
```



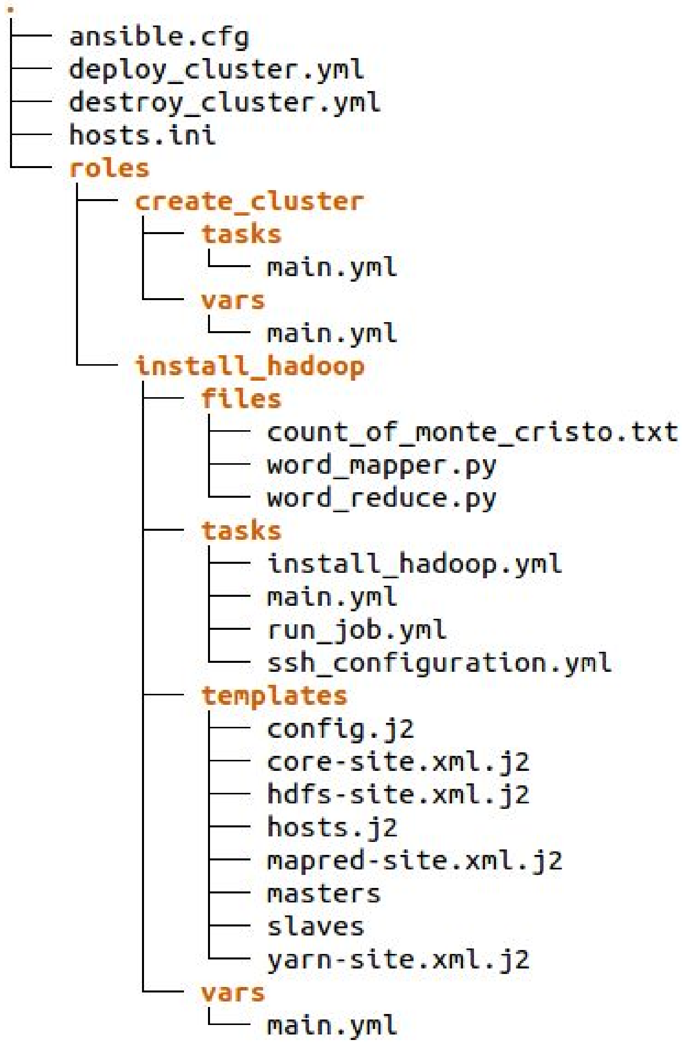
```
busra@busra-notebook:~/ansible_project$ cat site.yml

---
- hosts: all
  roles:
     - common
```

Case Study Explanation



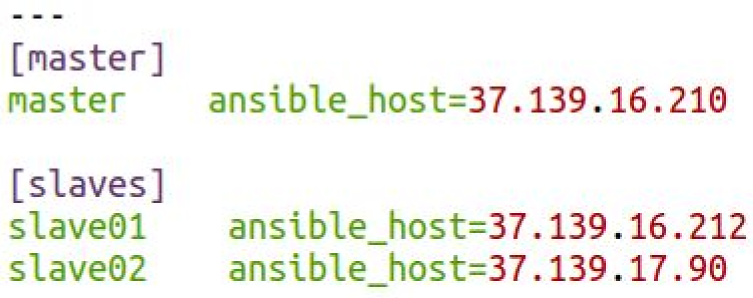
How my project looks

directory structure



```
.
├── ansible.cfg
├── deploy_cluster.yml
├── destroy_cluster.yml
├── hosts.ini
└── roles
    ├── create_cluster
    │   ├── tasks
    │   │   └── main.yml
    │   └── vars
    │       └── main.yml
    └── install_hadoop
        ├── files
        │   ├── count_of_monte_cristo.txt
        │   ├── word_mapper.py
        │   └── word_reduce.py
        ├── tasks
        │   ├── install_hadoop.yml
        │   ├── main.yml
        │   ├── run_job.yml
        │   └── ssh_configuration.yml
        ├── templates
        │   ├── config.j2
        │   ├── core-site.xml.j2
        │   ├── hdfs-site.xml.j2
        │   ├── hosts.j2
        │   ├── mapred-site.xml.j2
        │   ├── masters
        │   ├── slaves
        │   └── yarn-site.xml.j2
        └── vars
            └── main.yml
```

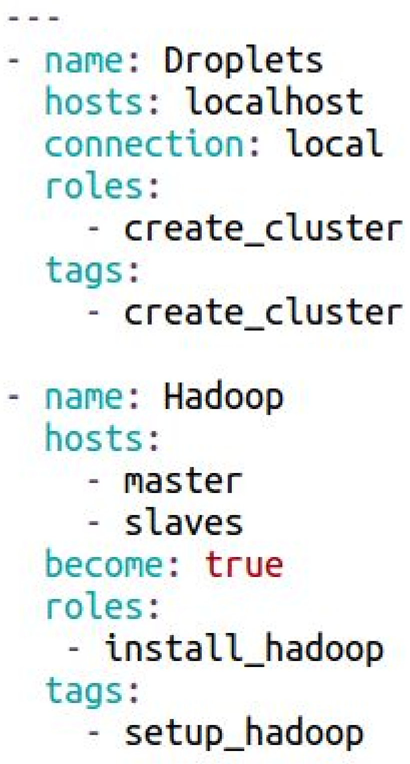
Host.ini file



```
---
[master]
master    ansible_host=37.139.16.210

[slaves]
slave01    ansible_host=37.139.16.212
slave02    ansible_host=37.139.17.90
```

deploy_cluster.yml

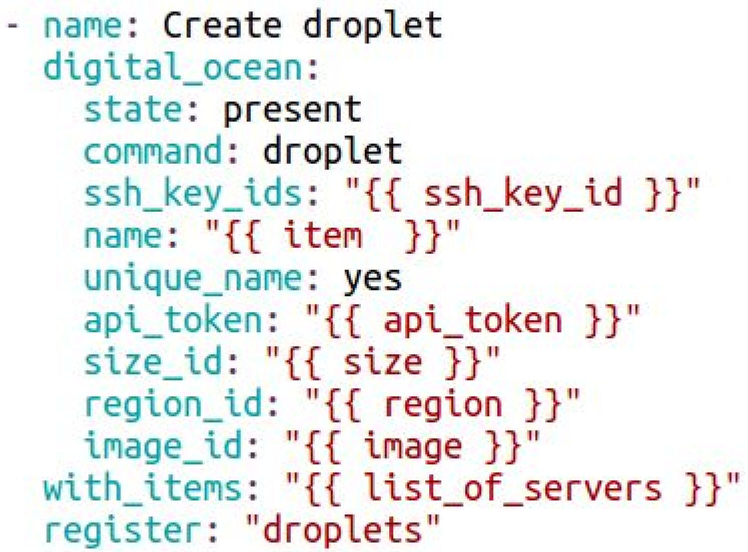
map hosts to roles



```yaml
---
- name: Droplets
  hosts: localhost
  connection: local
  roles:
    - create_cluster
  tags:
    - create_cluster

- name: Hadoop
  hosts:
    - master
    - slaves
  become: true
  roles:
   - install_hadoop
  tags:
    - setup_hadoop
```

Create Droplet

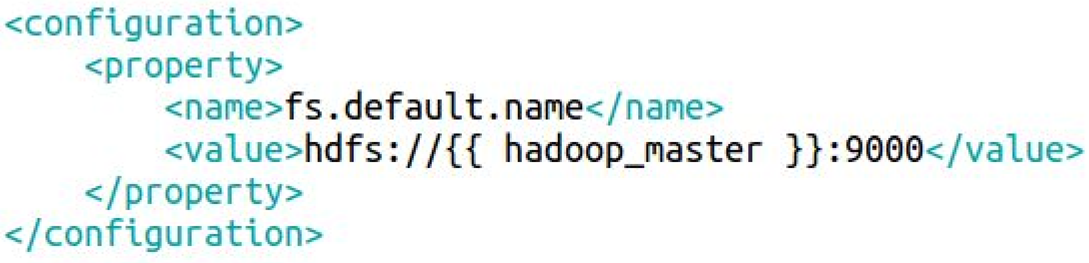
with digital_ocean module



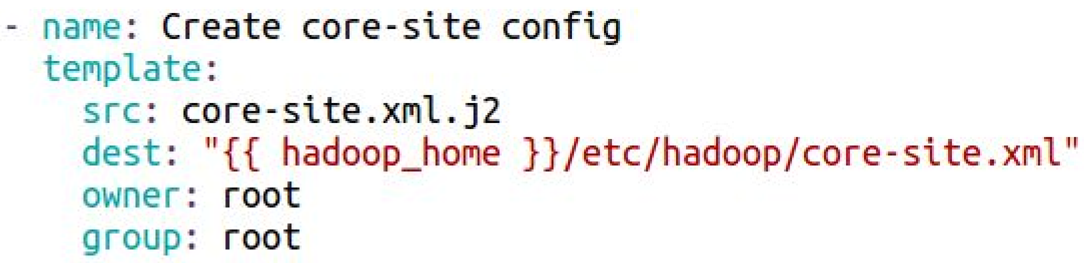
```
- name: Create droplet
  digital_ocean:
    state: present
    command: droplet
    ssh_key_ids: "{{ ssh_key_id }}"
    name: "{{ item  }}"
    unique_name: yes
    api_token: "{{ api_token }}"
    size_id: "{{ size }}"
    region_id: "{{ region }}"
    image_id: "{{ image }}"
  with_items: "{{ list_of_servers }}"
  register: "droplets"
```

Hadoop Configuration

with template module



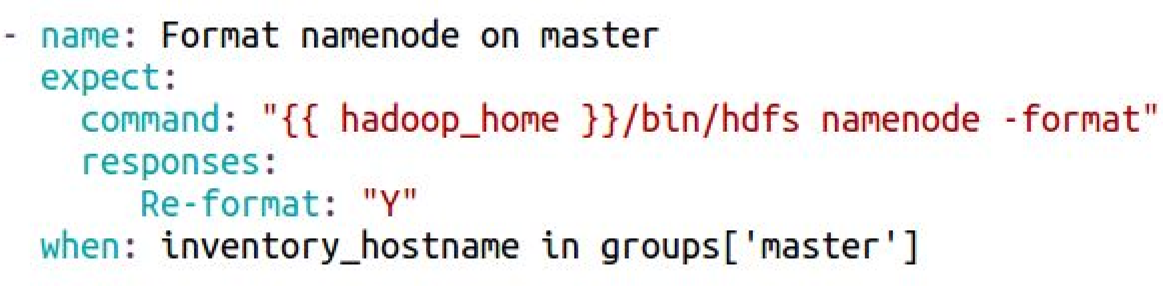
```
<configuration>
    <property>
        <name>fs.default.name</name>
        <value>hdfs://{{ hadoop_master }}:9000</value>
    </property>
</configuration>
```



```
- name: Create core-site config
  template:
    src: core-site.xml.j2
    dest: "{{ hadoop_home }}/etc/hadoop/core-site.xml"
    owner: root
    group: root
```

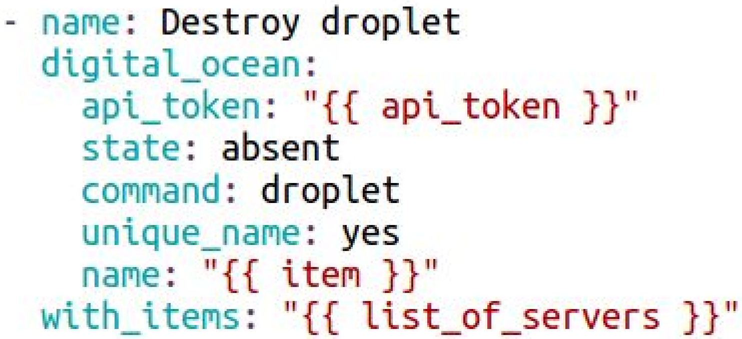
If you need to say “yes” at some point...

you can do it with expect* module!

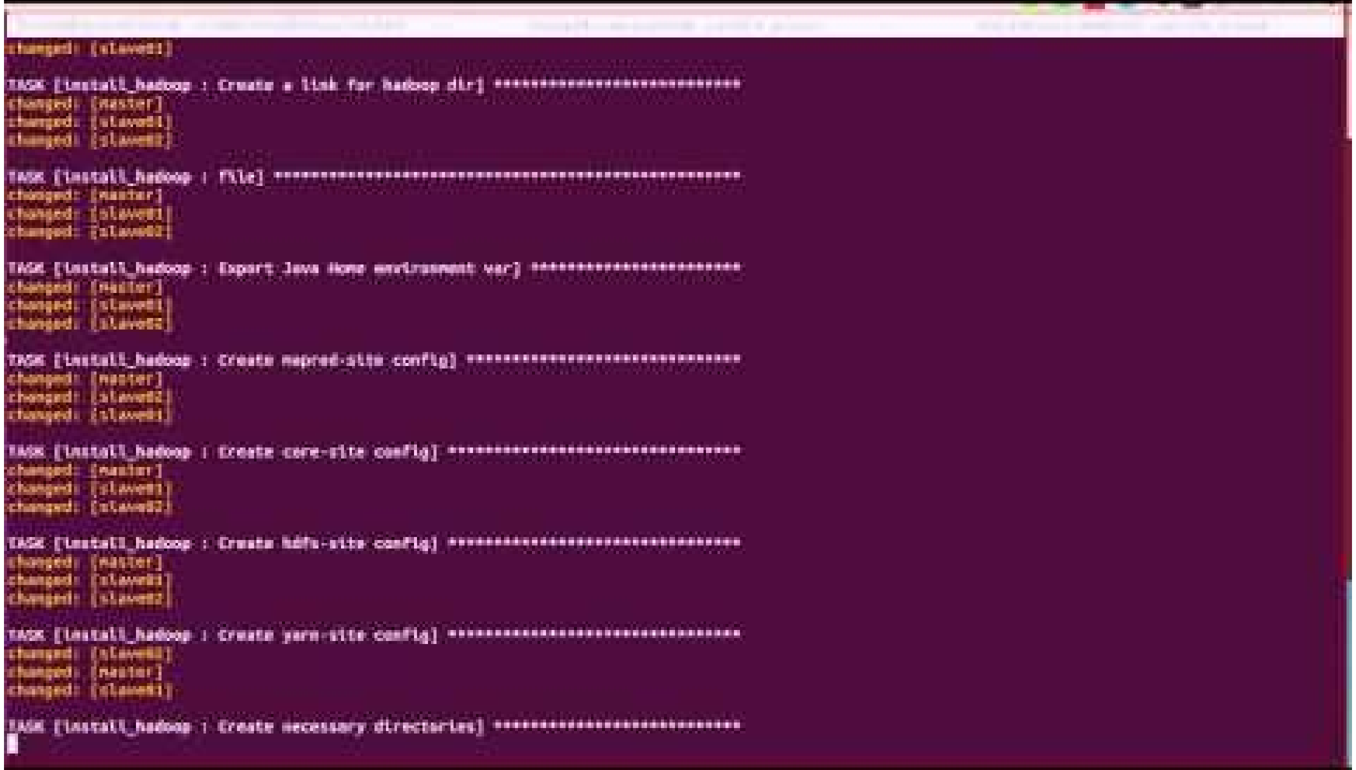


```
- name: Format namenode on master
  expect:
    command: "{{ hadoop_home }}/bin/hdfs namenode -format"
    responses:
       Re-format: "Y"
  when: inventory_hostname in groups['master']
```

Destroy created servers now!



```
- name: Destroy droplet
  digital_ocean:
    api_token: "{{ api_token }}"
    state: absent
    command: droplet
    unique_name: yes
    name: "{{ item }}"
  with_items: "{{ list_of_servers }}"
```



```
changed: [slave01]

TASK [install_hadoop : Create a link for hadoop dir] ***************************
changed: [master]
changed: [slave01]
changed: [slave02]

TASK [install_hadoop : file] ***************************************************
changed: [master]
changed: [slave01]
changed: [slave02]

TASK [install_hadoop : Export Java Home environment var] ***********************
changed: [master]
changed: [slave01]
changed: [slave02]

TASK [install_hadoop : Create mapred-site config] ******************************
changed: [master]
changed: [slave02]
changed: [slave01]

TASK [install_hadoop : Create core-site config] ********************************
changed: [master]
changed: [slave01]
changed: [slave02]

TASK [install_hadoop : Create hdfs-site config] ********************************
changed: [master]
changed: [slave01]
changed: [slave02]

TASK [install_hadoop : Create yarn-site config] ********************************
changed: [slave02]
changed: [master]
changed: [slave01]

TASK [install_hadoop : Create necessary directories] ***************************
```

Let’s Review What We Have Seen

Easy to get started just like Python

Clean architecture

Straightforward configuration

Extendable module structure

Makes complex tasks easy

Any questions ?

You can contact me via;

- linkedin.com/in/busrakoken
- busra.koken@ericsson.com

- Project Github link: **github/busrakoken/cloudy-hadoop**