29.11.2023

ESD WORKSHOP

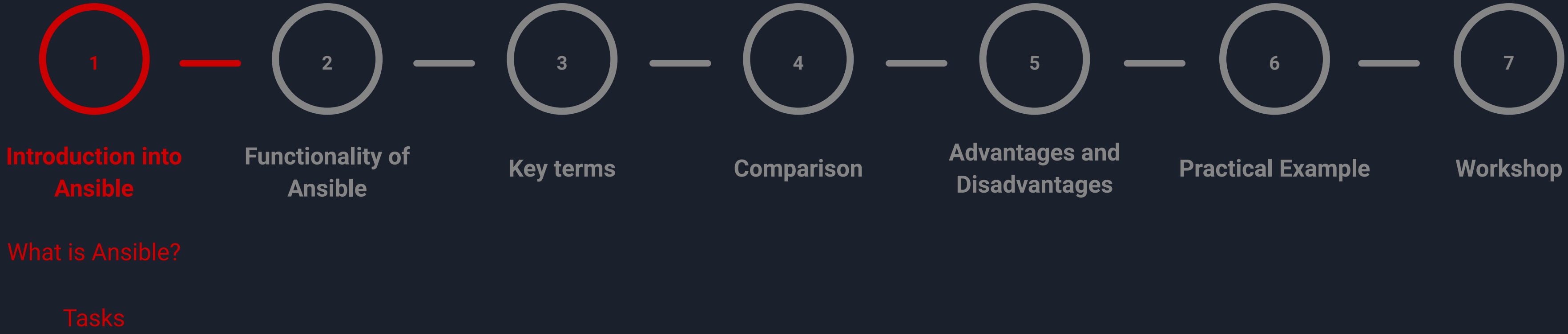
ANSIBLE

Jonas Klotzbach, Jan Polfers and Jannik Schwerdtner

GITHUB REPOSITORY

https://github.com/sebivenlo/ESD-2023-Ansible

CONTENT



1 Introduction into Ansible

What is Ansible?

Tasks

2 Functionality of Ansible

3 Key terms

4 Comparison

5 Advantages and Disadvantages

6 Practical Example

7 Workshop

WHAT IS ANSIBLE?

- Open-source automation
- YAML-based syntax
- Utilizes Playbooks to define tasks and configurations
- Playbooks configure systems and deploy applications

TASKS

- Configuration management
- Cloud provisioning
- Creating and managing DBs
- Application orchestration
- OS level updates
- Application deployment

CONTENT

1 Introduction into Ansible

2 Functionality of Ansible

Ansible Architecture

3 Key terms

4 Comparison

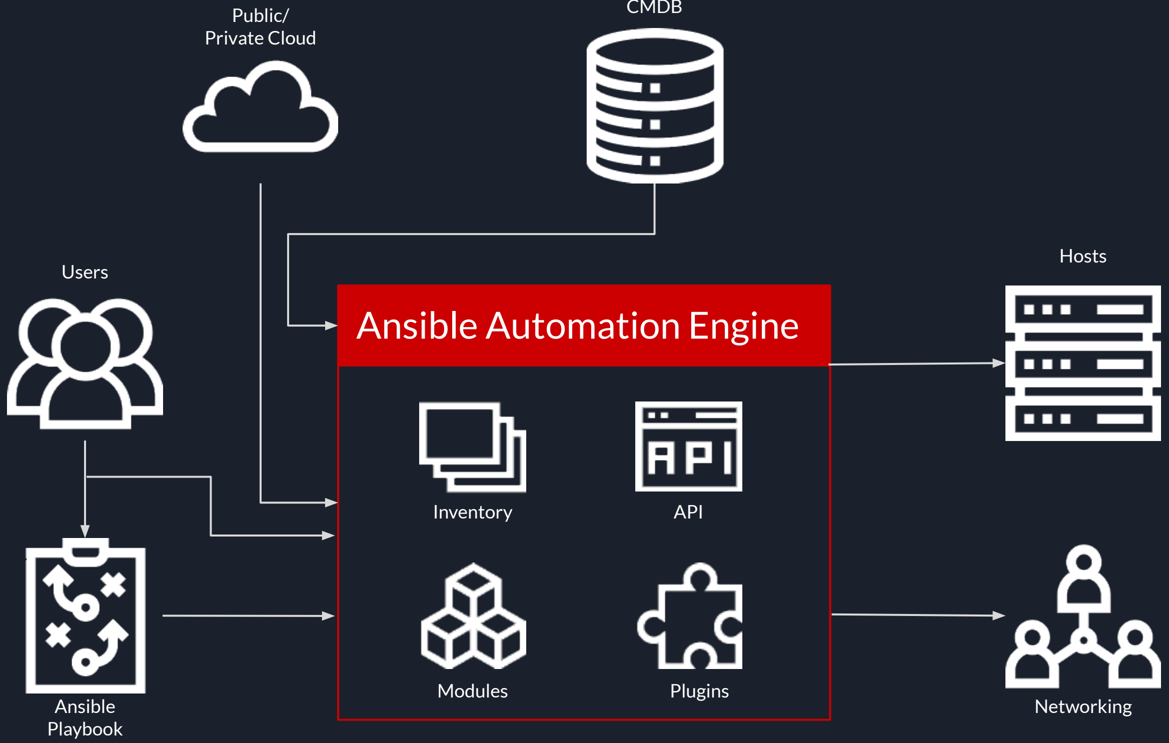
5 Advantages and Disadvantages

6 Practical Example

7 Workshop

ANSIBLE ARCHITECTURE

- Playbooks define tasks and configurations
- Tasks are individual actions within Playbooks
- Inventory lists all managed systems
- Modules are reusable automation actions
- Communication via SSH or WinRM



CONTENT

1 Introduction into Ansible

2 Functionality of Ansible

3 **Key terms**

- Terms
- Exercise 1
- Variables
- Loops & Conditionals

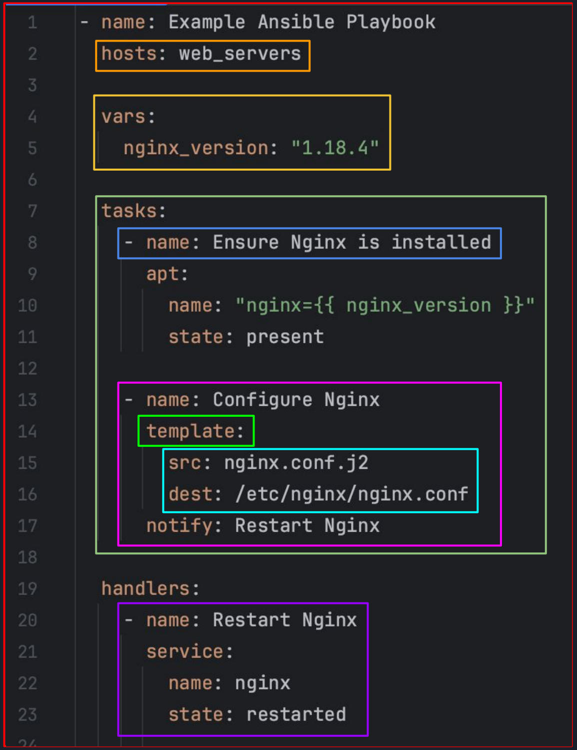
4 Comparison

5 Advantages and Disadvantages

6 Practical Example

7 Workshop

TERMS



```yaml
- name: Example Ansible Playbook
  hosts: web_servers

  vars:
    nginx_version: "1.18.4"

  tasks:
    - name: Ensure Nginx is installed
      apt:
        name: "nginx={{ nginx_version }}"
        state: present

    - name: Configure Nginx
      template:
        src: nginx.conf.j2
        dest: /etc/nginx/nginx.conf
      notify: Restart Nginx

  handlers:
    - name: Restart Nginx
      service:
        name: nginx
        state: restarted
```

Playbook

Host definition

Variable definition

Play

Task name

Task

Module name

Arguments

Handler

1. Exercise

https://github.com/sebivenlo/ESD-2023-Ansible

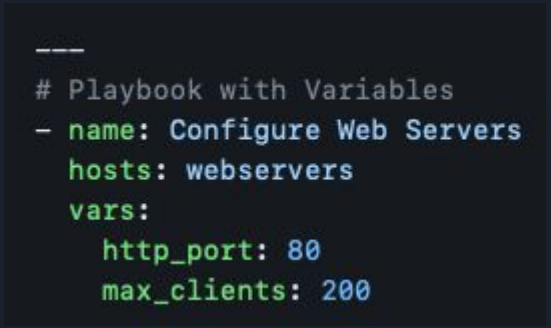
VARIABLES

Types of Ansible Variables:

1. **Global Variables:**
   - Defined in the group_vars or host_vars directory.
   - Apply to all hosts or specific hosts respectively.
2. **Playbook Variables:**
   - Defined within a playbook using the vars keyword.
   - Scoped to a specific playbook.
3. **Facts:**
   - Automatically collected by Ansible about remote systems.
   - Accessed using the ansible_facts variable.
4. Environment Variables:
   - Set externally and accessed within Ansible playbooks.

Variable Syntax:

- Ansible variables are enclosed in double curly braces, like {{ variable_name }}
- You can also use the Jinja2 templating language for more advanced variable usage



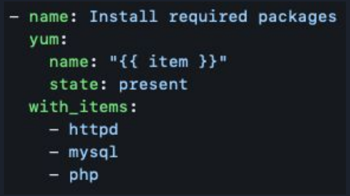
```
---
# Playbook with Variables
- name: Configure Web Servers
  hosts: webservers
  vars:
    http_port: 80
    max_clients: 200
```

LOOPS & CONDITIONALS

LOOPS

1. With_Items Loop:

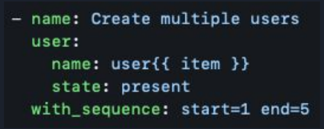
- Iterates over a list of items.
- Executes tasks for each item.



```
- name: Install required packages
  yum:
    name: "{{ item }}"
    state: present
  with_items:
    - httpd
    - mysql
    - php
```

2. Looping over a Range:

- Iterates over a range of numbers.

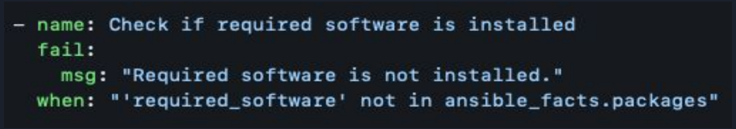


```
- name: Create multiple users
  user:
    name: user{{ item }}
    state: present
  with_sequence: start=1 end=5
```

CONDITIONALS

1. When Statement:

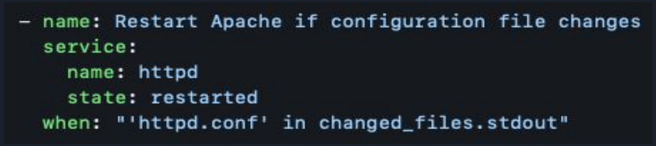
- Executes a task based on a specified condition.



```
- name: Check if required software is installed
  fail:
    msg: "Required software is not installed."
  when: "'required_software' not in ansible_facts.packages"
```

2. Fail Module:

- Aborts the playbook if a condition is not met.



```
- name: Restart Apache if configuration file changes
  service:
    name: httpd
    state: restarted
  when: "'httpd.conf' in changed_files.stdout"
```

CONTENT

1 Introduction into Ansible

2 Functionality of Ansible

3 Key terms

4 **Comparison**

5 Advantages and Disadvantages

6 Practical Example

7 Workshop

Puppet

Chef

Ansible

COMPARISON

	Puppet *(2005)*	**Chef** *(2009)*	**Ansible** *(2012)*
Configuration	Puppet DSL	Ruby-based Recipes	YAML-based Playbooks
Scalability	High	Scales well with larger systems	High
Availability	Puppet Agent	Chef Client	Agentless
Programming Language	Ruby	Ruby/ Erlang	Python
Implementation	Puppet Agent on managed nodes	Chef Client on managed nodes	SSH and Python on managed nodes
Ease of use	Moderate	Requires more initial setup	Easy
Orchestration	Limited	Limited	Strong support

CONTENT

1 Introduction into Ansible

2 Functionality of Ansible

3 Key terms

4 Comparison

5 **Advantages and Disadvantages**

Why Ansible?

6 Practical Example

7 Workshop

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Simplicity
- Scalability
- Extensibility
- Idempotency
- Agentlessness
- Large and Active Community
- Wide Platform Support
- Orchestration Capabilities
- Integration with Cloud Services

- Learning curve
- SSH-based communication
- Limited Built-in Error Handling
- Lack of Formal Windows Support for Controller
- Performance for Large Scale Deployments
- Limited Graphical Interface
- Security Concerns

DISADVANTAGES

CONTENT

1 Introduction into Ansible

2 Functionality of Ansible

3 Key terms

4 Comparison

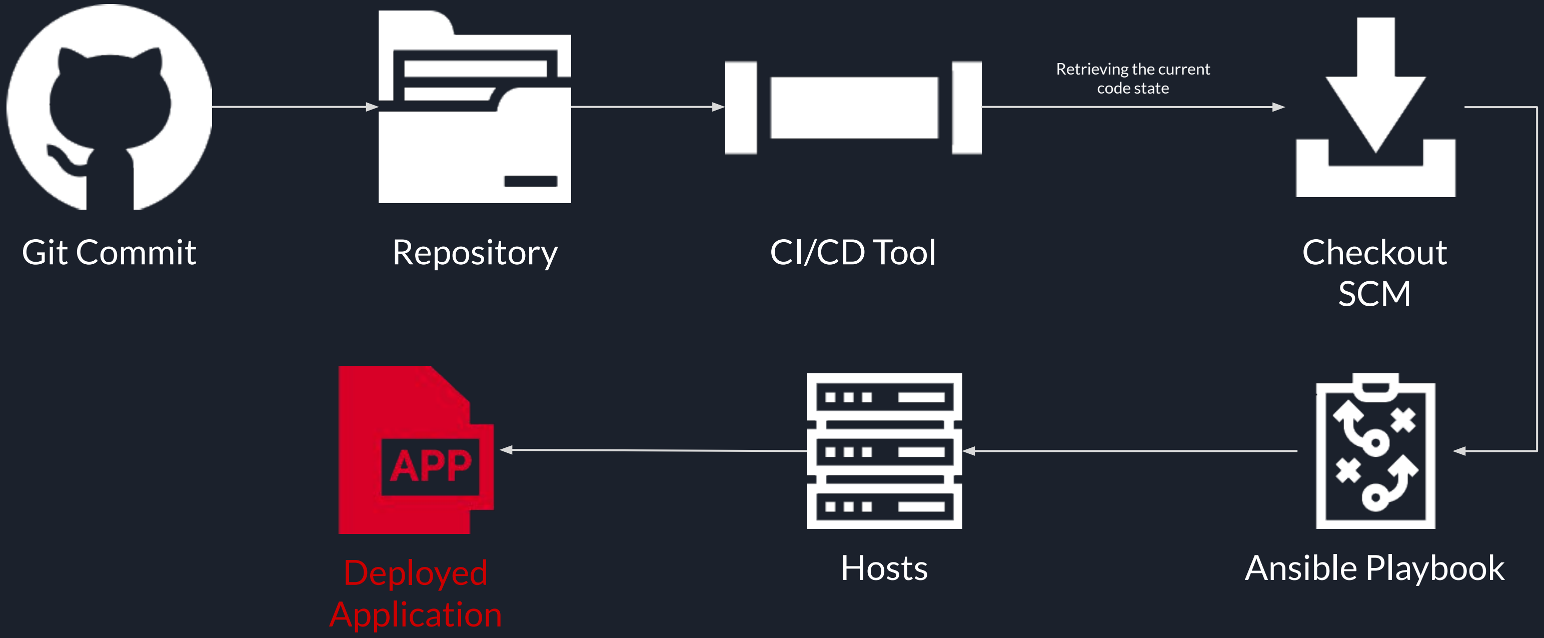
5 Advantages and Disadvantages

6 Practical Example

Deployment

7 Workshop

DEPLOYMENT


Git Commit
Repository
CI/CD Tool
Retrieving the current code state
Checkout SCM
Ansible Playbook
Hosts
APP
Deployed Application

CONTENT

1. Introduction into Ansible
2. Functionality of Ansible
3. Key terms
4. Comparison
5. Advantages and Disadvantages
6. Practical Example
7. **Workshop**

Exercise 2

Exercise 3

Kafka Example

2./3. Exercise

https://github.com/sebivenlo/ESD-2023-Ansible

Kafka Update and Deployment Management

A real-life example