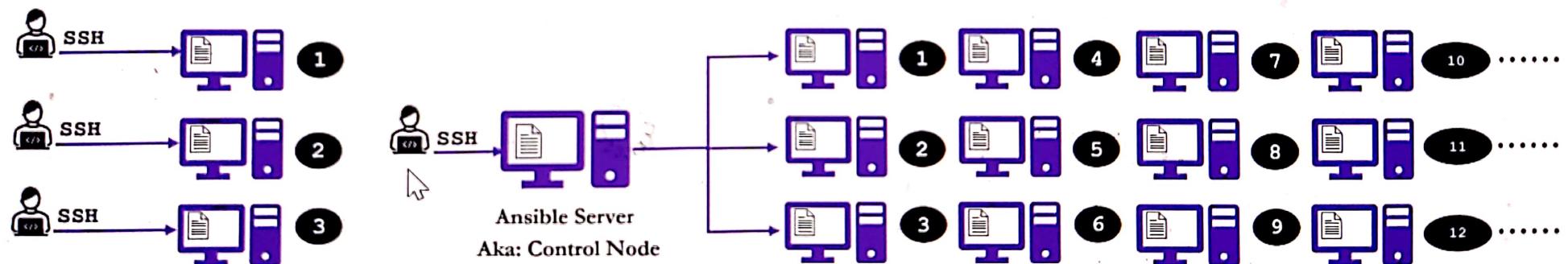


What is Ansible?

- Ansible is an open-source software provisioning, configuration management, and application-deployment tool enabling infrastructure as code. It runs on many Unix-like systems, and can configure both Unix-like systems as well as Microsoft Windows (*Wikipedia*)
- In simple words: Ansible is a free automation tool that can automate IT tasks on local machine where it is running and on remote machines



- Please note:** Ansible is written in python language, but it does not mean you need python knowledge to use Ansible

By: Imran Afza
www.utclisolutions.com

What is Ansible?

- Ansible can be used to:



Provision system

Configure system

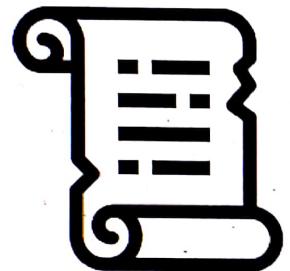
Deploy Apps

Manage system and Apps

By: Imran Afz
tclisolutions.co

Brief History of Ansible

- The Ansible project was started in 2012 by Michael DeHaan
 - It is open source and community driven
 - Ansible Inc was purchased by Red Hat in 2015
 - It is available for most of the Linux distributions such as, Red Hat, CentOS, Fedora, Ubuntu, Debian and SUSE
-
- Need?
 - Infrastructure management (specially virtualization)
 - Configuration management (System or Application configuration)
 - Multi-tier application automation (e.g. app, web and db servers)
 - Single point of automation (having scripts on each system vs. one automation management platform).



By: Imran Afzal

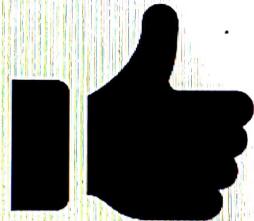
www.utclisolutions.com



Scanned with OKEN Scanner

Benefits of Ansible

- Agentless!!!
- Open-source
- Avoid human errors
- Saves time by automating repetitive or tedious tasks
- Increase productivity
- Easy to use
- Simple (human readable text files)
- Flexible
- Secure (over SSH).



By: Imran Afzal

www.utclisolutions.com

Terminologies in Ansible

- **Control node or Ansible Server**
 - Server which runs Ansible application
- **Modules**
 - Module is a command meant to be executed on the client-side
 - Most of the IT tasks modules are already created and can be found on Ansible website
 - www.docs.ansible.com → search for module index
 - www.galaxy.ansible.com
 - Example of modules:
 1. Install http
 2. Enable http service
 3. Start http service

```
graph LR; A[1. Install http] --- B[2. Enable http service]; B --- C[3. Start http service]; B --- D[Task];
```
- **Task**
 - A task is a section that consists of a single procedure to be completed. A task can have multiple modules



By: Imran

www.utclisolutions.com

Terminologies in Ansible

- **Playbook**
 - Automation file with step-by-step execution of multiple tasks
 - **YAML**
 - A Playbook written in YAML language (Yet another markup language)
 - **Inventory**
 - File that has information about remote clients where tasks are executed
 - **Tag**
 - A reference or alias to a specific task
 - **Variable**
 - Variables are like containers that holds the defined value which can be used repetitively
 - **Role**
 - Splitting of Playbook into smaller groups. Roles let you automatically load related vars, files, tasks, handlers, and other Ansible artifacts based on a known file structure. After you group your content in roles, you can easily reuse them and share them with other users



By: Imran Afzal

www.utclisolutions.com

How Ansible Works?

Commands examples:

To run modules through yaml file:

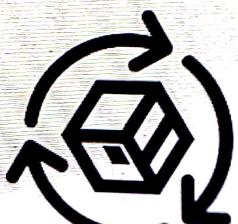
```
# ansible-playbook example.yml
```

To run module independently

```
# ansible myservers -m ping
```

Ansible configuration files:

- /etc/ansible/ansible.cfg
- /etc/ansible/hosts
- /etc/ansible/roles



YAML File Syntax Example

```
---
```

```
- name: sampleplaybook
hosts: all or localhost
become: yes
become_user: root
```

```
tasks:
- name: Install Apache httpd
  yum:
    name: httpd
    state: present
```

```
- name: 2nd task
  service:
    name: httpd
    state: started
```

Declare task

Name of the playbook

Where to run?

Run as a different user?

Define the name of the task

Run task module yum

Name of the package

What to do? -- Install

- Ansible modules and options
<https://docs.ansible.com/ansible/2.5/modules/>

By: Imran Afzal
www.utclisolutions.com



Creating First Playbook

- There are online yaml tools you can use to create Playbooks
 - <https://onlineyamltools.com/edit-yaml>
 - <https://codebeautify.org/yaml-editor-online>
 - Tools to download
 - Notepad++ Windows
 - Take a snapshot after installing Ansible software



```
# su - root
# mkdir /etc/ansible/playbook
# cd /etc/ansible/playbooks
# vim first.yml

---
- name: "My first playbook"
  hosts: localhost

  tasks:
    - name: "test connectivity"
      ping:
```

- Check syntax of playbook
`# ansible-playbook --syntax-check first.yml`
 - Or to do a dry run
`# ansible-playbook --check first.yml`
 - Run the playbook
`# ansible-playbook /root/ansible/first.yml`

Ultimatix - Digitally Connected! x | Ginger Software | English Gramm x | PDF 1-Introduction.pdf x | PDF 3-Ansible+Automation+with+Simple+Playbooks.pdf x | +

C H File | C:/Users/vxr214/Downloads/3-Ansible+Automation+with+Simple+Playbooks.pdf

- + + 7 of 9

Output Playbook

This playbook will print “Hello World” on localhost



```
# cd /etc/ansible/playbook
# vim helloworld.yml

---
- name: My Second playbook
  hosts: localhost
  tasks:
    - name: Print Hello World
      debug: msg="Hello World"
```

→ Name of the play or playbook
→ Run on localhost
→ Run the following task
→ Name of the task
→ Run debug module which prints statements during execution

Run the playbook
ansible-playbook helloworld.yml

By: Imran Afzal
www.utclisolutions.com



Scanned with OKEN Scanner

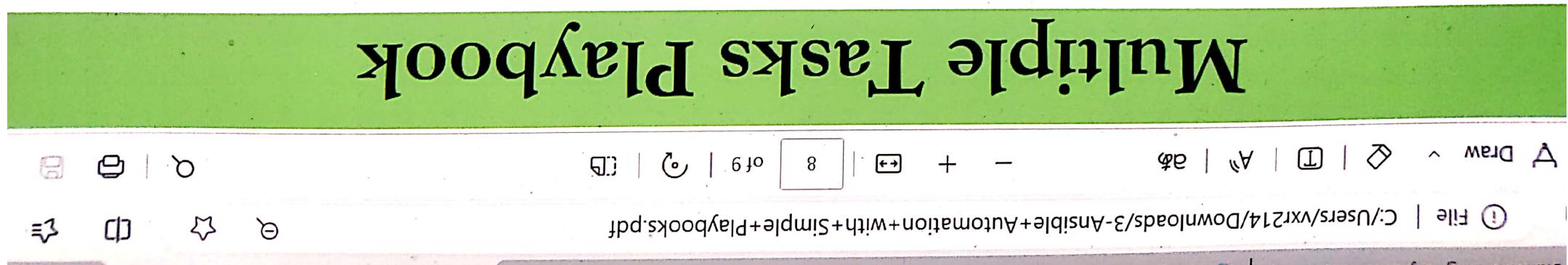


Multiple Tasks Playbook

```
# vim task.yaml
# ansible-playbook mtask.yaml

tasks:
  - name: Running 2 tasks
    hosts: localhost
    Name of the play
    Run it on local host
  - name: Test connectivity
    ping:
      Name of the task
    Run the following task
  - name: Test connectivity
    ping:
      Name of the task
    Run the ping module
  - name: Print Hello World
    debug:
      msg="Hello World"
      Name of the 2nd task
    Run the debug module
```

The playbook will ping localhost and print "Hello World".



Installing and Starting a Package

```
# vim packinstall.yml

---
- name: Installing and Running apache
  hosts: localhost

  tasks:
    - name: Install apache
      yum:
        name: httpd
        state: present

    - name: start httpd
      service:
        name: httpd
        state: started
```



Run the playbook

```
# ansible-playbook packinstall.yml
```

By: Imran Afzal
www.utclisolutions.com

Running Ansible Ad-Hoc Commands

Ad-hoc commands part 1

- Ad-hoc commands run on as needed basis and usually for those tasks that do not repeat

- Syntax for Ad-Hoc ansible command:

- **ansible [target] -m [module] -a "[module options]"**

- Example ansible ad-hoc commands:

- Ping localhost

- ```
ansible localhost -m ping
```

- Creating a file on all remote clients

- ```
# ansible all -m file -a "path=/home/iafzal/adhoc1 state=touch mode=700"
```

- Deleting a file on all remote clients

- ```
ansible all -m file -a "path=/home/iafzal/adhoc1 state=absent"
```

- Copying a file to remote clients

- ```
# ansible all -m copy -a "src=/tmp/adhoc2 dest=/home/iafzal/adhoc2"
```

By:

www.utcliso

Running Ansible Ad-Hoc Commands

Ad-hoc commands part 2

- Installing package (telnet and httpd-manual)

```
# ansible all -m yum -a "name=telnet state=present"  
# ansible all -m yum -a "name=httpd-manual state=present".
```

- Starting httpd package service

```
# ansible all -m service -a "name=httpd state=started"
```

- Start httpd and enable at boot time

```
# ansible all -m service -a "name=httpd state=started enabled=yes"
```

- Checking httpd service status on remote client

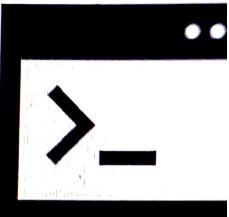
```
# ansible all -m shell -a "systemctl status httpd"
```

- Remove httpd package

```
# ansible all -m yum -a "name=httpd state=absent"
```

OR

```
# ansible all -m shell -a "yum remove httpd".
```



Running Ansible Ad-Hoc Commands

Ad-hoc commands part 3

- Creating a user on remote clients

```
# ansible all -m user -a "name=jsmith home=/home/jsmith shell=/bin/bash state=present"
```

- To add a user to a different group

```
# ansible all -m user -a "name=jsmith group=iafzal"
```

- Deleting a user on remote clients

```
# ansible all -m user -a "name=jsmith home=/home/jsmith shell=/bin/bash state=absent"  
OR
```

```
# ansible all -m shell -a "userdel jsmith"
```

- Getting system information from remote clients

```
# ansible all -m setup
```

- You can run commands on the remote host without a shell module e.g. reboot client1

```
# ansible client1 -a "/sbin/reboot"
```



By: Imran Afza
www.utclisolutions.com

to search



Scanned with OKEN Scanner