Complete Ansible Automation Training

Ansible for Remote Clients Management

Remote Clients hosts File Syntax

/etc/ansible/hosts

- All remote clients are considered inventory in Ansible
- Ansible keeps its inventory information in host file located: **/etc/ansible/hosts**
- The **hosts** file is created during Ansible installation

```
app1.example.com
app2.example.com
web1.example.com
206.198.210.35
205.168.20.13
```

One client listed in each line

```
[appservers]
app1.example.com
app2.example.com

[webservers]
web1.example.com
web2.example.com
206.198.210.35

[dbservers]
205.168.20.11
205.168.20.12
```

Headers to group clients

```
[allservers]
app1.example.com
app2.example.com
web1.example.com
web2.example.com
10.91.50.111
10.91.50.112
10.91.50.113
10.91.50.114
```

No need to define allservers because Ansible has a default for all

IP Range

```
205.168.20.[11:14]
```

- You can specify different location of the file

```
# ansible-playbook -i /home/iafzal/ansible/hosts
```

By: Imran Afzal
www.utclisolutions.com

Remote Clients hosts File Syntax

/etc/ansible/hosts

```
[servers]
server1 ansible_ssh_host=10.91.50.110
server2 ansible_ssh_host=10.91.50.111
server3 ansible_ssh_host=10.91.50.112
server4 ansible_ssh_host=10.91.50.113
server5 ansible_ssh_host=10.91.50.114
server6 ansible_ssh_host=10.91.50.115

[appserver]
server1
server2

[webserver]
server3
server4

[dbservers]
server5
server6
```

Aliases

By: Imran Afzal
www.utclisolutions.com

Remote Clients hosts File Syntax

/etc/ansible/hosts

- Inventory host file can either be static or dynamic *(using additional plug-ins)*

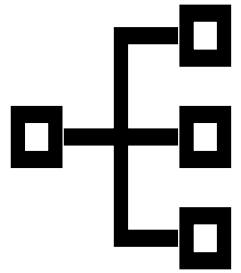
- Listing host file

```
# ansible-inventory --list
OR
# ansible all --list-hosts
```

By: Imran Afzal
www.utclisolutions.com

Establish Connection to Remote Clients

- Take a snapshot of our Linux client1 and then power it up
- Note down its IP address



- Populate the hosts file with IP or FQDN for our clients:

```
[labclients] = For grouping
10.253.1.18
10.253.1.20
```

- Generate SSH Keys on the control node and copy over to clients for password less SSH connections

```
# ssh-keygen
# Leave everything default and enter
# ssh-copy-id 10.253.1.18
# ssh-copy-id 10.253.1.20
```

- Now SSH into the clients to test

```
# ssh 10.253.1.18
```

- Run **Ansible** add-hoc to ping remote nodes *(make sure hosts file has remote clients IPs)*

```
# ansible all -m ping
# ansible -a "uptime" all
```
(To run a command on the remote clients)

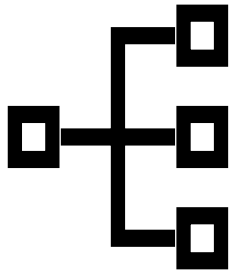
By: Imran Afzal
www.utclisolutions.com

Check Remote Clients Connectivity

```
# su - root
# cd /etc/ansible/playbooks
# vim clientstatus.yml

---
- name: "Check remote clients connectivity status"
  hosts: all

  tasks:
  - name: Test connectivity
    ping:
```

Run the playbook

```
# anisble-playbook clientstatus.yml
```



By: Imran Afzal
www.utclisolutions.com

Copy Files to Remote Clients

```
# echo somestuff > /home/iafzal/some.cfg
# vim copy.yml
```

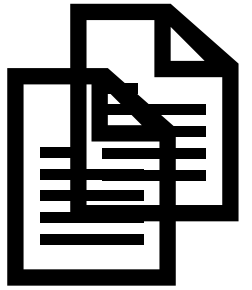
```
---
- name: Copy file from local to remote clients
  hosts: all

  tasks:
  - name: Copying file
    become: true
    copy:
     src: /home/iafzal/some.cfg
     dest: /tmp
     owner: iafzal
     group: iafzal
     mode: 0644
```

Description of the playbook
Run it on all hosts

Run the following task(s)
Description of the task
Transfer as a current user
Run **copy** module

Source of the file
Destination of the file

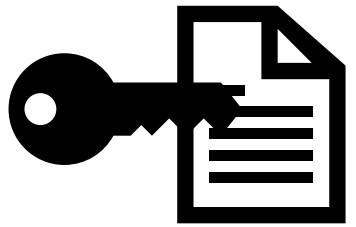
Change ownership and file permissions



By: Imran Afzal
www.utclisolutions.com

Change File Permissions

```
# Login to LinuxClient1
# touch /home/iafzal/linux2

# Login to ControlNode
# vim filepermission.yml
```



```
---
- name: Change file permissions
  hosts: all

  tasks:
  - name: Files Permissions
    file:
     path: /home/iafzal/linux2
     mode: a+w
```

File location

Permissions

Run the playbook

```
# anisble-playbook filepermission.yml
```

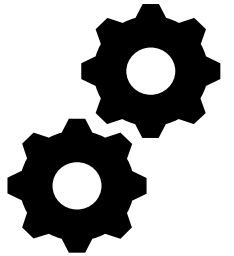
- Ansible modules and options

https://docs.ansible.com/ansible/2.5/modules/

By: Imran Afzal
www.utclisolutions.com

Setup Apache and Open Firewall Port

- The playbook will
  1. Install httpd package
  2. Start httpd service
  3. Open http service port in firewall
  4. Restart firewalld service



```
# Login to LinuxCleint1
# rpm -qa | grep http
# systemctl status firewalld

# Login to ControlNode
```

- Install additional Ansible collection for firewalld

```
# ansible-galaxy collection install ansible.posix

# cd /home/ansible/playbooks
# vim httpsetup.yml
```

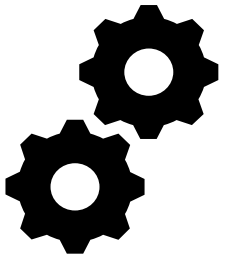
- Ansible modules and options

https://docs.ansible.com/ansible/2.5/modules/

By: Imran Afzal
www.utclisolutions.com

Setup Apache and Open Firewall Port

```
---
- name: Setup httpd and open firewall port
  hosts: all
  tasks:
  - name: Install apache packages
    yum:
     name: httpd
     state: present

  - name: Start httpd
    service:
     name: httpd
     state: started

  - name: Open port 80 for http access
    firewalld:
     service: http
     permanent: true
     state: enabled

  - name: Restart firewalld service to load firewall changes
    service:
     name: firewalld
     state: reloaded
```



State = What to do with the package?

- present or installed = Install
- absent or removed = Un-install
- latest = Upgrade

State = What to do with the service?

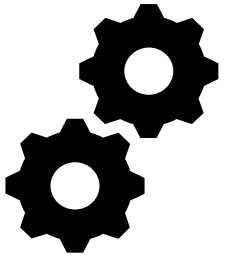
- started | stopped | reloaded | restarted

save httpsetup.yml

By: Imran Afzal
www.utclisolutions.com

Setup Apache and Open Firewall Port

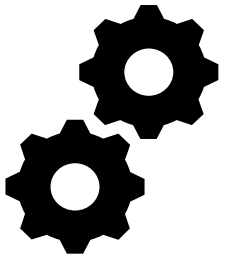
- Login back to **LinuxCleint1**
- Check httpd package status
  ```
  # rpm -qa | grep http
  ```
- Check httpd package service status
  ```
  # systemctl status httpd
  ```
- Check firewalld service status
  ```
  # systemctl status firewalld
  ```
- Check if http service is enabled in firewalld
  ```
  # firewall-cmd --list-all
  ```
- Open FireFox and go to 10.253.1.115



By: Imran Afzal
www.utclisolutions.com

Run Shell Scripts on Remote Clients

- The playbook will run shell script on the remote client (LinuxClient1)
- Create **/home/iafzal/cfile.sh** script on **LinuxClient1**
- The **cfile.sh** script should create a new file **example1**



```
# vim shellscript.yml

---
- name: Playbook for shell script
  hosts: all or 10.253.1.115

  tasks:
  - name: Run shell script
    shell: "/home/iafzal/cfile.sh"
```

Description of the playbook
Run on client1

Run the following task
Name/description of the task
Run **shell** module which will execute shell script on LinuxClient1

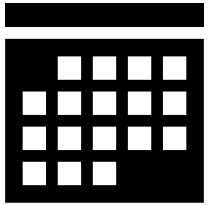
Run the playbook

```
# anisble-playbook shellscript.yml
```

By: Imran Afzal
www.utclisolutions.com

Schedule a job (crontab)

- The playbook **cronjob.yml** will
  - Schedule a job as a **root**
  - Every thursday at 10am
  - Define job **(/home/iafzal/cfile.sh)** to be executed by **root**



```
# vim cronjob.yml

---
- name: Create a cron job
  hosts: all

  tasks:
    - name: Schedule cron:
      cron:
        name: This job is scheduled by Ansible
        minute: "0"
        hour: "10"
        day: "*"
        month: "*"
        weekday: "4"
        user: root
        job: "/home/iafzal/cfile.sh"
```

Day of the month

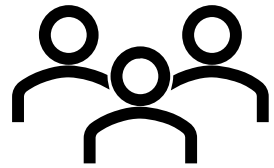
Day of the week

Run the playbook

```
# anisble-playbook cronjob.yml
```

By: Imran Afzal
www.utclisolutions.com

User Account Management



- The playbook will
  - Create a user **`george`** on remote clients
  - The user **`george`** will have a home directory as **`/home/George`**
  - The shell environment for user **`george`** will be **`/bin/bash`**

```
# vim adduser.yml

---
- name: Playbook for creating users
  hosts: all

  tasks:
  - name: Create users
    user:
     name: george
     home: /home/george
     shell: /bin/bash
```

Description of the playbook
Run on all clients
Run the following task
Name of the task
Run **user** module for account management
Username
User home directory
User shell

Run the playbook

```
# anisble-playbook adduser.yml
```

By: Imran Afzal
www.utclisolutions.com

Add or Update User Password

- The playbook will
  - Add/update a password for user **george**

Please note: Ansible does not allow us to pass a cleartext password through the user **module**

```
# vim changepass.yml
---
- name: Add or update user password
  hosts: all

  tasks:
  - name: Change "george" password
    user:
      name: george
      update_password: always
      password: "{{ newpassword|password_hash('sha512') }}"
```

- To run this playbook, run the command as below. This will input the **newpassword** variable that will be used by our playbook

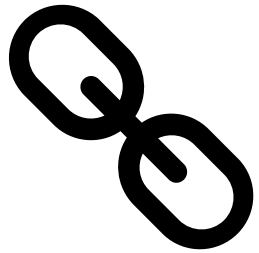
Run the playbook

```
# ansible-playbook changepass.yml --extra-vars newpassword=abc123
```

By: Imran Afzal
www.utclisolutions.com

Download Package from a URL

- The playbook tomcat.yml will
  - Create a directory for tomcat with required permissions
  - Download tomcat from a url and place it in that directory with modified permissions



```
vim tomcat.yml
---
- name: Download Tomcat from tomcat.apache.org
  hosts: localhost
  tasks:
   - name: Create a Directory /opt/tomcat
     file:
       path: /opt/tomcat
       state: directory
       mode: 0755
       owner: root
       group: root
   - name: Download Tomcat using get_url
     get_url:
       url: https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.78/bin/apache-tomcat-8.5.78.tar.gz
       dest: /opt/tomcat
       mode: 0755
       group: iafzal
       owner: iafzal
```

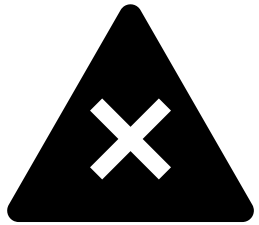
Run the playbook

```
# anisble-playbook tomcat.yml
```

By: Imran Afzal
www.utclisolutions.com

Kill a Running Process

- The playbook **`killprocess.yml`** will
  - Find a running process by process name
  - Ignore any errors
  - Hold the result in registry variable
  - Use **`shell`** module and run kill command to kill the registered variable



```
# vim killprocess.yml

---
- name: Find a process and kill it
  hosts: 10.253.1.115

  tasks:
    - name: Get running processes from remote host
      ignore_errors: yes
      shell: "ps -few | grep top | awk '{print $2}'"
      register: running_process

    - name: Kill running processes
      ignore_errors: yes
      shell: "kill {{ item }}"
      with_items: "{{ running_process.stdout_lines }}"
```

Run the playbook

```
# anisble-playbook killprocess.yml
```

By: Imran Afzal
www.utclisolutions.com

Pick and Choose Steps

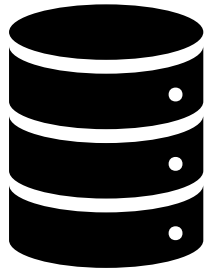
- Start a playbook at a specific task

```
# anisble-playbook yamlfile.yml --start-at-task 'Task name'
# anisble-playbook http.yml --start-at-task 'Intall telnet'
```

```
---
- name: httpd and telnet
  hosts: all

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

  - name: Start httpd
    service:
     name: httpd
     state: started

  - name: Install telnet
    yum:
     name: telnet
     state: present
```

By: Imran Afzal
www.utclisolutions.com

Create and Mount New Storage



- To create a new storage, we will power-off the VM add new disk (2GiB) from our virtualization software
- Also "**parted**" and "**mount**" module will be used in Ansible playbook
- Some Ansible distribution does not come with parted and mount module
  - Install **parted** and **mount** module
  - **ansible-galaxy collection install community.general**
  - **ansible-galaxy collection install ansible.posix**

```
ERROR! couldn't resolve module/action 'mount'. This often indicates
a misspelling, missing collection, or incorrect module path.
```

By: Imran Afzal
www.utclisolutions.com

Create and Mount New Storage

```
# vim newstorage.yml
---
- name: Create and mount new storage
  hosts: all

  tasks:
      - name: create new partition
        parted:
              name: files
              label: gpt
              device: /dev/sdb
              number: 1
              state: present
              part_start: 1MiB
              part_end: 1GiB
      - name: Create xfs filesystem
        filesystem:
              dev: /dev/sdb1
              fstype: xfs
      - name: Create mount directory
        file:
              path: /data
              state: directory
      - name: mount the filesystem
        mount:
              src: /dev/sdb1
              fstype: xfs
              state: mounted
```

1

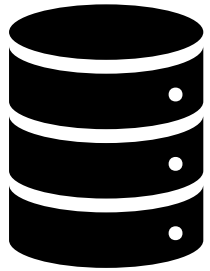
2

3

4

Run the playbook

```
# anisble-playbook newstorage.yml
```



By: Imran Afzal
www.utclisolutions.com