Ansible

What is Ansible?

Ansible is an agentless automation system developed by RedHat, designed for deploying changes across any number of machines.

Installing Ansible (Management Server)

The Ansible package only needs to be installed on the server pushing Ansible requests. It does not need to be installed on client/receiving servers.

```
apt install ansible
```

Configuring Ansible w/ Basic Examples

To use Ansible, you'll need to create a YAML file for Ansible to read from:

```
vim inventory.yaml
```

Basic YAML file example:

inventory.yaml

```
all:  # Define a section named "all"
  hosts:  # Define a section for target hosts within "all"
    server_name: ansible_host: IP_IP_IP_IP  # Host entry with label "server_name" (replace IP with actual server IP)
    server2_name: ansible_host: IP_IP_IP_IP  # Another host entry with label "server2_name" (replace IP with actual server2 IP)
  vars:  # Define a section for variables
    ansible_connection: ssh  # Specify SSH connection type
    ansible_ssh_user: ssh_username  # Define username for SSH authentication
    ansible_ssh_private_key_file: /path/to/key/on/ansible/server  # Define path to private key
```

```
file for SSH authentication on Ansible
```

This configuration doesn't make any automated changes to hosts, instead, it's just defining what the hosts are, and how to authenticate to them.

Usage example A:

Using this configuration, we can run the command below to ping all hosts in the YAML file:

```
ansible -i inventory.yaml all -m ping
```

The `-i` flag, also written as `--inventory`, is used with the `ansible` command to specify the inventory file that defines the target hosts for your Ansible playbooks.

`-m ping`: This flag specifies the module to be executed on the target hosts. In this case, `-m ping` (or `--module ping`) tells Ansible to run the "ping" module.

Usage example B:

The below example will check the installed kernel version for all hosts defined under 'all' in the inventory.yaml file:

```
ansible -i inventory.yaml all -a "/usr/bin/uname -a"
```

`-a` or `--module-args`: This flag specifies arguments to be passed to a module. However, in this case, it's not using a module but directly executing a shell command.

Ansible Playbooks

What is an Ansible Playbook?

An Ansible playbook is a blueprint or recipe that defines a series of automated tasks to be executed on one or more remote machines.

Basic example A:

Sticking with the inventory.yaml file created in the above example, I now want to create a 'playbook' for Ansible to run through and ping all defined hosts:

ping.yaml:

```
---  # Start of YAML document
- name: Test Ping  # Playbook name - describes the purpose
  hosts: all  # Target all hosts defined in the inventory
    tasks:  # Define tasks to be executed on target hosts
      - action: ping  # Execute the "ping" module to check connectivity
```

Now that we have our playbook, we can use this with the inventory file to initiate action against all defined hosts:

```
ansible-playbook -i inventory.yaml ping.yaml
```

- **`ansible-playbook`**: This is the specific command used to run Ansible playbooks.
- **`-i inventory.yaml`**: This flag specifies the inventory file, referencing your `inventory.yaml` file that defines the target hosts. The `-i` flag is also known as `--inventory`.
- **`ping.yaml`**: This is the filename of the Ansible playbook you want to execute. It likely contains instructions for the tasks you want to perform on the target hosts.

Basic example B:

Whilst still basic, this example is a tad more complex. Here the goal is to install the apache2 and nginx packages on the remote machines using APT.

install-software.yaml:

```
- hosts: client_server
  become: 'yes'  #enabled sudo privileges
  tasks:
    - name: Install software
      apt:
        pkg:
          - apache2
          - nginx
        state: present

- hosts: client_server  # Target host group named "client_server"
  become: 'yes'  # Enable sudo privileges for tasks (requires passwordless sudo)
  tasks:  # Define tasks to be executed on the target host(s)
    - name: Install software  # Task name - describes the purpose
      apt:  # Use the apt module for package management (assuming Debian/Ubuntu)
        pkg:  # Define the package(s) to be installed
          - apache2  # Install the apache2 web server package
```

```
      - nginx  # Install the nginx web server package
    state: present  # Ensure the packages are installed (present)
```

To execute this against the hosts defined in our inventory.yaml file:

```
ansible-playbook -i inventory.yaml install-software.yaml
```

More advanced example

Myself and Wajahat needed to remove existing software on 2 servers, and install different software versions.

```
- hosts: server_name server2_name
  tasks:
    - name: Add Elasticsearch GPG key
      ansible.builtin.apt_key:
        url: https://artifacts.elastic.co/GPG-KEY-elasticsearch
        state: present

    - name: Add Elasticsearch Repository
      ansible.builtin.apt_repository:
        repo: 'deb [arch=amd64] https://artifacts.elastic.co/packages/7.x/apt stable main'
        state: present

    - name: Add MariaDB GPG key
      ansible.builtin.apt_key:
        url: https://mariadb.org/mariadb_release_signing_key.pgp
        state: present

    - name: Add MariaDB Repository
      ansible.builtin.apt_repository:
        repo: 'deb [arch=amd64,arm64,ppc64el] https://deb.mariadb.org/10.4/ubuntu focal main'
        state: present

    - name: Remove Software
      apt:
        pkg:
          - redis-server
          - redis-tools
          - elasticsearch
```

```
        - percona-*
      state: absent
      purge: yes
  - name: Install desired Redis tools version
    apt:
      pkg:
        - redis-tools=6:6.0.20-1rl1~focal1
        - redis-server=6:6.0.20-1rl1~focal1
        - elasticsearch=7.9.0
        - mariadb-server
      state: present
      install_recommends: yes
```

This adds specific repositories and keys for MariaDB and Elasticsearch, removes any existing installations, and then installs a specific version from the repository.

Revision #13
Created 2024-06-05 17:10:31 UTC by Daniel
Updated 2026-09-11 11:10:27 UTC by Daniel