Ansible Research

#tech #aut #programming #howto #srsRAN #4GLTE #RnD

How it works:

I will be using my laptop as the control node (the machine that is running Ansible). The two Ubuntu laptops that Kat has secured will be the managed nodes (the machine that Ansible is instructing) and do not require Ansible to be installed, but require Python to run Ansible-generated code. The managed node also needs a user account that can connect through SSH to the node with an interactive POSIX shell (such as bash, or zsh).

So, to get Ansible to automate the Docker install (which contains the srsRAN install, config and program) we would need the following stack:

```
Edward's laptop running Ansible (we could also use a pre-config cloud VM)

+

Ubuntu laptop with Python installed with a SSH-capable user account already set up and
ready to go
```

THEN

```
Ansible tells Ubuntu to retrieve and run the Docker container from Github
```

THEN

```
The Docker container is a sealed environment that has the required dependencies for srsRAN
pre-installed - or it can download them as it sets up. We need to check which of these
would be the best method. We could also set it up so that srsRAN starts running once
installed, if that is preferable. The less CLI required from the end user the easier it
will be for them.
```

THEN

```
The user now has srsRAN ready to go.
```

Install and Usage:

Official documentation says Ansible needs to be installed with `pip`, not `homebrew`. Lots of information online about how installing it globally with `homebrew` can cause issues in the future.

Running `pip` with `sudo` will make global changes to the system. Since `pip` does not coordinate with system package managers, it could make changes to your system that leaves it in an inconsistent or non-functioning state. This is particularly true for macOS. Installing with `--user` is recommended.

Depending on your shell type some of the following commands will be different. These are for brew, but I have since switched to zsh and there are a few adjustments. Check what you are using before proceeding.

In my situation installing it with the official `homebrew` package seemed to work fine. I used the following:

```
brew update
brew install ansible

#Validate Ansible:
ansible ---version
```

Output (Ansible install successful, no conflicts):

```
Edwards-MacBook-Pro-3:~ edwardkeith$ ansible --version

ansible [core 2.17.1]

  config file = None

  configured module search path = ['/Users/edwardkeith/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']

  ansible python module location =
/usr/local/Cellar/ansible/10.1.0/libexec/lib/python3.12/site-packages/ansible

  ansible collection location =
/Users/edwardkeith/.ansible/collections:/usr/share/ansible/collections

  executable location = /usr/local/bin/ansible

  python version = 3.12.4 (main, Jun  6 2024, 18:26:44) [Clang 15.0.0 (clang-1500.3.9.4)]
(/usr/local/Cellar/ansible/10.1.0/libexec/bin/python)

  jinja version = 3.1.4

  libyaml = True

Edwards-MacBook-Pro-3:~ edwardkeith$
```

You can also add shell command completion to Ansible to make things easier:

```
activate-global-python-argcomplete --user
```

Playbook:

```
#Ansible-playbook for srsRAN setup

#
```

```
#Explanation:

#

#   Python Installation: The raw module is used to check if Python is installed and install
it if missing.

#   Docker Installation: Docker dependencies, GPG key, and repository are added before
installing Docker itself.

#   Docker Pull & Run: The docker_image module pulls the required srsRAN Docker image, and
docker_container ensures it is running.

#

#Steps to Execute:

#

#    Save the playbook as srsran_setup.yml.

#    Define the inventory file (inventory.ini) with the IPs or hostnames of the Ubuntu
laptops.

#    Run the playbook with the following command (zsh):

#

#                   ansible-playbook -i inventory.ini srsran_setup.yml

#

#Notes:

#

#    Replace <docker_image_name> with the actual Docker image name for the srsRAN
container.

#    Adjust any paths, repositories, or configurations to match the environment.

#    Ensure that the SSH user on the Ubuntu laptops has the necessary sudo privileges.



---

- hosts: all

  become: yes  # This ensures commands requiring root are executed with sudo

  tasks:

    - name: Ensure Python is installed

      raw: |

        if ! command -v python3 &>/dev/null; then

          apt-get update
```

```
      apt-get install -y python3
    fi

- name: Install Docker dependencies
  apt:
    name:
      - apt-transport-https
      - ca-certificates
      - curl
      - software-properties-common
    state: present
    update_cache: yes

- name: Add Docker GPG key
  apt_key:
    url: https://download.docker.com/linux/ubuntu/gpg
    state: present

- name: Add Docker repository
  apt_repository:
    repo: deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable
    state: present

- name: Install Docker
  apt:
    name: docker-ce
    state: latest
    update_cache: yes

- name: Start and enable Docker service
  systemd:
```

```
        name: docker
        enabled: yes
        state: started

    - name: Pull srsRAN Docker container
      docker_image:
        name: <docker_image_name>  # Replace with the Docker image name for srsRAN
        source: pull

    - name: Run srsRAN container
      docker_container:
        name: srsran
        image: <docker_image_name>  # Replace with the Docker image name for srsRAN
        state: started
        restart_policy: always
        volumes:
          - /path/to/config:/config  # Adjust the path to match configuration files

    - name: Ensure srsRAN is running
      shell: docker ps | grep srsran
      register: result

    - debug:
        msg: "{{ result.stdout }}"
```

inventory.ini file:

```
[testbed_nodes]
laptop1 ansible_host=192.168.1.101 ansible_user=your_ssh_username
laptop2 ansible_host=192.168.1.102 ansible_user=your_ssh_username
[all:vars]
ansible_python_interpreter=/usr/bin/python3
```

Execute:

```
ansible-playbook -i inventory.ini srsran_setup.yml
```

Important notes or data:

Ansible Community Documentation - Installation