Deployment Documentation for "deploy_ur_applications"

Auto-gen

Oct 01, 2024

README

CHAPTER

ONE

DEPLOYMENT OVERVIEW

This documentation is about deploying a software system consisting of

- bt_manipulation_framwork_core
- moveit_skill_server
- control_ur_gripper_via_io_skill
- realsense_camera_calibrated
- aruco_marker_detection_skill_server
- ur_control

to a target environment including devices as follows:

- cmh_tower_400 (type: PC)
- arm (type: UR5E)
- gripper (type: EGP50)
- camera_realsense (type: Realsense_camera)

The sources of components are as follows:

CHAPTER

TWO

TARGET ENVIRONMENT CONFIGURATION

2.1 Check device configurations

- Configured properties for Device: “cmh_tower_400”:

Table 1: Configured properties for Device: “cmh_tower_400”

property name	value
os_name	ubuntu
os_version	focal
processor_architecture	x86

- Configured properties for Device: “arm”:

Table 2: Configured properties for Device: “arm”

property name	value
device_type	ur5e

- Configured properties for Device: “gripper”:

  There are no properties defined for Device: “gripper”

- Configured properties for Device: “camera_realsense”:

Table 3: Configured properties for Device: “camera_realsense”

property name	value
device_type	d435
device_name	camera
serial_number	920312073127

2.2 Check communication connection between devices

- arm, gripperand are connected via **Io_tip**
- arm, cmh_tower_400and are connected via **Ethernet**
- cmh_tower_400, camera_realsenseand are connected via **Usb_01**

Check configurations for Device: “cmh_tower_400” are as follows:

For Ethernet:

```
IP address: 192.168.56.1
Subnet mask: 255.255.255.0
gateway: 192.168.56.1
DNS server: 0.0.0.0
```

For USB_01:

```
Volumes:
  - /dev/video0
  - /dev/video1
  - /dev/video2
  - /dev/video3
  - /dev/video4
  - /dev/video5
```

Check configurations for Device: “arm” are as follows:

For Ethernet:

```
IP address: 192.168.56.2
Subnet mask: 255.255.255.0
gateway: 192.168.56.1
DNS server: 0.0.0.0
```

Check configurations for Device: “gripper” are as follows:

Check configurations for Device: “camera_realsense” are as follows:

CHAPTER

THREE

SETUP ANSIBLE

Ansible automates the management of remote systems and controls their desired state. A basic Ansible environment has three main components:

Control node A system on which Ansible is installed. You run Ansible commands such as `ansible` or `ansible-inventory` on a control node.

Managed node A remote system, or host, that Ansible controls.

Inventory A list of managed nodes that are logically organized. You create an inventory on the control node to describe host deployments to Ansible.

If you did not install Ansible yet, you can install it as follows:

1. Install Ansible:

```
python3 -m pip install --user ansible
```

2. Fill an inventory by adding the IP address, username (default `anisble`) and also add IP address in `/etc/ansible/hosts`. The following example adds the IP addresses of three virtual machines in KVM:

```

```

   192.168.1.11 ansible_user=user_name

3. Verify the hosts in your inventory.

```
ansible all --list-hosts
```

4. Set up SSH connections so Ansible can connect to the managed nodes.

   a. Add your public SSH key to the `authorized_keys` file (path: `.ssh/authorized_keys`) on each remote system as follow:

      - cmh_tower_400

   b. Test the SSH connections, for example:

```
ssh username@192.0.2.50
```

   If the username on the control node is different on the host, you need to pass the `-u` option with the `ansible` command.

5. Ping the managed nodes.

```
ansible all -m ping
```

6. Add and encrypt “sudo passward”

To run containers with docker, we need to ensure docker is installed in each remote system. Therefore, it needs ''sudo password" to perform that. ''sudo password" will be encrypted!

To do so, you can run:

```
  ansible-vault create vars/cmh_tower_400/passwords.yml


It requires you to provide a password that you will use to edit your passwords.

After creating files, now you provide your ''sudo password'' by
```

```
  ansible-vault edit vars/cmh_tower_400/passwords.yaml


In each file, you need to enter as follows:
```

```
cmh_tower_400_sudo: "sudo password"
```

7. Start application.

```
ansible-playbook -i inventory.yaml playbook.yaml --ask-vault-pass --extra-vars
↪'@vars/cmh_tower_400/passwords.yml' --verbose
```