Ansible/Dify

- **Ansible**是一个自动化工具, 主要用于 **配置管理\应用程序部署\任务自动化**. 使用SSH连接到目标服务器, 以声明式方式描述系统状态.
  - 使用YAML编写PlayBook
  - 可以编写自定义模块扩展功能
  - 实现自动化部署和测试
- **Dify**是一个开源LLM应用开发平台, 旨在**简化基于LLM的应用程序的创建部署及维护**, 提供可视化界面, 允许用户通过简单拖拽和配置构建复杂LLM应用.
  - 支持多种LLM模型
  - 提供知识库管理功能
  - 允许用户制作插件和API扩展功能

1. **安装Ansible**
   - 以源主机IP <172.20.88.178>, 目标主机IP <172.20.88.220> 为例, 文件位置/etc/ansible
   - `apt install ansible` 安装ansible
   - `ansible -m ping target_hosts` 测试是否联通
     - 其中 **target_hosts** 为自己取名

```
// 1.Ansible默认需要ssh连接，不用ssh需要额外下载安装包sshpass
apt install sshpass

// 在/etc/ansible/hosts中添加如下信息
[target_hosts] # 定义一个主机组，可以自定义名称
172.20.88.220 ansible_user=your_name ansible_password=your_password
ansible_ssh_common_args='-o StrictHostKeyChecking=no'


// 2.使用ssh连接
ssh-keygen -t rsa
/*
在控制节点172.20.88.178生成ssh密钥对
生成文件 id_rsa(私钥), id_rsa.pub(公钥) 在 ~/.ssh/
*/
ssh-copy-id username@172.20.88.220 //其中username为目标主机用户名

//
[target_hosts] # 定义一个主机组，可以自定义名称
172.20.88.220 ansible_user=your_name
ansible_ssh_private_key_file=~/.ssh/id_rsa
```

2. **修改python默认版本**
   - `ls /usr/bin/python*` 展示python已有版本
   - `vi ~/.bashrc` 编辑文件
   - 在末尾加入你需要的版本`alias python='/usr/bin/python3'`
   - `source ~/.bashrc`

3. **部署Docker Compose, Dify**

   - 部署dify, 项目地址: **https://github.com/langgenius/dify.git**
   - 将docker-compose.yml文件与其他相关配置文件复制 **/etc/ansible/projects/dify_deploy/**
     - .env.example 改名为.env
     - ansible项目地址可以按自己需求更改
     - 需要自己配置nginx, 则将nginx文件夹也scp进去
   - Docker Compose离线部署

```
// 1.下载docker-compose的二进制版本
curl -L
"https://github.com/docker/compose/releases/download/${VERSION}/docker-
compose-$(uname -s)-$(uname -m)" -o docker-compose

//scp传输至离线机器
scp docker-compose user@target_machine:/usr/local/bin/

//设置权限
sudo chmod +x /usr/local/bin/docker-compose

//检查
docker-compose --version
//运行
docker-compose up


//2. docker-compose-插件安装
//docker官方GPG密钥
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor
-o /usr/share/keyrings/docker-archive-keyring.gpg

//添加官方源
echo "deb [arch=$(dpkg --print-architecture) signed-
by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo
tee /etc/apt/sources.list.d/docker.list > /dev/null

//更新软件包
sudo apt update
//安装
sudo apt-get install --download-only docker-compose-plugin
//scp传输至离线机器
scp docker-compose user@target_machine:/usr/local/bin/
```

   - **deploy_dify.yml文件**

```
---
- hosts: target_hosts
become: true
tasks:
    - name: Check if Docker is installed
    command: docker --version
    register: docker_check
    ignore_errors: true  # Docker 安装检查的

    - name: Install Docker if not installed
    block:
        - name: Install prerequisite packages
        apt:
            name: ['apt-transport-https', 'ca-certificates', 'curl', 'gnupg-
agent', '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
{{ ansible_distribution_release }} stable
            state: present
        - name: Update apt cache
        apt:
            update_cache: yes
        - name: Install Docker Engine
        apt:
            name: ['docker-ce', 'docker-ce-cli', 'containerd.io']
            state: present
    when: docker_check.rc != 0

    - name: Check if Docker Compose is installed  # Docker Compose安装检查
    command: docker-compose --version
    register: docker_compose_check
    ignore_errors: true

    - name: Install Docker Compose if not installed
    apt:
        name: docker-compose
        state: present
    when: docker_compose_check.rc != 0 and ansible_distribution == "Ubuntu"

    - name: Create Dify deployment directory
    file:
        path: /opt/dify
        state: directory
        mode: '0755'

    - name: Copy Dify deployment files
```

```
    copy:
        src: ./dify_deploy/
        dest: /opt/dify/
        owner: <username>   # [修改为目标用户名]
        group: <username>   # [修改为目标用户组]

    - name: Start Dify containers
    community.docker.docker_compose_v2:
        project_src: /opt/dify/
        state: present

    - name: Verify Dify containers are running
    community.docker.docker_compose_v2:
        project_src: /opt/dify/
        state: present
    register: docker_compose_output

    - debug:
        msg: "Docker Compose Output: {{ docker_compose_output }}"
```

4. dify启动

   - 配置文件 /etc/docker/daemon.json [172.20.88.220]

```
{
"registry-mirrors":
    [
        "https://docker.m.daocloud.io/",
        "https://huecker.io/",
        "https://dockerhub.timeweb.cloud",
        "https://noohub.ru/",
        "https://dockerproxy.com",
        "https://docker.mirrors.ustc.edu.cn",
        "https://docker.nju.edu.cn",
        "https://xx4bwyg2.mirror.aliyuncs.com",
        "http://f1361db2.m.daocloud.io",
        "https://registry.docker-cn.com",
        "http://hub-mirror.c.163.com",
        "https://docker.mirrors.ustc.edu.cn"
    ],
"dns": ["8.8.8.8", "8.8.4.4", "114.114.114.114"]
}
//重启服务
systemctl daemon-reload
systemctl restart docker
//查看docker详细日志
sudo journalctl -u docker.service -xe
//查看端口占用
netstat -tulnp | grep :443
```

- 配置文件 /etc/resolv.conf [172.20.88.220]
  - `nameserver 8.8.8.8` 加入公共DNS
  - `systemctl restart systemd-resolved.service` 重新加载
- 在ansible控制节点[172.20.88.178]执行**Playbook** [/etc/ansible/ansible_*]:
  - `ansible-playbook -i /etc/ansible/hosts deploy_dify.yml -e ansible_python_interpreter=/usr/bin/python2.7 [获取详细日志信息+: -vvvv]`

- **Notice**
  - 只支持python2.7 及 python3.8版本及以上, 系统内拥有多个版本, 建议显式指定需要的python版本.
  - docker-compose 及 ansible的版本兼容存在问题