

nsd_1902_ansible_project

项目目标

1. 通过web页显示所有主机的信息
2. 在网页中可以注册主机和组
3. 在网页中可以注册ansible模块和参数
4. 通过网页可以实现对远程主机/组的管理
5. 实现ansible动态主机清单

URL规划

<http://x.x.x.x/> 用于显示所有的功能

<http://127.0.0.1/webadmin/> 显示所有服务器的主机信息

<http://127.0.0.1/webadmin/addhosts/> 显示、添加主机/组

<http://127.0.0.1/webadmin/addmodules/> 显示、添加模块和参数

<http://127.0.0.1/webadmin/tasks/> 在主机/组执行任务

创建ansible项目

- 通过pycharm创建名为myansible的项目
- 创建mainpage和webadmin的应用

```
(nsd1902) [root@room8pc16 myansible]# python manage.py startapp mainpage
(nsd1902) [root@room8pc16 myansible]# python manage.py startapp webadmin
```

- 修改配置文件

```
# myansible/settings.py
ALLOWED_HOSTS = '*'
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'webadmin',
    'mainpage',
]
LANGUAGE_CODE = 'zh-hans'
TIME_ZONE = 'Asia/Shanghai'
USE_TZ = False
STATICFILES_DIRS = [
```

```
os.path.join(BASE_DIR, "static"),  
]
```

- 授权

```
# myansible/urls.py  
from django.conf.urls import url, include  
from django.contrib import admin  
  
urlpatterns = [  
    url(r'^admin/', admin.site.urls),  
    url(r'^webadmin/', include('webadmin.urls')),  
    url(r'', include('mainpage.urls')),  
]  
  
# mainpage/urls.py  
from django.conf.urls import url  
  
urlpatterns = [  
  
]  
  
# webadmin.urls.py  
from django.conf.urls import url  
  
urlpatterns = [  
  
]
```

mainpage应用

1.url

```
# mainpage/urls.py  
from django.conf.urls import url  
from . import views  
  
urlpatterns = [  
    url(r'^$', views.mainpage, name='mainpage'),  
]
```

2. 视图函数

```
# mainpage/views.py
from django.shortcuts import render

# Create your views here.

def mainpage(request):
    return render(request, 'mainpage.html')
```

3. 模板文件

```
# templates/mainpage.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Ansible Webadmin</title>
</head>
<body>
<h1>Ansible Webadmin</h1>
</body>
</html>
```

4. 启动开发服务器，测试首页

```
(nsd1902) [root@room8pc16 myansible]# python manage.py runserver 0:80
```

5. 引入bootstrap：将static目录拷贝到项目目录下，并修改网页头部信息

webadmin应用

管理模型

```
# webadmin/models.py
from django.db import models

# Create your models here.

class HostGroup(models.Model):
    groupname = models.CharField(max_length=100, unique=True)

    def __str__(self):
        return self.groupname

class Host(models.Model):
    hostname = models.CharField(max_length=100, unique=True)
    ipaddr = models.CharField(max_length=15)
    group = models.CharField(HostGroup)

    def __str__(self):
```

```

        return '%s=>%s' % (self.group, self.hostname)

class Module(models.Model):
    module_name = models.CharField(max_length=100, unique=True)

    def __str__(self):
        return self.module_name

class Args(models.Model):
    arg_text = models.CharField(max_length=100)
    module = models.ForeignKey(Module)

    def __str__(self):
        return '%s=>%s' % (self.module, self.arg_text)

```

生成数据库中的表

```

(nsd1902) [root@room8pc16 myansible]# python manage.py makemigrations
(nsd1902) [root@room8pc16 myansible]# python manage.py migrate

```

项目的根目录下的db.sqlite3是数据库文件。sqlite数据库是文件型数据库，一个文件就是一个库。

```

[root@room8pc16 myansible]# sqlite3 db.sqlite3
sqlite> .help    # 查看帮助
sqlite> .table   # 显示所有的表
sqlite> .schema webadmin_host # 查看表结构
sqlite> select * from webadmin_host;

```

创建管理员用户

```

(nsd1902) [root@room8pc16 myansible]# python manage.py createsuperuser

```

将模型注册到后台管理界面

```

# webadmin/admin.py
from django.contrib import admin
from .models import HostGroup, Host, Module, Args

# Register your models here.

for item in [HostGroup, Host, Module, Args]:
    admin.site.register(item)

```

登录到<http://x.x.x.x/admin>进行管理

准备ansible工作环境

```
(nsd1902) [root@room8pc16 myansible]# mkdir ansicfg
(nsd1902) [root@room8pc16 myansible]# vim ansicfg/ansible.cfg
[defaults]
inventory = dhosts.py
remote_user = root

(nsd1902) [root@room8pc16 myansible]# touch ansicfg/dhosts.py
(nsd1902) [root@room8pc16 myansible]# chmod +x ansicfg/dhosts.py
```

创建动态主机清单

ansible动态主机清单文件是一个脚本，脚本的输出要求是以下格式：

```
{
  '组1': {'hosts': ['主机1', '主机2']},
  '组2': {'hosts': ['主机1', '主机2']},
}

result = {}
result['webservers'] = {} # {'webservers': {}}
result['webservers']['hosts'] = [] # {'webservers': {'hosts': []}}
```

动态主机清单脚本：

```
#!/root/nsd1902/bin/python
import json
from sqlalchemy import create_engine, Column, Integer, String, Date, ForeignKey
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker

engine = create_engine(
    'sqlite:///var/ftp/nsd2019/nsd1902/devweb/ansible_pro/myansible/db.sqlite3',
    encoding='utf8',
)
Session = sessionmaker(bind=engine)
Base = declarative_base()

class HostGroup(Base):
    __tablename__ = 'webadmin_hostgroup'
    id = Column(Integer, primary_key=True)
    groupname = Column(String(100), unique=True)

class Host(Base):
    __tablename__ = 'webadmin_host'
    id = Column(Integer, primary_key=True)
    hostname = Column(String(100), unique=True)
    ipaddr = Column(String(15))
    group_id = Column(Integer, ForeignKey('webadmin_hostgroup.id'))

if __name__ == '__main__':
    result = {}
```

```

session = Session()
qset = session.query(HostGroup.groupname, Host.ipaddr).join(Host)
for group, ip in qset:
    if group not in result:
        result[group] = {}
    if not result[group]:
        result[group]['hosts'] = []
    result[group]['hosts'].append(ip)
print(json.dumps(result))

```

测试

```
(nsd1902) [root@room8pc16 ansicfg]# ansible all -m ping
```

制作webadmin应用的首页

1. URL

```

# webadmin/urls.py
from django.conf.urls import url
from . import views

urlpatterns = [
    url(r'^$', views.webadmin_index, name='webadmin_index'),
]

```

2. 视图函数

```

# webadmin/views.py
from django.shortcuts import render

# Create your views here.

def webadmin_index(request):
    return render(request, 'webadmin_index.html')

```

3. 生成模板文件

```

(nsd1902) [root@room8pc16 ansicfg]# ansible all -m setup --tree /tmp/output
(nsd1902) [root@room8pc16 ansicfg]# ansible-cmdb /tmp/output >
../templates/webadmin_index.html

```

注意：模板文件是静态的，它反映的只是执行命令那个时间点的服务器情况。如果需要获取相对实时的信息，可以把上面的命令放到计划任务中，如每隔2小时执行一次。

4. 修改mainpage.html中“主机信息”的超链接

```
<a href="{% url 'webadmin_index' %}" target="_blank">
```

5. 验证

制作添加主机页

1. URL

```
# webadmin/urls.py
from django.conf.urls import url
from . import views

urlpatterns = [
    url(r'^$', views.webadmin_index, name='webadmin_index'),
    url(r'^addhosts/$', views.add_hosts, name='add_hosts'),
]
```

2. 视图函数

```
# webadmin/views.py
from .models import HostGroup
def add_hosts(request):
    groups = HostGroup.objects.all()
    return render(request, 'addhosts.html', {'groups': groups})
```

3. 模板文件

```
# templates/addhosts.html
{% extends 'base.html' %}
{% load static %}
{% block title %}添加主机/组{% endblock %}
{% block content %}
    <div class="col-sm-12">
        <form action="" method="post" class="form-inline">
            {% csrf_token %}
            <div class="form-group">
                <label>主机组 : </label>
                <input type="text" class="form-control" name="group">
            </div>
            <div class="form-group">
                <label>主机 : </label>
                <input type="text" class="form-control" name="host">
            </div>
            <div class="form-group">
                <label>IP地址 : </label>
                <input type="text" class="form-control" name="ip">
            </div>
            <div class="form-group">
                <input class="btn btn-primary" type="submit" value="提交">
            </div>
        </form>
    </div>
    <hr>
    <div class="col-sm-12">
        <table class="table table-hover table-striped table-bordered">
```

```

<thead class="bg-primary text-center">
    <tr>
        <td>主机组</td>
        <td>主机</td>
    </tr>
</thead>
<tbody>
    {% for group in groups %}
        <tr>
            <td>{{ group.groupname }}</td>
            <td>
                <ul class="list-unstyled">
                    {% for host in group.host_set.all %}
                        <li>{{ host.hostname }}: {{ host.ipaddr }}</li>
                    {% endfor %}
                </ul>
            </td>
        </tr>
    {% endfor %}
</tbody>
</table>
</div>
{% endblock %}

```

4. 修改mainpage.html中“添加主机”的超链接

```

<a href="{% url 'add_hosts' %}" target="_blank">

```

5. 修改函数，实现添加主机的功能

```

# webadmin/views.py

def add_hosts(request):
    if request.method == 'POST':
        group = request.POST.get('group').strip()
        host = request.POST.get('host').strip()
        ip = request.POST.get('ip').strip()
        if group:
            # get_or_create返回元组：(组实例, 0/1)
            g = HostGroup.objects.get_or_create(groupname=group)[0]
            if host and ip:
                g.host_set.get_or_create(hostname=host, ipaddr=ip)

    groups = HostGroup.objects.all()
    return render(request, 'addhosts.html', {'groups': groups})

```