1. 功能分析
2. 显示所有服务器的信息
3. 添加主机/主机组
4. 添加模块及参数
5. 执行任务
6. 创建myansible项目

**1、激活虚拟环境**

[root@gjq ~]# source /opt/djenv/bin/activate

**2、创建项目**

(djenv) [root@gjq ~]# cd /root/PycharmProjects/ansible\_project/

(djenv) [root@gjq ansible\_project]# django-admin startproject myansible

1. **创建应用**

(djenv) [root@gjq ansible\_project]# cd myansible/

(djenv) [root@gjq myansible]# python manage.py startapp webansi

**4、修改settings.py配置**

[root@gjq myansible]# vim myansible/settings.py

ALLOWED\_HOSTS = '\*'

INSTALLED\_APPS = [

... ...

'webansi',

]

LANGUAGE\_CODE = 'zh-hans'

TIME\_ZONE = 'Asia/Shanghai'File -> Settings ->

USE\_TZ = False

**5、配置Pycharm**

File -> Settings -> Project：ansible\_project -> Project Interpreter选虚拟环境

File -> Settings -> Languages&.. -> Django设置项目与settings.py

**6、URL规划**

http://x.x.x.x/ #显示各个功能模块

http://x.x.x.x/webansi/ #显示服务器信息

http://x.x.x.x/webansi/addhosts/ #添加主机/主机组

http://x.x.x.x/webansi/addmodules/ #添加模块及参数

http://x.x.x.x/webansi/del/1 #删除参数

http://x.x.x.x/webansi/tasks/ #执行任务

**7、URL授权**

[root@gjq myansible]# vim myansible/urls.py

from django.conf.urls import url, include

from django.contrib import admin

urlpatterns = [

... ...

url(r'^webansi/', include('webansi.urls')),

]

1. 编写webansi应用

**1、URL规划**

http://x.x.x.x/ #显示各个功能模块

http://x.x.x.x/webansi/ #主机信息

http://x.x.x.x/webansi/addhosts/ #添加主机/主机组

http://x.x.x.x/webansi/addmodules/ #添加模块及参数

http://x.x.x.x/webansi/del/1 #删除参数

http://x.x.x.x/webansi/tasks/ #执行任务

1. **创建modules模型**

**(1) 主机相关的模型**

webansi\_hostgroup表：id(主)、主机组

webansi\_host表：id(主)、主机名、IP地址、group\_id(外)

**(2) 模块相关模型**

webansi\_module表：id(主)、模块名

webansi\_argument表：id(主)、模块参数、module\_id(外)

[root@gjq myansible]# vim webansi/models.py

from django.db import models

class HostGroup(models.Model):

group\_name = models.CharField(max\_length=50, unique=True, null=False)

def \_\_str\_\_(self):

return self.group\_name

class Host(models.Model):

hostname = models.CharField(max\_length=50, unique=True)

ipaddr = models.CharField(max\_length=15)

group = models.ForeignKey(HostGroup)

def \_\_str\_\_(self):

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

class Module(models.Model):

module\_name = models.CharField(max\_length=50, unique=True, null=False)

def \_\_str\_\_(self):

return self.module\_name

class Argument(models.Model):

argument\_text = models.CharField(max\_length=100, null=False)

module = models.ForeignKey(Module)

def \_\_str\_\_(self):

return "%s => %s" % (self.module, self.argument\_text)

1. **生成splite3数据库**

本例采用的是文件型数据库，一个文件就是一个数据库。默认django在manage.py所在的目录下将会创建名为db.sqlite3的数据库

(djenv) [root@gjq myansible]# python manage.py makemigrations

(djenv) [root@gjq myansible]# python manage.py migrate

(djenv) [root@gjq myansible]# ls

db.sqlite3

(djenv) [root@gjq myansible]# sqlite3 db.sqlite3 #连接数据库

sqlite> .tables #show tables

sqlite> .schema webansi\_host #desc

**4、注册模型到后台**

[root@gjq myansible]# vim webansi/admin.py

from django.contrib import admin

from .models import HostGroup, Host, Module, Argument

for item in [HostGroup, Host, Module, Argument]:

admin.site.register(item)

1. **创建超级管理员、连接后台**

(djenv) [root@gjq myansible]# python manage.py createsuperuser

Username (leave blank to use 'root'): admin

Email address: 442425799@qq.com

Password:

Password (again):

Superuser created successfully.

(djenv) [root@gjq myansible]# python manage.py runserver 0:80

访问http://127.0.0.1/admin/

**6、克隆3台虚拟机**

node4.tedu.cn 192.168.1.4

node5.tedu.cn 192.168.1.5

node6.tedu.cn 192.168.1.6

配置yum源、免密登陆

1. **部署ansible工作目录**

**(1) 创建工作目录**

(djenv) [root@gjq myansible]# mkdir ansicfg

**(2) ansible配置文件**

(djenv) [root@gjq myansible]# vim ansicfg/ansible.cfg

[defaults]

inventory = dhosts.py

remote\_user = root

说明：主机信息存在数据库中，因此要使用动态主机，将inventory指定为脚本，运行后返回JSON格式的主机列表。

**(3) 编写dhosts.py，给x权限**

a. 为了能看到效果，在后台页面添加几台主机和组

b. dhosts.py运行后，输出的样式必是json格式，格式如下：

{'dbservers': {'hosts': ['192.168.1.4']},

'webservers': {'hosts': ['192.168.1.5', '192.168.1.6']}}

c. 编写dhosts.py

(djenv) [root@gjq myansible]# vim ansicfg/dhosts.py

#!/opt/djenv/bin/python

from sqlalchemy import create\_engine

from sqlalchemy.ext.declarative import declarative\_base

from sqlalchemy import Column, Integer, String, ForeignKey, Date

from sqlalchemy.orm import sessionmaker

import json

engine = create\_engine(

'sqlite:////root/PycharmProjects/ansible\_project/myansible/db.sqlite3',

encoding='utf8',

)

Base = declarative\_base()

Session = sessionmaker(bind=engine)

class HostGroup(Base):

\_\_tablename\_\_ = 'webansi\_hostgroup'

id = Column(Integer, primary\_key=True)

group\_name = Column(String(50), unique=True, nullable=False)

def \_\_str\_\_(self):

return self.group\_name

class Host(Base):

\_\_tablename\_\_ = 'webansi\_host'

id = Column(Integer, primary\_key=True)

hostname = Column(String(50), nullable=False)

ipaddr = Column(String(15))

group\_id = Column(Integer, ForeignKey('webansi\_hostgroup.id'))

if \_\_name\_\_ == '\_\_main\_\_':

session = Session()

query = session.query(HostGroup.group\_name, Host.ipaddr).join(Host)

#query结果格式：[('g1', '192.168.1.4'), ('g2', '192.168.1.5')...]

result = {}

for group, ip in query: #取出查询结果的组和IP地址

if group not in result: #构建result = {'g1': {}, 'g2': {}}

result[group] = {}

result[group]['hosts'] = []

result[group]['hosts'].append(ip)

print(json.dumps(result))

(djenv) [root@gjq myansible]# chmod +x ansicfg/dhosts.py

**(4) 测试ansible是否正常**

(djenv) [root@gjq myansible]# cd ansicfg/

(djenv) [root@gjq ansicfg]# ansible all --list-hosts

四、创建首页

URL：http://x.x.x.x/，写在项目的urls.py

视图函数：调用webansi应用的views.py

模板：放置在webansi应用的templates下

静态文件：放置在webansi应用的static下

1. **添加URLConfig**

(djenv) [root@gjq myansible]# vim myansible/urls.py

from django.conf.urls import url, include

from django.contrib import admin

from webansi import views

#django运行时默认以项目目录为根加载模块，在使用Pycharm导入模块时，

右击项目目录 -> Mark directory as -> Sources Root

urlpatterns = [

url(r'^admin/', admin.site.urls),

url(r'^webansi/', include('webansi.urls')),

url(r'^$', views.index, name='index')

]

1. **编写视图函数**

(djenv) [root@gjq myansible]# vim webansi/views.py

from django.shortcuts import render

def index(request):

return render(request, 'index.html')

1. **创建模板文件**

(djenv) [root@gjq myansible]# mkdir webansi/templates

(djenv) [root@gjq myansible]# touch webansi/templates/index.html

**4、引入bootstrap与其他静态文件**

(djenv) [root@gjq myansible]# cp -r /root/static webansi/

(djenv) [root@gjq myansible]# ls webansi/static/

css fonts imgs js

1. **制作基础模板**

(djenv) [root@gjq myansible]# vim webansi/templates/base.html

{% load static %} #红色字体：引入bootstrap

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>{% block title %}{% endblock %}</title> #黄色背景：个性信息

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">

</head>

<body>

<div class="container"> #页眉：词云轮播图

<div id="linux-carousel" class="carousel slide">

<ol class="carousel-indicators">

<li class="active" data-target="#linux-carousel" data-slide-to="0"></li>

<li data-target="#linux-carousel" data-slide-to="1"></li>

<li data-target="#linux-carousel" data-slide-to="2"></li>

</ol>

<div class="carousel-inner">

<div class="item active">

<a href="http://www.sogou.com" target="\_blank">

<img src="{% static 'imgs/first.png' %}" width="1200px">

</a>

</div>

<div class="item">

<img src="{% static 'imgs/second.png' %}" width="1200px">

</div>

<div class="item">

<img src="{% static 'imgs/third.png' %}" width="1200px" height="400px">

</div>

</div>

<a href="#linux-carousel" data-slide="prev" class="carousel-control left">

<span class="glyphicon glyphicon-chevron-left"></span>

</a>

<a href="#linux-carousel" data-slide="next" class="carousel-control right">

<span class="glyphicon glyphicon-chevron-right"></span>

</a>

</div>

<div class="main" style="padding-top: 50px">

{% block content %}{% endblock %}

</div>

#页脚：作者信息

<div class="footer text-center h3" style="padding-top: 50px">

<a href="#">作者：高锦全</a>&emsp;电话：18813756505&emsp;邮箱：jerry\_gao123@foxmail.com

</div>

</div>

<script src="{% static 'js/jquery.min.js' %}"></script>

<script src="{% static 'js/bootstrap.min.js' %}"></script>

<script type="text/javascript">

$('#linux-carousel').carousel({

interval : 3000

});

</script>

</body>

</html>

1. **继承模板：制作首页**

(djenv) [root@gjq myansible]# vim webansi/templates/index.html

{% extends 'base.html' %}

{% load static %}

{% block title %}首页{% endblock %}

{% block content %}

<div class="row">

<div class="col-md-3 text-center">

<a href="#">

<img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>

主机信息

</a>

</div>

<div class="col-md-3 text-center">

<a href="#">

<img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>

添加主机

</a>

</div>

<div class="col-md-3 text-center">

<a href="#">

<img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>

添加模块

</a>

</div>

<div class="col-md-3 text-center">

<a href="#">

<img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>

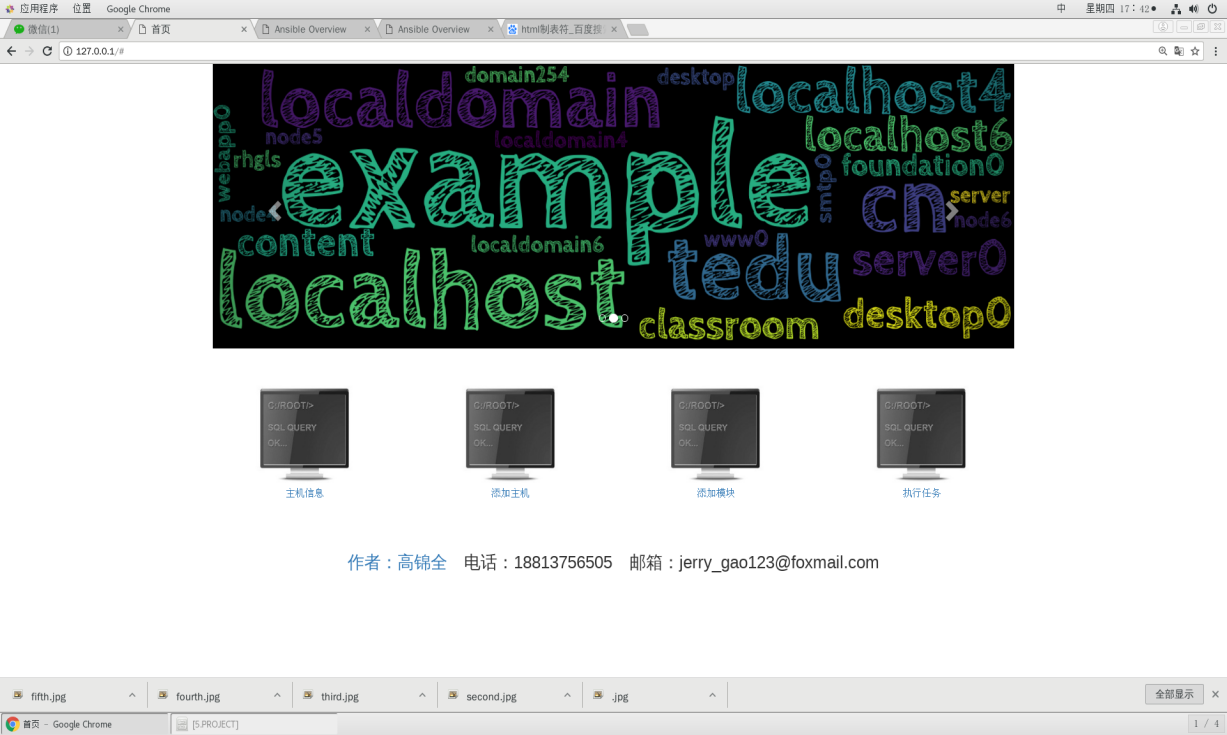
执行任务

</a>

</div>

</div>

{% endblock %}



五、制作"主机信息"页面

URL：http://x.x.x.x/webansi/

**1、添加URLConfig**

(djenv) [root@gjq myansible]# vim webansi/urls.py

from django.conf.urls import url

from webansi import views

urlpatterns = [

url(r'^$', views.mainpage, name='mainpage'),

]

1. **编写视图函数**

(djenv) [root@gjq myansible]# vim webansi/views.py

from django.shortcuts import render

def index(request):

... ...

def mainpage(request):

return render(request, "mainpage.html")

**3、制作模板文件**

**(1) ansible生成主机信息**

(djenv) [root@gjq myansible]# cd ansicfg/

(djenv) [root@gjq ansicfg]# ansible all -m setup --tree out

(djenv) [root@gjq ansicfg]# ls out/

192.168.1.4 192.168.1.5 192.168.1.6

**(2) 利用ansible-cmdb生成网页文件**

(djenv) [root@gjq ansicfg]# ansible-cmdb out/ > ../webansi/templates/mainpage.html

**(3) 编写计划任务：周期更新主机信息**

[root@gjq ~]# vim out.sh

#!/bin/bash

cd /root/PycharmProjects/ansible\_project/myansible/ansicfg/

ansible all -m setup --tree out

ansible-cmdb out/ > ../webansi/templates/mainpage.html

[root@gjq ~]# crontab -e -u root

\* \*/2 \* \* \* /bin/bash /root/out.sh

**(4) 修改主页超链接**

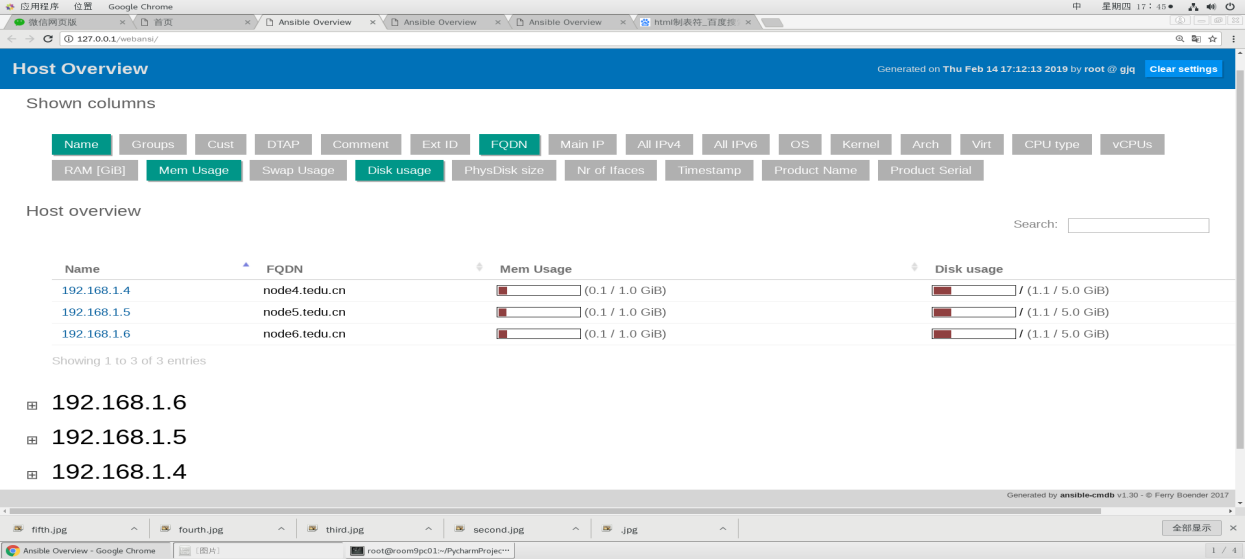
(djenv) [root@gjq myansible]# vim webansi/templates/index.html

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

<img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>

主机信息

</a>



1. 制作"添加主机/组"页面

URL：http://x.x.x.x/webansi/addhosts/

1. **添加URLConfig**

(djenv) [root@gjq myansible]# vim webansi/urls.py

from django.conf.urls import url

from webansi import views

urlpatterns = [

... ...

url(r'^addhosts/$', views.addhosts, name='addhosts'),

]

1. **视图函数与模板文件**

Part1：以表格形式显示所有主机和组信息

Part2：通过表单创建主机/组

**(1) Part1-编写视图函数**

(djenv) [root@gjq myansible]# vim webansi/views.py

from django.shortcuts import render

from webansi.models import HostGroup

... ...

def addhosts(request):

groups = HostGroup.objects.all()

return render(request, "addhosts.html", {'groups': groups})

**(2) Part1-创建模板文件**

(djenv) [root@gjq myansible]# vim webansi/templates/addhosts.html

{% extends 'base.html' %}

{% load static %}

{% block title %}添加主机/组{% endblock %}

{% block content %}

<div>

<table class="table table-bordered table-hover">

<tr class="info text-center h3">

<td>主机组</td>

<td>主机</td>

</tr>

{% for group in groups %}

<tr class="text-center h4">

<td>{{ group.group\_name }}</td>

<td>

<ul class="list-unstyled">

{% for host in group.host\_set.all %}

<li>

{{ host.hostname }}=>{{ host.ipaddr }}

</li>

{% endfor %}

</ul>

</td>

</tr>

{% endfor %}

</table>

</div>

{% endblock %}

**(3) 修改主页超链接**

(djenv) [root@gjq myansible]# vim webansi/templates/index.html

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

<img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>

添加主机

</a>

**(4) Part2-修改模板文件**

(djenv) [root@gjq myansible]# vim webansi/templates/addhosts.html

... ...

{% block content %}

<div>

<form action="" method="post" class="form-inline">

<div class="form-group">

<label>主机名: </label>

<input type="text" class="form-control" name="host">

&emsp;

</div>

<div class="form-group">

<label>ip地址: </label>

<input type="text" class="form-control" name="ip">

&emsp;

</div>

<div class="form-group">

<label>主机组: </label>

<input type="text" class="form-control" name="group">

</div>

<div class="form-group">

<input type="submit" class="btn btn-primary" value="提 交">

</div>

</form>

</div>

<hr>

<div>

... ... #此为Part1的html内容

</div>

{% endblock %}

注意：form action为空，表示post提交给自己(http://x.x.x.x/webansi/addhosts/)

**(5) Part2-修改视图函数**

(djenv) [root@gjq myansible]# vim webansi/views.py

... ...

def addhosts(request):

if request.method == 'POST':

host = request.POST.get('host').strip()

ip = request.POST.get('ip').strip()

group = request.POST.get('group').strip()

if group:

hostgroup = HostGroup.objects.get\_or\_create(group\_name=group)[0]

if host and ip:

hostgroup.host\_set.get\_or\_create(hostname=host, ipaddr=ip)

groups = HostGroup.objects.all()

return render(request, "addhosts.html", {'groups': groups})

#注意：get\_or\_create返回的是一个元组->(实例名, True|False)

**(6) Part2-由于csrf安全功能打开，在模板的form语句下添加语句**

(djenv) [root@gjq myansible]# vim webansi/templates/addhosts.html

... ...

<form action="" method="post" class="form-inline">

{% csrf\_token %}

... ...



1. 制作"模块/参数"页面

模块/参数URL：http://x.x.x.x/webansi/addmodules/

删除参数URL：http://x.x.x.x/webansi/del/1

1. **完成"添加模块/参数"URL功能**

**1.1添加URLConfig**

(djenv) [root@gjq myansible]# vim webansi/urls.py

from django.conf.urls import url

from webansi import views

urlpatterns = [

... ...

url(r'^addmodules/$', views.addmodules, name='addmodules'),

]

**1.2视图函数与模板文件**

Part1：以表格形式显示所有模块和参数

Part2：通过表单创建模块/参数

参考：“添加主机/组”页面的视图函数与模板文件

**(1) 编写视图函数**

(djenv) [root@gjq myansible]# vim webansi/views.py

from webansi.models import HostGroup, Module

... ...

def addmodules(request):

if request.method == 'POST':

module = request.POST.get('module').strip()

argument = request.POST.get('argument').strip()

if module:

moduleobj = Module.objects.get\_or\_create(module\_name=module)[0]

if argument:

moduleobj.argument\_set.get\_or\_create(argument\_text=argument)

modules = Module.objects.all()

return render(request, "addmodules.html", {'modules': modules})

**(2) 创建模板文件**

(djenv) [root@gjq myansible]# vim webansi/templates/addmodules.html

{% extends 'base.html' %}

{% load static %}

{% block title %}添加模块/参数{% endblock %}

{% block content %}

<div>

<form action="" method="post" class="form-inline">

{% csrf\_token %}

<div class="form-group">

<label>模块: </label>

<input type="text" class="form-control" name="module">

&emsp;

</div>

<div class="form-group">

<label>参数: </label>

<input type="text" class="form-control" name="argument">

&emsp;

</div>

<div class="form-group">

<input type="submit" class="btn btn-primary" value="提 交">

</div>

</form>

</div>

<hr>

<div>

<table class="table table-bordered table-hover">

<tr class="info text-center h3">

<td>模块</td>

<td>参数</td>

</tr>

{% for module in modules %}

<tr class="text-center h4">

<td>{{ module.module\_name }}</td>

<td>

<ul class="list-unstyled">

{% for argument in module.argument\_set.all %}

<li>{{ argument.argument\_text }}</li>

{% endfor %}

</ul>

</td>

</tr>

{% endfor %}

</table>

</div>

{% endblock %}

**(3) 修改主页超链接**

(djenv) [root@gjq myansible]# vim webansi/templates/index.html

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

<img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>

添加模块

</a>

1. **完成"删除参数"URL功能**

**2.1 添加URLConfig**

(djenv) [root@gjq myansible]# vim webansi/urls.py

from django.conf.urls import url

from webansi import views

urlpatterns = [

... ...

url(r'^del/(?P<arg\_id>\d+)/$', views.delarg, name='delarg'),

]

**2.2 视图函数与模板文件**

**(1) 编写视图函数**

(djenv) [root@gjq myansible]# vim webansi/views.py

from django.shortcuts import render, redirect

from webansi.models import HostGroup, Module, Argument

... ...

def delarg(request, arg\_id):

argument = Argument.objects.get(id=arg\_id)

argument.delete()

return redirect('addmodules')

**(2) 更新模板文件**

(djenv) [root@gjq myansible]# vim webansi/templates/addmodules.html

... ...

{% for module in modules %}

<tr class="text-center h4">

<td>{{ module.module\_name }}</td>

<td>

<ul class="list-unstyled">

{% for argument in module.argument\_set.all %}

<li>

<div class="col-md-10">{{ argument.argument\_text }}</div>

<div class="col-md-2"><a href="{% url 'delarg' arg\_id=argument.id %}">删除</a></div>

</li>

{% endfor %}

</ul>

</td>

</tr>

{% endfor %}



1. 制作"执行任务"页面

URL：http://x.x.x.x/webansi/tasks/

**1、完成"执行任务"URL跳转**

**1.1 添加URLConfig**

(djenv) [root@gjq myansible]# vim webansi/urls.py

from django.conf.urls import url

from webansi import views

urlpatterns = [

... ...

url(r'^tasks/$', views.tasks, name='tasks')

]

**1.2 编写视图函数**

(djenv) [root@gjq myansible]# vim webansi/views.py

... ...

def tasks(request):

return render(request, 'tasks.html')

**1.3 制作模板文件**

(djenv) [root@gjq myansible]# vim webansi/templates/tasks.html

{% extends 'base.html' %}

{% load static %}

{% block title %}执行任务{% endblock %}

{% block content %}

... ...

{% endblock %}

**1.4 修改主页超链接**

(djenv) [root@gjq myansible]# vim webansi/templates/index.html

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

<img src="{% static 'imgs/linux.jpg' %}" width="150px"><br>

执行任务

</a>

**2、完成"执行任务"前端页面显示**

**2.1 编写视图函数**

(djenv) [root@gjq myansible]# vim webansi/views.py

from django.shortcuts import render, redirect

from webansi.models import Host, HostGroup, Module, Argument

... ...

def tasks(request):

groups = HostGroup.objects.all()

hosts = Host.objects.all()

modules = Module.objects.all()

info = {'groups': groups, 'hosts': hosts, 'modules': modules}

return render(request, 'tasks.html', info)

#将后台数据库内容发送到前台

**2.2 编写模板文件**

(djenv) [root@gjq myansible]# vim webansi/templates/tasks.html

... ...

{% block content %}

<div>

<ul class="nav nav-tabs"> #选项卡

<li class="active"><a href="#host" data-toggle="tab">主机</a></li>

<li><a href="#hostgroup" data-toggle="tab">主机组</a></li>

</ul>

<form action="" method="post" style="padding-top: 10px">

<div class="tab-content">

<div class="tab-pane active fade in" id="host">

<select class="form-control" name="ip"> #下拉选项

<option value="">无</option>

{% for host in hosts %}

<option value="{{ host.ipaddr }}">

{{ host.hostname }}: {{ host.ipaddr }}

</option>

{% endfor %}

</select>

</div>

<div class="tab-pane fade" id="hostgroup">

<select class="form-control" name="group">

<option value="">无</option>

{% for group in groups %}

<option value="{{ group.group\_name }}">

{{ group.group\_name }}

</option>

{% endfor %}

</select>

</div>

</div>

<table class="table table-bordered table-hover">

<tr class="info text-center h3">

<td>模块</td>

<td>参数</td>

</tr>

{% for module in modules %}

<tr class="text-center h4">

<td>

<div class="radio">

<label><input type="radio" name="module" value="{{ module.module\_name }}">{{ module.module\_name }}</label>

</div>

</td>

<td>

{% for arg in module.argument\_set.all %}

<div class="radio">

<label><input type="radio" name="arg" value="{{ arg.argument\_text }}">{{ arg.argument\_text }}</label>

</div>

{% endfor %}

</td>

</tr>

{% endfor %}

</table>

<div class="form-group text-center">

<input class="btn btn-primary" type="submit" value="执 行">

</div>

</form>

</div>

{% endblock %}

**3、完成"执行任务"后端处理**

**3.1 准备myansible.py导入模块**

(djenv) [root@gjq myansible]# vim webansi/myansible.py

import shutil

from collections import namedtuple

from ansible.parsing.dataloader import DataLoader

from ansible.vars.manager import VariableManager

from ansible.inventory.manager import InventoryManager

from ansible.playbook.play import Play

from ansible.executor.task\_queue\_manager import TaskQueueManager

import ansible.constants as C

def ad\_hoc(inventory\_path=None, hosts=None, module=None, args=None):

Options = namedtuple('Options', ['connection', 'module\_path', 'forks', 'become', 'become\_method', 'become\_user', 'check', 'diff'])

options = Options(connection='smart', module\_path=['/to/mymodules'], forks=10, become=None, become\_method=None, become\_user=None, check=False, diff=False)

loader = DataLoader()

passwords = dict()

inventory = InventoryManager(loader=loader, sources=inventory\_path)

variable\_manager = VariableManager(loader=loader, inventory=inventory)

play\_source = dict(

name = "Ansible Play",

hosts = hosts,

gather\_facts = 'no',

tasks = [

dict(action=dict(module=module, args=args), register='shell\_out'),

# dict(action=dict(module='debug', args=dict(msg='{{shell\_out.stdout}}')))

]

)

play = Play().load(play\_source, variable\_manager=variable\_manager, loader=loader)

tqm = None

try:

tqm = TaskQueueManager(

inventory=inventory,

variable\_manager=variable\_manager,

loader=loader,

options=options,

passwords=passwords,

)

result = tqm.run(play)

finally:

if tqm is not None:

tqm.cleanup()

shutil.rmtree(C.DEFAULT\_LOCAL\_TMP, True)

**3.2编写视图函数**

(djenv) [root@gjq myansible]# vim webansi/views.py

from django.shortcuts import render, redirect

from webansi.models import Host, HostGroup, Module, Argument

from webansi.myansible import ad\_hoc

... ...

def tasks(request):

if request.method == 'POST':

ip = request.POST.get('ip')

group = request.POST.get('group')

module = request.POST.get('module')

argument = request.POST.get('arg')

if ip:

target = ip

else:

target = group

ad\_hoc(

inventory\_path=['ansicfg/dhosts.py'],

hosts=target,

module=module,

args=argument

)

groups = HostGroup.objects.all()

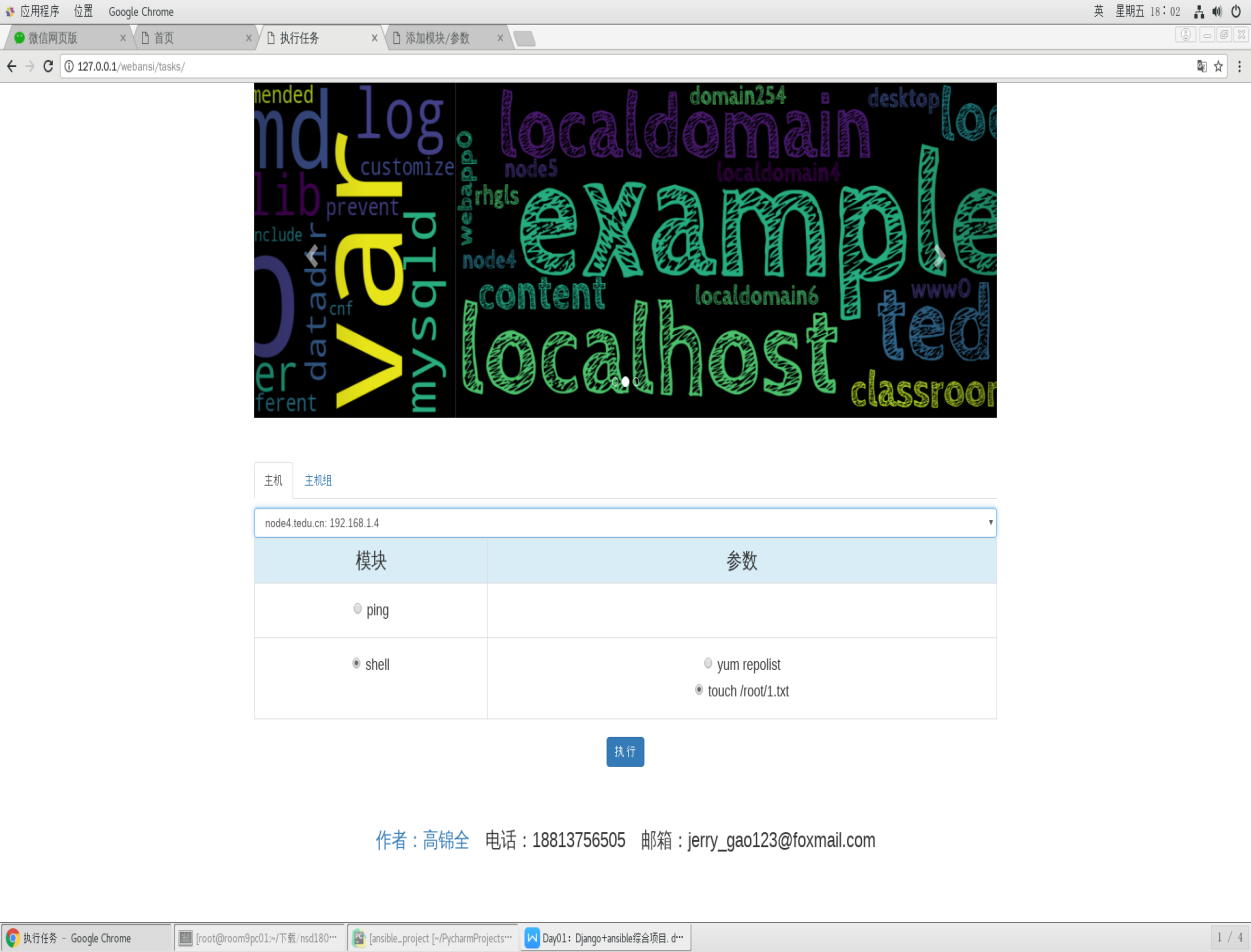
hosts = Host.objects.all()

modules = Module.objects.all()

info = {'groups': groups, 'hosts': hosts, 'modules': modules}

return render(request, 'tasks.html', info)

**3.3 测试"执行任务"功能**



[root@node4 ~]# ls /root/

1.txt