Практика 3: Создание уровня сервиса

Необходимые компоненты

- Практика 2
- Git

Начинаем работать с Git

```
# Текущая директория – project

$ git init

$ git config user.name "Username"
$ git config user.email user@example.com

$ git add *
$ git commit –m "Initial commit"
```

```
[master (root-commit) 07da5f1] Initial commit
 19 files changed, 346 insertions(+)
create mode 100644 .ansible-retry/app_demo.retry
create mode 100644 .ansible-retry/aws.retry
create mode 100755 ansible.cfg
create mode 100755 aws.yml
create mode 100644 command
create mode 100644 environments/dev/group_vars/all
create mode 100644 environments/dev/inventory
create mode 100755 export.sh
create mode 100644 id_rsa_ansible-2016-sep
create mode 100644 id_rsa_ansible-2016-sep.pub
create mode 100644 playbooks/app_demo.yml
create mode 100755 requirements.txt
create mode 100644 roles/nginx/defaults/main.yml
create mode 100644 roles/nginx/handlers/main.yml
create mode 100644 roles/nginx/tasks/main.yml
create mode 100644 roles/nginx/templates/countries.html.j2
create mode 100644 roles/nginx/templates/default.j2
create mode 100644 roles/nginx/templates/index.html.j2
create mode 100644 vault.key
```

requirements.yml

touch requirements.yml

base-service-app

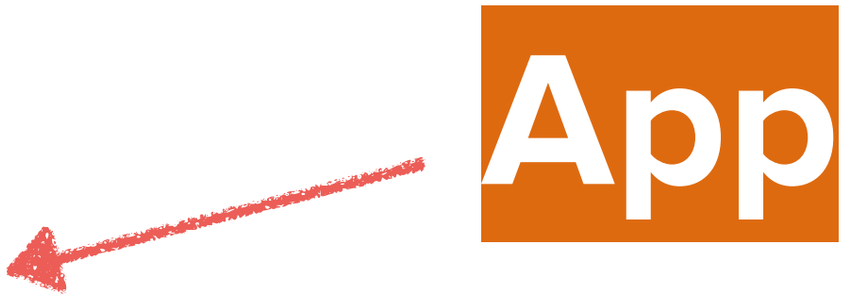
Layer	Responsibility
application	Dev Ops*
service (nginx, pgsql,etc)	Ops
base (OC, backups, monitoring)	Ops

Service

```
- name: install nginx
  apt: name=nginx state=present

- name: enable nginx
  service: name=nginx state=started enabled=yes

- name: configure nginx
  template: src=default.j2 dest=/etc/nginx/sites-enabled/default
  notify:
    - restart nginx
```

```
- name: make folder for project
  file: path=/srv/web state=directory

- name: create page
  template: src=index.html.j2 dest=/srv/web/index.html

- name: create page for countries
  template: src=countries.html.j2 dest=/srv/web/countries.html
```

```
- name: install nginx
  apt: name=nginx state=present

- name: enable nginx
  service: name=nginx state=started enabled=yes

- name: configure nginx
  template: src=default.j2 dest=/etc/nginx/sites-enabled/default
  notify:
    - restart nginx

- name: make folder for project
  file: path=/srv/web state=directory

- name: create page
  template: src=index.html.j2 dest=/srv/web/index.html

- name: create page for countries
  template: src=countries.html.j2 dest=/srv/web/countries.html
```

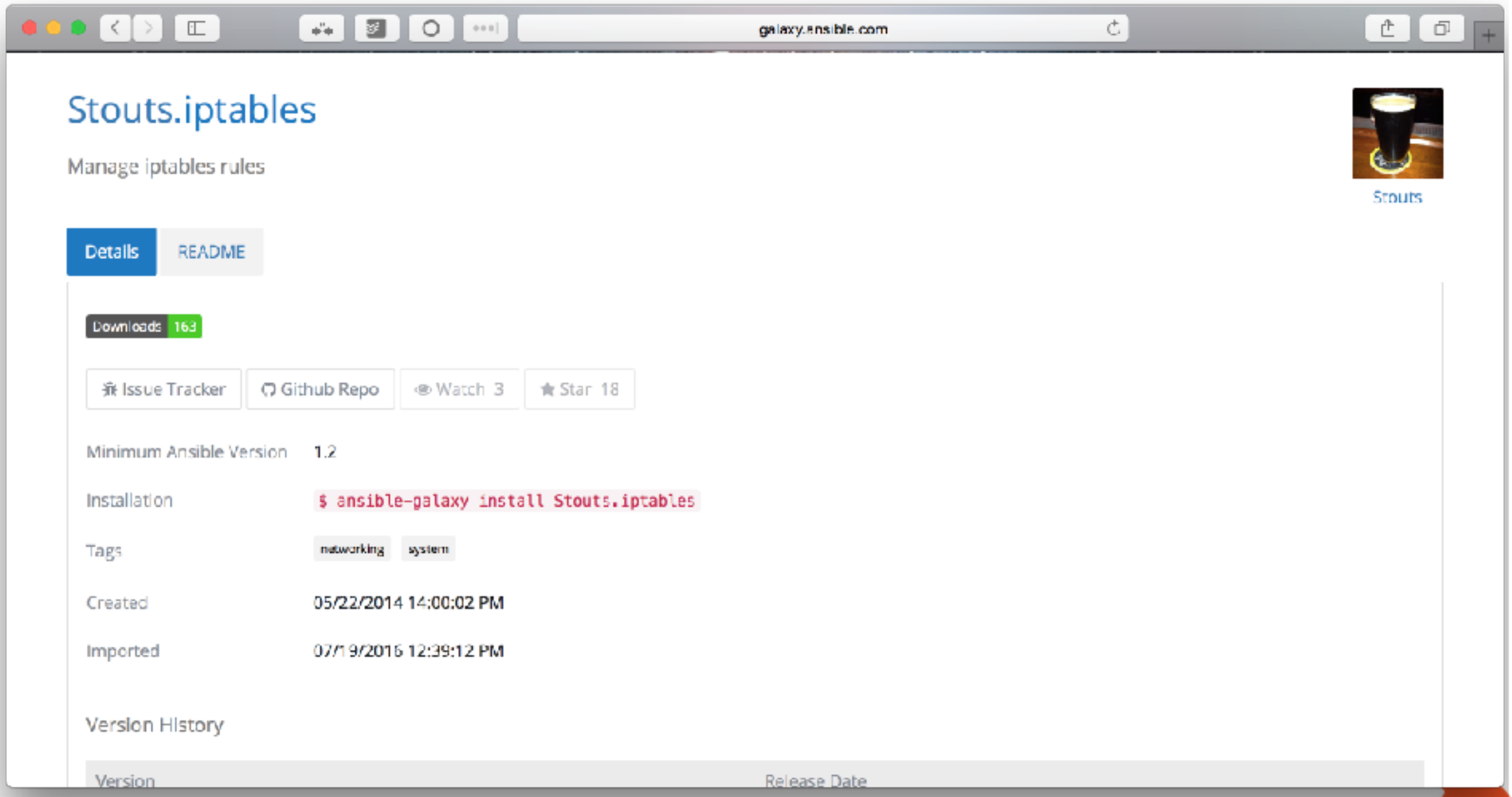


Уровень Base

base
(OC, backups, monitoring)

Ops

iptables из Galaxy



requirements.yml

```
- src: https://github.com/Stouts/Stouts.iptables.git
  version: 1.1.2
```

Устанавливаем

```
$ ansible-galaxy install -r requirements.yml
```

Добавляем в git

```
$ git add requirements.yml
```

```
$ git commit -m "Add iptables role to project"
```

Пишем playbook для базового слоя

playbooks/base.yml

```
- hosts: all
  become: true
  roles:
    - Stouts.iptables
  tasks:
    - apt: update_cache=yes
```

Добавляем переменную

environments/dev/group_vars/web

```
iptables_allowed_tcp_ports: [22,80]
```

Применяем конфигурацию

```
$ ansible-playbook playbooks/base.yml
```

Iptables на машине

```
$ sudo iptables-save

# Completed on Sun Aug 28 19:13:25 2016
# Generated by iptables-save v1.4.21 on Sun Aug 28 19:13:25 2016
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [88:12204]
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -p udp -m udp --sport 123 -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -m limit --limit 15/min -j LOG --log-prefix "Dropped by
iptables: " --log-level 7
-A INPUT -j DROP
-A OUTPUT -p udp -m udp --dport 123 -j ACCEPT
COMMIT
```

Добавляем в git

```
$ git add .
$ git commit -m "add base playbook to get
ability control iptables parameters"
```

Уровень Service

service (nginx, pgsql,etc)

Ops

Nginx роль

https://github.com/jdauphant/ansible-role-nginx

requirements.yml

```
- src: Stouts.iptables
- src: jdauphant.nginx
```

Устанавливаем

```
$ ansible-galaxy install -r requirements.yml
```

Добавляем в git

```
$ git add .
$ git commit -m "Add Nginx role"
```

Пишем playbook для сервисного слоя

playbooks/service.yml

```
- hosts: web
  become: true
  roles:
    - jdauphant.nginx
```

Добавляем переменные

environments/dev/group_vars/web

```
iptables_allowed_tcp_ports: [22,80]
nginx_sites:
  default:
    - listen 80
    - server_name "localhost"
```

Обновляем app_demo

```
- include: base.yml
- include: service.yml
```

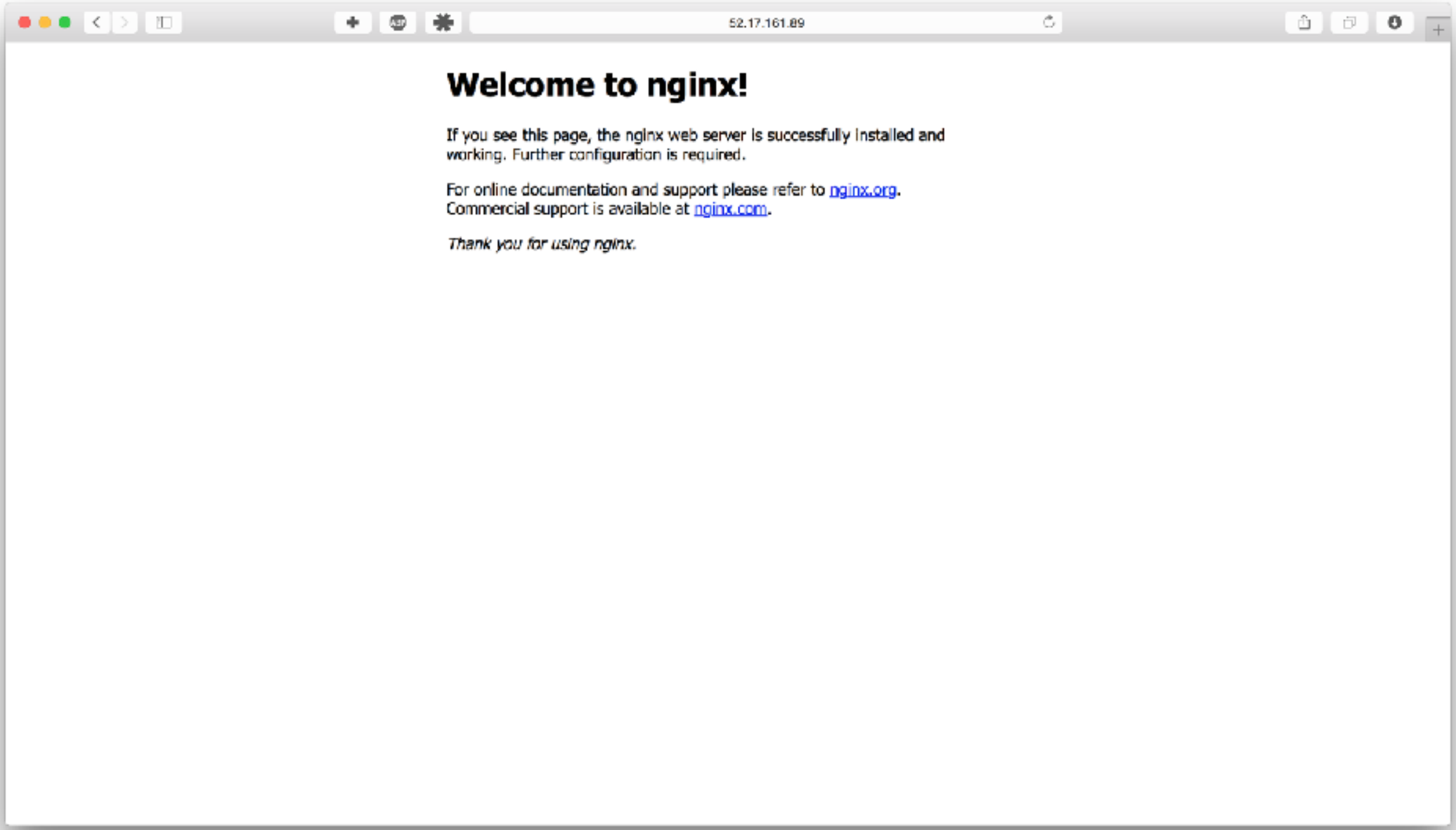
Добавляем в git

```
$ git add .
$ git commit -m "migrate app_demo to base-service-app model "
```

Применяем конфигурацию

```
$ ansible-playbook playbooks/app_demo.yml
```

Проверяем


52.17.161.89
Welcome to nginx!
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.
Thank you for using nginx.

Проверяем

```
$ git log

* 2016-08-28 76ed646 (HEAD -> master) migrate app_demo to base-service-
app model  [Alexander Titov]
* 2016-08-28 29ddcc6 use nginx role from galaxy for site  [Alexander
Titov]
* 2016-08-28 791c162 add nginx role  [Alexander Titov]
* 2016-08-28 113f56c add base playbook to get ability control iptables
parameters  [Alexander Titov]
* 2016-08-28 b6aa4de Add dependency resolving and iptables role to
project  [Alexander Titov]
* 2016-08-28 07da5f1 Initial commit  [Alexander Titov]
```

requirements.yml

```
- src: Stouts.iptables
- src: jdauphant.nginx
- src: ANXS.postgresql
```

Устанавливаем

```
$ ansible-galaxy install -r requirements.yml
```

Настраиваем БД

environments/dev/group_vars/web

```
postgresql_version: 9.5
postgresql_users:
  - name: appuser
    pass: password
    encrypted: no

postgresql_databases:
  - name: application
    owner: appuser
```

Шифруем данные

```
$ ansible-vault encrypt environments/dev/
group_vars/web
```

Обновляем playbook сервисного слоя

playbooks/service.yml

```
- hosts: web
become: true
roles:
  - jdauphant.nginx
  - ANXS.postgresql
```

Применяем конфигурацию

```
$ ansible-playbook playbooks/app_demo.yml
```

Добавляем в git

```
$ git add .
$ git commit -m "add database role and
configuration"
```

Уровень приложения

application

Dev Ops*

requirements.yml

```
- src: Stouts.iptables
- src: jdauphant.nginx
- src: ANXS.postgresql
- src: ragingbal.java8
```

Устанавливаем

```
$ ansible-galaxy install -r requirements.yml
```

Обновляем playbook приложения

playbooks/app_demo.yml

```
- include: base.yml
- include: service.yml

- hosts: web
  become: true
  roles:
    - ragingbal.java8
```

Применяем конфигурацию

```
$ ansible-playbook playbooks/app_demo.yml
```

Добавляем в git

```
$ git add .
```

```
$ git commit -a -m "add java 8 role"
```

requirements.yml

```
- src: Stouts.iptables
- src: jdauphant.nginx
- src: ANXS.postgresql
- src: ragingbal.java8
- src: ragingbal.tomcat
```

Устанавливаем

```
$ ansible-galaxy install -r requirements.yml
```

Редактируем шифрованные данные

```
$ ansible-vault edit environments/dev/group_vars/
web
```

Настраиваем tomcat

environments/dev/group_vars/web

```
http_port: 8080
https_port: 8443
admin_username: tomcat
admin_password: tomcat
```

Настраиваем nginx

environments/dev/group_vars/web

```
nginx_sites:
  default:
  - listen 80
  - server_name "localhost"
  - location / {
      client_max_body_size 32M;
      proxy_pass http://127.0.0.1:8080;
    }
```

Обновляем playbook приложения

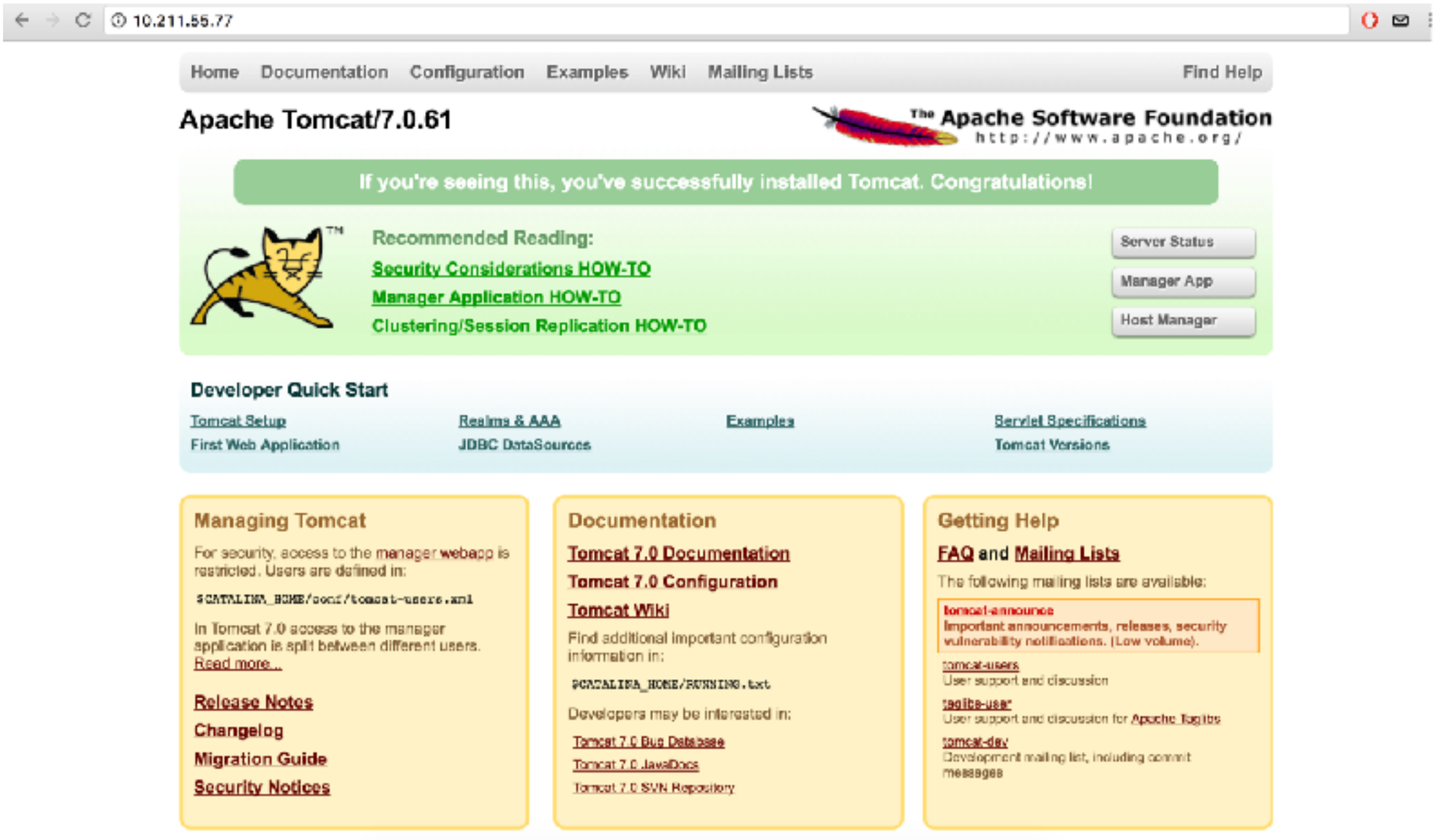
playbooks/app_demo.yml

```
        - hosts: all
          become: true
        roles:
          - ragingbal.java8
          - jdauphant.nginx
          - ragingbal.tomcat

        handlers:
          - name: restart tomcat
            service: name=tomcat state=restarted
```

Применяем конфигурацию

```
$ ansible-playbook playbooks/app_demo.yml
```

Проверяем


10.211.55.77
Home
Documentation
Configuration
Examples
Wiki
Mailing Lists
Find Help
Apache Tomcat/7.0.61
The Apache Software Foundation
http://www.apache.org/
If you're seeing this, you've successfully installed Tomcat. Congratulations!
Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO
Server Status
Manager App
Host Manager
Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions
Managing Tomcat
For security, access to the manager webapp is restricted. Users are defined in:
$CATALINA_HOME/conf/tomcat-users.xml
In Tomcat 7.0 access to the manager application is split between different users.
Read more...
Release Notes
Changelog
Migration Guide
Security Notices
Documentation
Tomcat 7.0 Documentation
Tomcat 7.0 Configuration
Tomcat Wiki
Find additional important configuration information in:
$CATALINA_HOME/RUNNING.txt
Developers may be interested in:
Tomcat 7.0 Bug Database
Tomcat 7.0 JavaDocs
Tomcat 7.0 SVN Repository
Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).
tomcat-users
User support and discussion
taglibs-user
User support and discussion for Apache Taglibs
tomcat-dev
Development mailing list, including commit messages

Добавляем в git

```
$ git add .
$ git commit -a -m "add tomcat role and
configure nginx as proxy"
```

Проверяем зависимости

```
|– environments
|   `– dev
|– files
|   |– petstore-schema.sql
|   |– petstore-data.sql
|   `– jpetstore-1.0.0.war
|– playbooks
|   |– app_demo.yml
|   |– base.yml
|   `– service.yml
|– requirements.yml
```

Обновляем playbook приложения

```
- hosts: web
  become: true
  roles:
    - ragingbal.java8
    - ragingbal.tomcat

tasks:
  - name: 'Copy database schema file'
    copy:
      src: ../files/petstore-schema.sql
      dest: /var/lib/postgresql/petstore-schema.sql
      owner: postgres
      group: postgres
      mode: 0600
```

Обновляем playbook приложения

```
tasks:
  - name: 'Copy database schema file'
    copy:
      src: ../files/petstore-schema.sql
      dest: /var/lib/postgresql/petstore-schema.sql
      owner: postgres
      group: postgres
      mode: 0600

- name: 'Copy database data file'
  copy:
    src: ../files/petstore-data.sql
    dest: /var/lib/postgresql/petstore-data.sql
    owner: postgres
    group: postgres
    mode: 0600
```

Обновляем playbook приложения

```
- name: 'Load data into database'
  shell: psql -f {{item}} application && touch {{item}}.lock
  args:
    chdir: /var/lib/postgresql
    creates: "{{item}}.lock"
  become: yes
  become_user: postgres
  with_items:
    - petstore-schema.sql
    - petstore-data.sql
```

Применяем конфигурацию

```
$ ansible-playbook playbooks/app_demo.yml
```

Добавляем в git

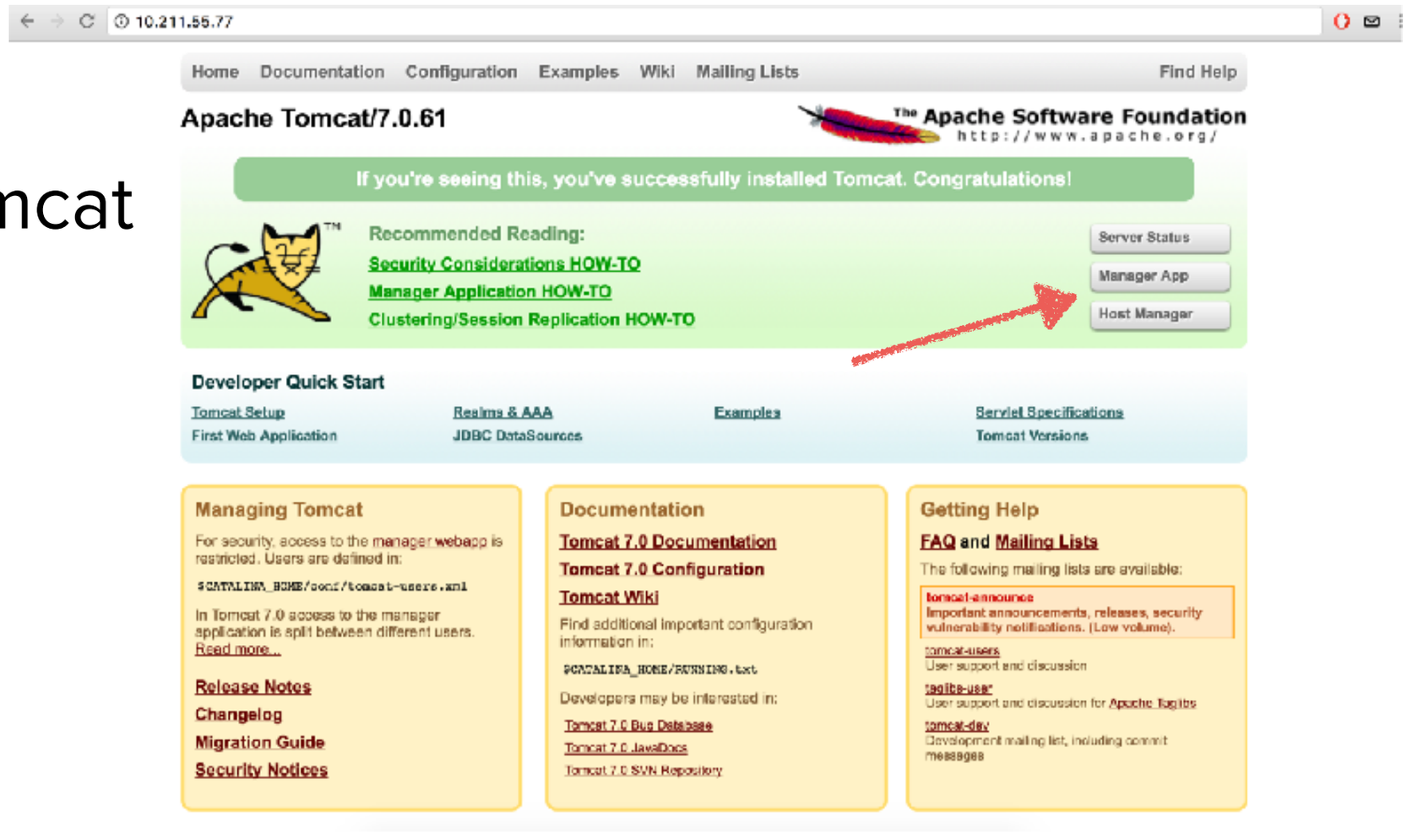
```
$ git add .
$ git commit -a -m "add database schema and data
files loading for application"
```

Заливаем приложение

Переходим на /manager и авторизуемся

login: tomcat

password: tomcat



Заливаем приложение

manager	*specified*	Application	true	0	Expire sessions with idle ≥ 30 minutes
/manager	*None specified*	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

Deploy

WAR file to deploy

Select WAR file to upload Choose File No file chosen

Deploy

Diagnostics

Check to see if a web application has caused a memory leak on stop, reload or undeploy

Find leaks This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.

Заливаем приложение

The Apache Software Foundation
http://www.apache.org/

Tomcat Web Application Manager

Message:	OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/jpetstore-1.0.0	None specified	Spring JPetStore	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Проверяем


JPetStore
Demo
| Sign-in |?
Search
Fish | Dogs | Reptiles | Cats | Birds
<<Manx
Great for reducing mouse populations
EST-15
With tail Manx
Manx
10 000 in stock.
$23,50
Add to Cart
Powered By
Spring
Powered By
iBATIS
(Currently running on the Spring web tier)

Задание со звездочкой

- Научите nginx раздавать статику вместо tomcat