

# Инструкция по использованию пакета развертывания ПО absence-control

## 1. Описание

Программное обеспечение absence-control предназначено для учета часов отсутствия сотрудников в рабочее время. Пакет приложения распространяется в виде ролей для системы управления конфигурацией Ansible, управление осуществляется через SSH-доступ. Структурная схема взаимодействия управляющей и целевых машин приведена на Рис. 1.

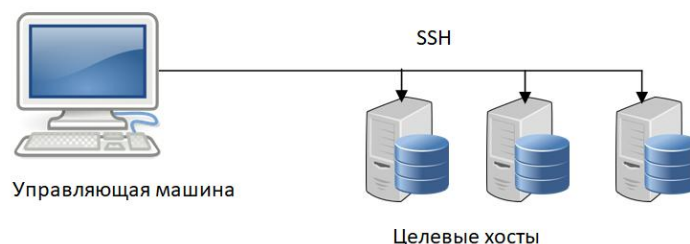


Рис. 1. Структурная схема развертывания конфигурации

Пакет установки размещается на управляющей машине, откуда осуществляется настройка конфигурации и развертывания ПО на целевые машины. При необходимости, управляющая машина может быть и целевой, при этом в инвентори-файле необходимо указать адрес локального целевого хоста **127.0.0.1**.

**Примечание:** Данный пакет предназначен для развертывания на Linux Ubuntu

## 2. Подготовка к установке

Прежде чем приступить к установке, убедитесь, что на управляющей машине установлены Python и Ansible, а также что с управляющей машины имеется ssh-доступ к целевой машине. Для установки необходимого ПО выполните в консоли управляющей машины следующую команду:

**\$sudo apt-get update & sudo apt-get install python ansible**

После окончания установки можно переходить к развертыванию конфигурации целевой машины и сборке приложения

## 3. Развертывание инфраструктурного ПО и сборка приложения

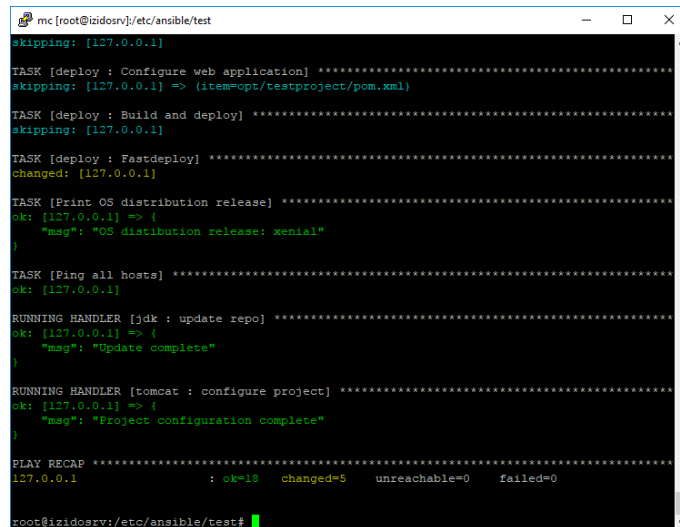
Перед запуском развертывания необходимо внести следующие изменения в конфигурацию ролей:

- Добавить записи целевых хостов в инвентори-файле (inventory/hosts)
- Для роли Tomcat при необходимости изменить значение переменной TomcatURL, скопировав адрес ссылки на архив с приложением с официального сайта. Файл с описанием переменных roles/tomcat/vars/main.yml

Для запуска инсталляции приложения выполните из каталога пакета установки скрипт **test.sh**:

```
$/test.sh
```

Запустится скрипт установки, который развернет окружение, после чего выполнит сборку и установку приложения. При необходимости повысить привилегии, скрипт запросит SUDO – пароль для целевой машины. Сборка из исходников происходит на целевой машине. Вывод отладочной информации о ходе установки и сборки происходит в каталог `./log`. После окончания работы скрипт выведет в консоль количество выполненных задач по установке и закончит работу (Рис.2).



```
mc [root@izidosrv:/etc/ansible/test]
skipping: [127.0.0.1]
TASK [deploy : Configure web application] *****
skipping: [127.0.0.1] => (item=opt/testproject/pom.xml)
TASK [deploy : Build and deploy] *****
skipping: [127.0.0.1]
TASK [deploy : Fastdeploy] *****
changed: [127.0.0.1]
TASK [Print OS distribution release] *****
ok: [127.0.0.1] => {
  "msg": "OS distribution release: xenial"
}
TASK [Ping all hosts] *****
ok: [127.0.0.1]
RUNNING HANDLER [jdk : update repo] *****
ok: [127.0.0.1] => {
  "msg": "Update complete"
}
RUNNING HANDLER [tomcat : configure project] *****
ok: [127.0.0.1] => {
  "msg": "Project configuration complete"
}
PLAY RECAP *****
127.0.0.1 : ok=18  changed=5  unreachable=0  failed=0
root@izidosrv:/etc/ansible/test#
```

Рис. 2. Окно консоли после развертывания приложения

#### 4. Быстрое развертывание приложения

Режим быстрого развертывания используется для развертывания заранее собранного веб-архива из роли `deploy/files`. При быстром развертывании сборка проекта не осуществляется. Для выполнения быстрого развертывания следует установить переменную в плейбуке `'FASTDEPLOY=true'` либо использовать для запуска скрипт **test\_fastdeploy.sh**:

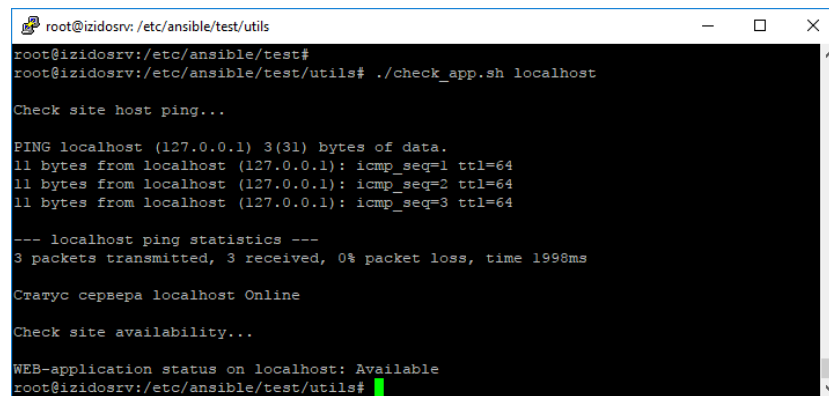
```
$/test_fastdeploy.sh
```

#### 5. Проверка установки

Для проверки установки следует выполнить скрипт проверки `check_app.sh` из каталога `utils`. Скрипт проверяет отклик целевого хоста и доступность развернутого приложения. В качестве параметра скрипта следует указать адрес целевого хоста (127.0.0.1 для разворачивания на локальной машине):

```
$/check_app.sh <IP>
```

При успешном развертывании приложения в окно консоли выводится следующая информация (Рис. 3):



```
root@izidosrv: /etc/ansible/test/utls
root@izidosrv:/etc/ansible/test#
root@izidosrv:/etc/ansible/test/utls# ./check_app.sh localhost

Check site host ping...

PING localhost (127.0.0.1) 3(31) bytes of data.
11 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64
11 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64
11 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64

--- localhost ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms

Статус сервера localhost Online

Check site availability...

WEB-application status on localhost: Available
root@izidosrv:/etc/ansible/test/utls#
```

Рис. 3. Окно вывода скрипта проверки правильности установки приложения

При выводе сообщения о доступности веб-приложения работа по установке и настройке ПО выполнена корректно. При отсутствии отклика **ping** следует проверить доступность целевого хоста, при отображении статуса **Not available** для приложения следует проверить журнал работы скрипта на наличие ошибок развертывания.