**Aplana performance testing framework (Ptf). Платформа для проведения нагрузочного тестирования**

***Руководство по установке***

2019 г

# Введение

* 1. **Назначение документа**

Данное руководство предназначено для администраторов, осуществляющих настройку платформы для проведения нагрузочного тестирования. В руководстве описываются процедуры подготовки ведущего сервера к работе.

* 1. **Термины, определения и сокращения**

|  |  |
| --- | --- |
| **Название** | **Определение** |
| **Система** | Программный комплекс Aplana PTF |
| **ПО** | Программное обеспечение |
| **НТ** | Нагрузочное тестирование |
| **БД** | База данных |
| **СУБД** | Система управления базами данных |
| **ОС** | Операционная система |

* 1. **Системные требования**

Серверная часть:

* Процессор – не менее 4vCPU, 3 GHz+
* Оперативная память – не менее 16 GB
* Дисковая память – объем не менее 50 GB для Системы, не менее 150 GB для БД.
* Пропускная способность внешнего канала связи – не менее 100 Мбит/сек 2)

Клиентская часть гарантированно работает в web-браузерах:

* Google Chrome
* Internet Explorer 10+.
	1. **Требования к программному обеспечению**

На сервере приложений и БД должно быть предустановлено следующее ПО:

1. ОС Linux ALT 8.2 Server
2. Выделены права sudo
3. Установленный пакет ssh
4. systemd
	1. **Требования к квалификации персонала**

Пользователь, выполняющий операции по установке, обновлению и переносу Системы должен обладать знаниями ОС Linux на уровне не ниже администратора.

* 1. **Порядок действий развертывания Aplana PTF**

**Для master-сервера:**

1. Создание учетной записи ansible.
2. Настройка файла-конфигурации ssh.
3. Обмен публичными ключами с удаленными машинами для nopass ssh.
4. Установка ansible.
5. Установка и запуск компонент PTF с помощью playbook на удаленном сервере.

**Для slave-сервера:**

1. Создание учетной записи ansible.
2. Обмен публичными ключами с мастером для nopass ssh.
3. Установка дополнительных библиотек.

# Создание учетной записи.

На всех машинах необходимо выполнить следующие действия:

Дать права на чтение и запуск исполняемого файла программы sudo всем пользователям.

sudo chmod -R a+rx /usr/bin/sudo

Необходимо создать пользователя ansible с помощью команды:

sudo useradd ansible

Установить пароль для пользователя ansible

sudo passwd ansible

Добавить пользователя в sudoers.

sudo echo "ansible ALL=(ALL) NOPASSWD: ALL" | sudo tee -a /etc/sudoers

Фрагмент итогового содержания файла sudoers:



[iv@alt-master ~]$ sudo tail /etc/sudoers

## Uncomment to allow any user to run sudo if they know the password

## of the user they are running the command as (root by default).

# Defaults targetpw # Ask for the password of the target user

# ALL ALL=(ALL) ALL # WARNING: only use this together with 'Defaults targetpw'

## Read drop-in files from /etc/sudoers.d

## (the '#' here does not indicate a comment)

#includedir /etc/sudoers.d

ansible ALL=(ALL) NOPASSWD: ALL

Далее все действия на машинах необходимо производить под пользователем ***ansible***.

su ansible

**В случае если папка /home/ansible не существует**, следует также выполнить шаги 5-8:

Создание домашней директории

sudo mkdir /home/ansible

Создание папки .ssh

sudo mkdir /home/ansible/.ssh

Изменение владельца на ansible.

sudo chown -R ansible:ansible /home/ansible

Изменение прав на папку .ssh

sudo chmod 700 /home/ansible/.ssh

# Настройка файла-конфигурации ssh.

На всех машинах в файле /etc/ssh/sshd\_config должна быть включена опция:

***PasswordAuthentication yes***

# Обмен публичными ключами.

**На мастере:**

1. Необходимо создать ключ-пару с помощью команды:

ssh-keygen



[ansible@alt-master ~]$ ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/home/ansible/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/ansible/.ssh/id\_rsa.

Your public key has been saved in /home/ansible/.ssh/id\_rsa.pub.

The key fingerprint is:

SHA256:kbbKKz3TyzaEXAngbQTdM4dJ++Cu4A+tvHZIcQo0GEk ansible@alt-master.localdomain

The key's randomart image is:

+---[RSA 2048]----+

|oE o+.o.o |

|o o. oo \*o. |

| . .. o.\*= |

| . ...oo= |

| . = oS . |

| oooo. |

| .oo+o. |

| o++=++ |

| .==++oo |

+----[SHA256]-----+

1. Далее добавить созданный публичный ключ на удаленную машину

ssh-copy-id <IP-слейва>



[ansible@alt-master ~]$ ssh-copy-id 192.168.1.233

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansible/.ssh/id\_rsa.pub"

The authenticity of host '192.168.1.233 (192.168.1.233)' can't be established.

ED25519 key fingerprint is SHA256:kK3bJEoCPErmstAzzyKyb4qp+RoxEfFRIrFUZhTxg2s.

Are you sure you want to continue connecting (yes/no)? yes

/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys

ansible@192.168.1.233's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh '192.168.1.233'"

and check to make sure that only the key(s) you wanted were added.

1. Для проверки соединения необходимо выполнить следующие команды:

ssh ansible@<IP слейва>



[ansible@alt-master ~]$ ssh ansible@192.168.1.233

[ansible@alt-t1 ~]$

**На слейв-машине:**

1. Необходимо создать ключ-пару с помощью команды:

ssh-keygen



[ansible@alt-t1 ~]$ ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/home/ansible/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/ansible/.ssh/id\_rsa.

Your public key has been saved in /home/ansible/.ssh/id\_rsa.pub.

The key fingerprint is:

SHA256:SuRjrtttdaYxBNwpDKmLvmoeQgmjIK9DAOxbBNvJWjc ansible@alt-t1.localdomain

The key's randomart image is:

+---[RSA 2048]----+

|o.. .= . . |

|..+.. . = o |

|B..= E. o |

|\*++.oo. . |

|.=+. .= S. |

|oo. .+ o + o |

|+o o . \* |

|.oo o .. . |

|+o..o.... |

+----[SHA256]-----+

1. Далее добавить созданный публичный ключ на машину мастера

ssh-copy-id <IP-мастера>



[ansible@alt-t1 ~]$ ssh-copy-id 192.168.1.236

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansible/.ssh/id\_rsa.pub"

The authenticity of host '192.168.1.236 (192.168.1.236)' can't be established.

ED25519 key fingerprint is SHA256:kK3bJEoCPErmstAzzyKyb4qp+RoxEfFRIrFUZhTxg2s.

Are you sure you want to continue connecting (yes/no)? yes

/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys

ansible@192.168.1.236's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh '192.168.1.236'"

and check to make sure that only the key(s) you wanted were added.

1. Для проверки соединения необходимо выполнить следующие команды:

ssh ansible@<IP мастера>



[ansible@alt-t1 ~]$ ssh ansible@192.168.1.236

[ansible@alt-master ~]$

# Установка ansible .

**На мастере**:

Необходимо выполнить следующие команды:

sudo apt-get update

sudo apt-get install python

sudo apt-get install python-module-pip

curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py

sudo python get-pip.py

sudo pip install ansible==2.8.1

После установки, необходимо проверить корректно ли прошла установка, с помощью команды:

ansible --version



[ansible@alt-master ~]$ ansible --version

ansible 2.8.1

 config file = None

 configured module search path = [u'/home/ansible/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']

 ansible python module location = /usr/lib/python2.7/site-packages/ansible

 executable location = /usr/bin/ansible

 python version = 2.7.11 (default, May 30 2018, 12:09:46) [GCC 5.3.1 20151207 (ALT Linux 5.3.1-alt3)]

# Установка дополнительных библиотек.

**На слейв-машинах**:

Необходимо выполнить следующие команды:

sudo apt-get update

sudo apt-get install python

sudo apt-get install python-module-pip

# Настройка файла hosts.

**На мастере:**

Необходимо создать папку /etc/ansible:

sudo mkdir /etc/ansible

Необходимо создать файл /etc/ansible/hosts, следующего содержания (и перечислить **ip-адреса машин слейвов**):



[all:vars]

ansible\_user=ansible

[slave]

192.168.1.233

# Установка дополнительных модулей ansible.

**На мастере:**

Необходимо выполнить следующую команду для установки модуля rossmcdonald.telegraf:

ansible-galaxy install rossmcdonald.telegraf



[ansible@alt-master ~]$ ansible-galaxy install rossmcdonald.telegraf

- downloading role 'telegraf', owned by rossmcdonald

- downloading role from https://github.com/rossmcdonald/telegraf/archive/v1.1.0.tar.gz

- extracting rossmcdonald.telegraf to /home/ansible/.ansible/roles/rossmcdonald.telegraf

- rossmcdonald.telegraf (v1.1.0) was installed successfully

# Подготовка playbook для запуска инсталяции

**На мастере**:

Необходимо разархивировать приложенный архив aplana-ptf.tar.gz с помощью команды:

sudo tar xf aplana-ptf.tar.gz -C /etc/ansible/

Необходимо установить владельца директории /etc/ansible с помощью команды:

sudo chown -R ansible:ansible /etc/ansible

# Запуск инсталЛятора.

**На мастере**:

Необходимо запустить инсталляцию с помощью команды:

ansible-playbook /etc/ansible/playbooks/main.yml -v

Конечный результат выполнения playbook:



PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.1.233 : ok=89 changed=66 unreachable=0 failed=0 skipped=0 rescued=0 ignored=1

# Проверка доступности компонент

Последней задачей в playbook идет проверка доступности всех разворачиваемых компонент, корректный результат выглядит как сервисы со state=running:



TASK [debug] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [192.168.1.233] => (item=redis-binary.service) => {

 "ansible\_facts.services[\"redis-binary.service\"].state": "running",

 "ansible\_loop\_var": "item",

 "item": "redis-binary.service"

}

ok: [192.168.1.233] => (item=telegraf-binary.service) => {

 "ansible\_facts.services[\"telegraf-binary.service\"].state": "running",

 "ansible\_loop\_var": "item",

 "item": "telegraf-binary.service"

}

ok: [192.168.1.233] => (item=jenkins-binary.service) => {

 "ansible\_facts.services[\"jenkins-binary.service\"].state": "running",

 "ansible\_loop\_var": "item",

 "item": "jenkins-binary.service"

}

ok: [192.168.1.233] => (item=grafana-binary.service) => {

 "ansible\_facts.services[\"grafana-binary.service\"].state": "running",

 "ansible\_loop\_var": "item",

 "item": "grafana-binary.service"

}

ok: [192.168.1.233] => (item=influxdb-binary.service) => {

 "ansible\_facts.services[\"influxdb-binary.service\"].state": "running",

 "ansible\_loop\_var": "item",

 "item": "influxdb-binary.service"

}

ok: [192.168.1.233] => (item=elastic-binary.service) => {

 "ansible\_facts.services[\"elastic-binary.service\"].state": "running",

 "ansible\_loop\_var": "item",

 "item": "elastic-binary.service"

}