Участник:Евгений

Содержание

Jitsi Meet и Ansible

Jitsi Meet - это приложение JavaScript, входящее в инфраструктуру open-source проектов Jitsi, позволяющее быстро настроить на своем сервере масштабируемую и безопасную платформу для видеоконференций.
В кластер проектов Jitsi также входят такие компоненты, как [1] (https://jitsi.github.io/handbook)[2] (https://2021.desosa.nl/projects/jitsi/posts/essay_2/) :

1. Jitsi Videobridge (JVB) - совместимый с WebRTC серверный компонент XMPP, обеспечивающий запуск до тысячи видеопотоков с одного сервера, а также измеряющий доступную пропускную способность для клиентов с целью управления потоками.
2. Jibri - набор решение для записи или потоковой передачи конференций Jitsi Meet, взаимодействующий с библиотекой FFmpeg, а также экземпляром Chrome
3. Prosody - внешний сервер XMPP, позволяющий каждому из связанных компонентов Jitsi обмениваться данными по протоколу XMPP.
4. Jicofo - XMPP-компонент, управляющий подключением клиентов к видеоконференции. Получая сигнал от Prosody, именно он входит в комнату и организует конференцию.
5. Jigasi - серверное приложение, позволяющие присоединиться к конференции SIP-клиентам. Данный компонент регистрируется как клиент и может быть использован для исходящих звонков.

В дальнейшей работе к серверной части не будет производиться подключение SIP-шлюза (jigasi) и рекордера (jibri).
Ansible - это open-source решение, автоматизирующее работу с конфигурационными файлами серверов, доставку и установку ПО на них.
Средствами Ansible в данной статьей будет произведена установка сервера веб-приложения Jitsi Meet на машину Альт Сервер.

Стенд, предварительные установки и настройки

Были скачаны и установлены образы Альт Рабочая Станция 10.1 и Альт Сервер 10.1 соответственно на виртуальные машины altwks1 и altsrv1. В качестве системы виртуализации использовалось ПО VirtualBox 6.1.38.

Настройки параметров сети

На ВМ altwks1 было настроено два сетевых адаптера: NAT и адаптер соединяющий машину с внутренней виртуальной сетью net1. На altsrv1 настроен один сетевой адаптер net1 во внутреннюю виртуальную сеть.

Настройки на altwks1

Далее все команды выполняются от имени суперпользователя.
Создаем второй конфигурационный файл для внутреннего интерфейса(enp0s8) наподобие конфига NAT-интерфейса(enp0s3) из /etc/net/ifaces:

```
# cd /etc/net/ifaces
# cp -r enp0s3 enp0s8
```

Для enp0s8 изменяем конфигурацию на статическую настройку сетевых параметров вместо DHCP, указываем ipv4-адрес интерфейса и перезапускаем его:

```
# sed -i '/BOOTPROTO/s/dhcp/static/' enp0s8/options
# echo "192.168.100.101/24" > enp0s8/ipv4address
# ifdown enp0s8; ifup enp0s8
```

Далее создаем правила NAT трансляции IP-адресов из подсети 192.168.100.0/24 в IP-адрес внешнего интерфейса, записываем их в файл enp0s3/ifup-post для автозагрузки правил в таблицу после включения enp0s3, делаем этот файл исполняемым и перезапускаем интерфейс:

```
# echo "iptables -t nat -A POSTROUTING -s 192.168.100.0/24 -j MASQUERADE" > enp0s3/ifup-post
# chmod +x enp0s3/ifup-post
# ifdown enp0s3; ifup enp0s3
```

Включим возможность перенаправлять IP-пакеты с одного сетевого интерфейса на другой и перезапустим службу NetworkManager:

```
# sed -i '/net\.ipv4\.ip_forward/s/0/1/' /etc/net/sysctl.conf
# systemctl restart NetworkManager
```

Настройки на altsrv1

Далее все команды выполняются от имени суперпользователя.
Для enp0s8 изменяем конфигурацию на статическую настройку сетевых параметров вместо DHCP, указываем ipv4-адрес интерфейса и перезапускаем его:

```
# sed -i '/BOOTPROTO/s/dhcp/static/' enp0s8/options
# echo "192.168.100.121/24" > enp0s8/ipv4address
# ifdown enp0s8; ifup enp0s8
```

Дальше указываем маршрут по умолчанию и адрес DNS-сервера и перезапускаем интерфейс:

```
# echo "default via 192.168.100.101" > enp0s8/ipv4route
# echo "nameserver 8.8.8.8" > enp0s8/resolv.conf
# ifdown enp0s8; ifup enp0s8
```

Настройки ansible

В роли управляющего узла выступает узел altwks1, а в роли управляемого altsrv1. Для возможности взаимодействия ansible с управляемыми узлами укажем настройки разрешения имен на двух машинах в файле `/etc/hosts`, дописав в файл:

```
# 192.168.100.101 altwks1.courses.alt altwks1
# 192.168.100.121 altsrv1.courses.alt altsrv1
```

Чтобы плейбук мог исполнять команды от имени суперпользователя на управляемых узлах необходимо настроить ssh подключение к altsrv1:

```
sysadmin@altwks1 ~ $ ssh-keygen
sysadmin@altwks1 ~ $ ssh-copy-id sysadmin@altsrv1
sysadmin@altwks1 ~ $ ssh sysadmin@altsrv1
[sysadmin@altsrv1 ~]$ su -
[sysadmin@altsrv1 ~]$ cp /home/sysadmin/.ssh/authorized_keys .ssh/authorized_keys
```

Проверить возможность подключения от root-пользователя с помощью команды `ssh root@altsrv1`.
Далее, на altwks1 необходимо скачать сам пакет ansible с помощью команды `apt-get update && apt-get install ansible`, а на узле altsrv1 пакеты с интерпретатором python `apt-get update && apt-get install python3 python3-module-yaml python3-module-jinja2 python3-module-jsonlib`.
На altwks1 отредактируем файл инвентаризации /etc/ansible/hosts, где объявим группу "servers" с нашим управляемым узлом, а также от имени какого пользователя будут исполняться команды и в каком интерпретаторе (у нас это python3):

```
[all:vars]
ansible_user=root
ansible_python_interpreter=/usr/bin/python3

[servers]
altsrv1.courses.alt
```

Создадим каталог ролей и перейдем в него, а затем средствами утилиты ansible-galaxy создадим роль jitsi-meet командами:

```
# mkdir roles && cd roles
# ansible-galaxy init jitsi-meet
```

Если ваши политики по умолчанию блокируют весь трафик на altsrv1 не попавший под действие ни одного из правил в цепочке, то необходимо открыть порты 10000/udp и 4443/tcp для jitsi-videobridge, 443/tcp для веб-сервера, 5280/tcp для prosody, 22/tcp для возможности SSH-подключения, 53/udp для разрешения DNS-имен следующими командами, 80/tcp для выпуска и обновления TLS-сертификата следующей конфигурацией в /etc/ansible/roles/jitsi-meet/tasks/main.yml:

```
- hosts: servers

  tasks:
  - name: Allow TCP port 22 (SSH)
    ansible.builtin.iptables:
      chain: INPUT
      protocol: tcp
      destination_port: "22"
      jump: ACCEPT
  - name: Allow TCP port 4443 (jitsi-videobridge)
    ansible.builtin.iptables:
      chain: INPUT
      protocol: tcp
      destination_port: "4443"
      jump: ACCEPT
  - name: Allow TCP port 5280 (prosody)
    ansible.builtin.iptables:
      chain: INPUT
      protocol: tcp
      destination_port: "5280"
      jump: ACCEPT
  - name: Allow TCP port 80 (TLS)
    ansible.builtin.iptables:
      chain: INPUT
      protocol: tcp
      destination_port: "80"
      jump: ACCEPT
  - name: Allow UDP port 10000 (jitsi-videobridge)
    ansible.builtin.iptables:
      chain: INPUT
      protocol: udp
      destination_port: "10000"
      jump: ACCEPT
  - name: Allow UDP port 53 (DNS)
    ansible.builtin.iptables:
      chain: INPUT
      protocol: udp
      destination_port: "53"
      jump: ACCEPT
  - name: Allow TCP port 443 (HTTPS)
    ansible.builtin.iptables:
      chain: INPUT
      protocol: tcp
      destination_port: "443"
      jump: ACCEPT
```

Данной конфигурацией мы автоматически создадим требуемые цепочки. Далее на altsrv1 нам нужно скачать пакеты prosody jitsi-meet-prosody jitsi-meet-web jitsi-meet-web-config jicofo jitsi-videobridge. Отредактируем тот же файл mail.yml:

```
  - name: Install a list pf packages for jitsi-meet
    apt_rpm:
      pkg:
      - prosody
      - jitsi-meet-prosody
      - jitsi-meet-web
      - jitsi-meet-web-config
      - jicofo
      - jitsi-videobridge
      state: present
      update_cache: true
```

Изменим имя хоста системы, а также разрешение имени для localhost и его доменное имя на altsrv1 и altsrv11.courses.alt соответственно:

```
  - name: Set a hostname altsrv1
    ansible.builtin.hostname:
      name: altsrv1
  - name: Change hostname in /etc/hosts
    replace:
      path: /etc/hosts
```

```
      regexp: '(\s+)localhost\.localdomain localhost(\s+.*)?$'
      replace: '\1altsrv11.courses.alt altsrv1\2'
      backup: yes
```

Автоматизация настройки prosody

Создадим дополнительную директорию для конфигурации prosody, а также добавим строку с подключением будущей директории, где будет находится конфиг для нашего XMPP-сервера:

```
  - name: Create dir /etc/prosody/conf.d/ if it does not exist
    file:
      path: /etc/prosody/conf.d/
      state: directory
  - name: Add line in /etc/prosody/prosody.cfg.lua
    lineinfile:
      path: /etc/prosody/prosody.cfg.lua
      line: Include "conf.d/*.cfg.lua"
      create: yes
```

Далее создадим в директории files нашей роли конфиг, который будет отправлен на сервер в каталог /etc/prosody/conf.d/altsrv1.courses.alt.cfg.lua с таким содержимым:

```
plugin_paths = { "/usr/share/jitsi-meet/prosody-plugins/" }

-- domain mapper options, must at least have domain base set to use the mapper
muc_mapper_domain_base = "altsrv1.courses.alt";
cross_domain_bosh = false;
consider_bosh_secure = true;

----------- Virtual hosts -----------
VirtualHost "altsrv1.courses.alt"
  authentication = "anonymous"
  ssl = {
    key = "/var/lib/prosody/altsrv1.courses.alt.key";
    certificate = "/var/lib/prosody/altsrv1.courses.alt.crt";
  }
  speakerstats_component = "speakerstats.altsrv1.courses.alt"
  conference_duration_component = "conferenceduration.altsrv1.courses.alt"
  -- we need bosh
   modules_enabled = {
    "bosh";
    "pubsub";
    "ping"; -- Enable mod_ping
    "speakerstats";
    "turncredentials";
    "conference_duration";
  }
  c2s_require_encryption = false


Component "conference.altsrv1.courses.alt" "muc"
 storage = "memory"
 modules_enabled = {
  "muc_meeting_id";
  "muc_domain_mapper";
   -- "token_verification";
 }
admins = { "focus@auth.altsrv1.courses.alt" }
 muc_room_locking = false
 muc_room_default_public_jids = true



VirtualHost "auth.altsrv1.courses.alt"
 ssl = {
  key = "/var/lib/prosody/auth.altsrv1.courses.alt.key";
  certificate = "/var/lib/prosody/auth.altsrv1.courses.alt.crt";
 }
```

```
 authentication = "internal_plain"


-- internal muc component, meant to enable pools of jibri and jigasi clients
Component "internal.auth.altsrv1.courses.alt" "muc"
 storage = "memory"
 modules_enabled = {
  "ping";
 }
 admins = { "focus@auth.altsrv1.courses.alt", "jvb@auth.altsrv1.courses.alt" }
 muc_room_locking = false
 muc_room_default_public_jids = true


Component "focus.altsrv1.courses.alt"
 component_secret = "secret1" -- достаточно длинный пароль, он же JICOFO_SECRET

Component "speakerstats.altsrv1.courses.alt" "speakerstats_component"
 muc_component = "conference.altsrv1.courses.alt"

Component "conferenceduration.altsrv1.courses.alt" "conference_duration_component"
 muc_component = "conference.altsrv1.courses.alt"
```

Пропишем в плейбуке (jitsi-meet/tasks/main.yml) команду для отправки файла на удаленный узел:

```
  - name: Copy file in altsrv1
    copy:
      src: /etc/ansible/roles/jitsi-meet/files/altsrv1.courses.alt.cfg.lua
      dest: /etc/prosody/conf.d/altsrv1.courses.alt.cfg.lua
```

Необходимо сгенерировать SSL сертификаты для auth.altsrv1.courses.alt и altsrv1.courses.alt. Оба сертификата будут находиться в директории /var/lib/prosody, откуда их необходимо будет добавить в доверенные сертификаты :

```
  - name: Generate SSL keypair for Prosody service.
    shell: >
      yes '' | (prosodyctl cert generate altsrv1.courses.alt &&  prosodyctl cert generate auth.altsrv1.courses.alt)
  - name: Create sym link
    file:
      src: /var/lib/prosody/altsrv1.courses.alt.crt
      dest: /etc/pki/ca-trust/source/anchors/altsrv1.courses.alt.crt
      state: link
  - name: Create symlink auth
    file:
      src: /var/lib/prosody/auth.altsrv1.courses.alt.crt
      dest: /etc/pki/ca-trust/source/anchors/auth.altsrv1.courses.alt.crt
      state: link
  - name: "Update-ca-cert"
    shell:  update-ca-trust
```

Создадим пользователя focus конференции (пользователь, отвечающий за подключение участников к конференции) и перезапустим XMPP-сервер:

```
  - name: Register jicofo agent with Prosody service.
    shell:  prosodyctl register focus auth.altsrv1.courses.alt secret2 && prosodyctl restart
```

Автоматизация настройки jicofo

Создадим файл config в директории /etc/ansible/roles/jitsi-meet/files/ с таким содержимым:

```
# Jitsi Conference Focus settings
# sets the host name of the XMPP server
JICOFO_HOST=localhost
```

```
# sets the XMPP domain (default: none)
JICOFO_HOSTNAME=altsrv1.courses.alt

# sets the secret used to authenticate as an XMPP component
JICOFO_SECRET=secret1

# overrides the prefix for the XMPP component domain. Default: "focus"
#JICOFO_FOCUS_SUBDOMAIN=focus

# sets the port to use for the XMPP component connection
JICOFO_PORT=5347

# sets the XMPP domain name to use for XMPP user logins
JICOFO_AUTH_DOMAIN=auth.altsrv1.courses.alt

# sets the username to use for XMPP user logins
JICOFO_AUTH_USER=focus

# sets the password to use for XMPP user logins
JICOFO_AUTH_PASSWORD=secret2

# extra options to pass to the jicofo daemon
JICOFO_OPTS="${JICOFO_FOCUS_SUBDOMAIN:+ --subdomain=$JICOFO_FOCUS_SUBDOMAIN}"
# adds java system props that are passed to jicofo (default are for home and logging config file)
JAVA_SYS_PROPS="-Dnet.java.sip.communicator.SC_HOME_DIR_LOCATION=/etc/jitsi
-Dnet.java.sip.communicator.SC_HOME_DIR_NAME=jicofo
-Dnet.java.sip.communicator.SC_LOG_DIR_LOCATION=/var/log/jitsi
-Djava.util.logging.config.file=/etc/jitsi/jicofo/logging.properties"
```

Отправим данный файл на управляемый узел в директорию /etc/jitsi/jicofo/ с помощью плейбука:

```
  - name: Copy file in altsrv1
    copy:
      src: /etc/ansible/roles/jitsi-meet/files/config
      dest:  /etc/jitsi/jicofo/config
```

Также рядом с файлом config необходимо создать файл sip-communicator.properties с содержимым:

```
org.jitsi.jicofo.health.ENABLE_HEALTH_CHECKS=true
org.jitsi.jicofo.BRIDGE_MUC=JvbBrewery@internal.auth.altsrv1.courses.alt
```

Отправим данный файл на управляемый узел в ту же директорию и запустим jicofo:

```
  - name: Copy .prop in altsrv1
    copy:
      src: /etc/ansible/roles/jitsi-meet/files/sip-communicator.properties
      dest:  /etc/jitsi/jicofo/sip-communicator.properties
  - name: Start jicofo
    systemd:
      name: jicofo
      state: started
```

У нас получилось подключить jicofo к XMPP-серверу как внешний XMPP-компонент, и как пользовательский аккаунт с JID focus@auth.altsrv1.courses.alt.

Автоматизация настройки jitsi-videobridge

Зарегистрируем на XMPP-сервере аккаунт jvb@auth.altsrv1.courses.alt:

```
   - name: Register jvb@auth.altsrv1.courses.alt
     shell:  prosodyctl jvb auth.altsrv1.courses.alt secret3
```

Создадим три файла (sip-communicator.properties, config, application.conf) в /etc/ansible/roles/jitsi-meet/files/jitsi-videobridge/, чтобы в дальнейшем переслать их в каталог /etc/jitsi/videobridge/. Содержимое файла config:

```
# Jitsi Videobridge settings
# extra options to pass to the JVB daemon
JVB_OPTS="--apis=,"
# adds java system props that are passed to jvb (default are for home and
logging config file)
JAVA_SYS_PROPS="-Dnet.java.sip.communicator.SC_HOME_DIR_LOCATION=/etc/jitsi
-Dnet.java.sip.communicator.SC_HOME_DIR_NAME=videobridge
-Dnet.java.sip.communicator.SC_LOG_DIR_LOCATION=/var/log/jitsi
-Djava.util.logging.config.file=/etc/jitsi/videobridge/logging.properties
-Dconfig.file=/etc/jitsi/videobridge/application.conf"
```

Содержимое файла application.conf:

```
videobridge {
  stats {
      enabled = true
      transports = [
          { type = "muc" }
      ]
  }
  apis {
      xmpp-client {
          configs {
            shard {
                hostname = "localhost"
                domain = "auth.altsrv1.courses.alt"
                username = "jvb"
                password = "secret3"
                muc_jids = "JvbBrewery@internal.auth.altsrv1.courses.alt"
                # The muc_nickname must be unique across all instances
                muc_nickname = "jvb-mid-123"
            }
          }
      }
  }
}
```

Содержимое файла sip-communicator.properties:

```
rg.ice4j.ice.harvest.DISABLE_AWS_HARVESTER=true
org.ice4j.ice.harvest.STUN_MAPPING_HARVESTER_ADDRESSES=meet-jit-siturnrelay.jitsi.net:443
org.jitsi.videobridge.ENABLE_STATISTICS=true
org.jitsi.videobridge.STATISTICS_TRANSPORT=muc
org.jitsi.videobridge.xmpp.user.shard.HOSTNAME=localhost
org.jitsi.videobridge.xmpp.user.shard.DOMAIN=auth.altsrv1.courses.alt
org.jitsi.videobridge.xmpp.user.shard.USERNAME=jvb
org.jitsi.videobridge.xmpp.user.shard.PASSWORD=secret3
org.jitsi.videobridge.xmpp.user.shard.MUC_JIDS=JvbBrewery@internal.auth.altsrv1.courses.alt
org.jitsi.videobridge.xmpp.user.shard.MUC_NICKNAME=6d8b40cb-fe32-49f5-a5f6-13d2c3f95bba
```

Далее в плейбуке мы копируем эти файлы в директорию /etc/jitsi/videobridge/ и запускаем сам videobridge:

```
  - name: Copy files in videobridge
    copy:
      src: /etc/ansible/roles/jitsi-meet/files/jitsi-videobridge/sip-communicator.properties
      dest:  /etc/jitsi/videobridge/sip-communicator.properties
  - name: Copy files in videobridge
    copy:
      src: /etc/ansible/roles/jitsi-meet/files/jitsi-videobridge/config
      dest: /etc/jitsi/videobridge/config
  - name: Copy files in videobridge
    copy:
```

```
      src: /etc/ansible/roles/jitsi-meet/files/jitsi-videobridge/sip-communicator.properties
      dest:  /etc/jitsi/videobridge/sip-communicator.properties
```

Источник — https://altwiki.inlinux.ru/index.php?title=Участник:Евгений&oldid=1044

Эта страница в последний раз была отредактирована 14 июля 2023 в 03:40.