WCS в Yandex.Cloud

Начиная со сборки 5.2.759, WCS может быть развернут в Yandex.Cloud как отдельно стоящий медиа сервер или часть CDN с низкой задержкой.

Для начала развертывания необходимы:

- активный аккаунт в Yandex.Cloud, облако и виртуальная приватная сеть в этом аккаунте
- лицензия WCS для активации на сервере/серверах
- дополнительно, доменные имена для привязки к экземплярам серверов

Развертывание сервера

Развертывание и запуск экземпляра виртуальной машины

1. В консоли Yandex,Cloud перейдите в раздел "Compute Cloud - Virtual machines" и нажмите "Create VM", чтобы начать создание экземпляра сервера.

2. Введите имя сервера, описание и регион расположения датацентра

3. В разделе "Computing resources" выберите тип процессора, количество процессоров, объем памяти. Укажите параметр "Guaranteed vCPU performance" равным "100%"

4. В разделе "Image/boot disk selection" выберите операционную систему Centos, версию 7 (допускаются также другие операционные системы, перечисленные здесь)

5. В разделе "Disks" выберите тип и размер диска

6. В разделе "Network settings" выберите доступную подсеть, при необходимости укажите вручную IP адреса

7. В разделе "Access" укажите имя пользователя и публичный SSH ключ для доступа к серверу

и нажмите "Create VM"

8. Созданный сервер появится в списке

9. Щелкните по строке сервера в списке, скопируйте внешний адрес из раздела Network для подключения к серверу

10. Подключитесь к серверу по ssh

Настройка межсетевого экрана

В настоящее время Yandex.Cloud не поддерживает группы безопасности (эта возможность находится в статусе Preview), поэтому необходимо настроить межсетевой экран на самом сервере:

iptables_setup.sh Expand source

```
#!/bin/bash
#
export IPT="iptables"

# External interface
export WAN=eth0

# Clean iptables
$IPT -F
$IPT -F -t nat
$IPT -F -t mangle
$IPT -X
$IPT -t nat -X
$IPT -t mangle -X

# Set default policies
$IPT -P INPUT ACCEPT
$IPT -P OUTPUT ACCEPT
$IPT -P FORWARD ACCEPT

# Allow local traffic
$IPT -A INPUT -i lo -s 127.0.0.0/8 -d 127.0.0.0/8 -j ACCEPT
$IPT -A OUTPUT -o lo -s 127.0.0.0/8 -d 127.0.0.0/8 -j ACCEPT

# Allow outgoing connections
$IPT -A OUTPUT -o $WAN -j ACCEPT

# Allow already established connections
$IPT -A INPUT -p all -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A OUTPUT -p all -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A FORWARD -p all -m state --state ESTABLISHED,RELATED -j ACCEPT

# Enable packet fragmentation
```

```
#$IPT -I FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu

# Drop invalid packets
$IPT -A INPUT -m state --state INVALID -j DROP
$IPT -A FORWARD -m state --state INVALID -j DROP
$IPT -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
$IPT -A OUTPUT -p tcp ! --syn -m state --state NEW -j DROP

# Allow pings
$IPT -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type time-exceeded -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type echo-request -j ACCEPT

# Allow SSH
$IPT -A INPUT -p tcp --dport 22 -j ACCEPT
# Allow DNS
#$IPT -A INPUT -i $WAN -p udp --dport 53 -j ACCEPT
# Allow NTP
#$IPT -A INPUT -i $WAN -p udp --dport 123 -j ACCEPT

# Allow WCS ports
$IPT -A INPUT -p tcp --dport 80 -j ACCEPT
$IPT -A INPUT -p tcp --dport 443 -j ACCEPT
$IPT -A INPUT -p tcp --dport 8888 -j ACCEPT
$IPT -A INPUT -p tcp --dport 8443 -j ACCEPT
$IPT -A INPUT -p tcp --dport 1935 -j ACCEPT
$IPT -A INPUT -p udp --dport 1935 -j ACCEPT
$IPT -A INPUT -p tcp --dport 554 -j ACCEPT
$IPT -A INPUT -p tcp --dport 3478 -j ACCEPT
$IPT -A INPUT -p tcp --dport 8080 -j ACCEPT
$IPT -A INPUT -p tcp --dport 8081 -j ACCEPT
$IPT -A INPUT -p tcp --dport 8084 -j ACCEPT
$IPT -A INPUT -p tcp --dport 8082 -j ACCEPT
$IPT -A INPUT -p tcp --dport 8085 -j ACCEPT
$IPT -A INPUT -p tcp --dport 8445 -j ACCEPT
$IPT -A INPUT -p tcp --dport 8444 -j ACCEPT
$IPT -A INPUT -p tcp --dport 10000:50000 -j ACCEPT
$IPT -A INPUT -p udp --dport 10000:50000 -j ACCEPT
$IPT -A INPUT -p tcp --dport 50999 -j ACCEPT

$IPT -A INPUT -j DROP
$IPT -A FORWARD -j DROP

# Save the rules to file
/sbin/iptables-save  > /etc/sysconfig/iptables
```

Установка и настройка WCS

1. Установите JDK. Для работы в условиях больших нагрузок рекомендуется JDK 12 или 14

```
#!/bin/bash
sudo rm -rf jdk*
```

```
curl -s
https://download.java.net/java/GA/jdk12.0.2/e482c34c86bd4bf8b56c0b35558996b9/10/
12.0.2_linux-x64_bin.tar.gz | tar -zx
[ ! -d jdk-12.0.2/bin ] && exit 1
sudo mkdir -p /usr/java
[ -d /usr/java/jdk-12.0.2 ] && sudo rm -rf /usr/java/jdk-12.0.2
sudo mv -f jdk-12.0.2 /usr/java
[ ! -d /usr/java/jdk-12.0.2/bin ] && exit 1
sudo rm -f /usr/java/default
sudo ln -sf /usr/java/jdk-12.0.2 /usr/java/default
sudo update-alternatives --install "/usr/bin/java" "java" "/usr/java/jdk-
12.0.2/bin/java" 1
sudo update-alternatives --install "/usr/bin/jstack" "jstack" "/usr/java/jdk-
12.0.2/bin/jstack" 1
sudo update-alternatives --install "/usr/bin/jcmd" "jcmd" "/usr/java/jdk-
12.0.2/bin/jcmd" 1
sudo update-alternatives --install "/usr/bin/jmap" "jmap" "/usr/java/jdk-
12.0.2/bin/jmap" 1
sudo update-alternatives --set "java" "/usr/java/jdk-12.0.2/bin/java"
sudo update-alternatives --set "jstack" "/usr/java/jdk-12.0.2/bin/jstack"
sudo update-alternatives --set "jcmd" "/usr/java/jdk-12.0.2/bin/jcmd"
sudo update-alternatives --set "jmap" "/usr/java/jdk-12.0.2/bin/jmap"
```

2. Установите дополнительные инструменты и библиотеки

```
sudo yum install -y tcpdump mc iperf3 fontconfig
```

3. Отключите SELinux

```
sudo setenforce 0
```

4. Установите WCS

```
curl -OL
https://flashphoner.com/downloads/builds/WCS/5.2/FlashphonerWebCallServer-
5.2.xxx.tar.gz
tar -xzf FlashphonerWebCallServer-5.2.xxx.tar.gz
cd FlashphonerWebCallServer-5.2.xxx
sudo ./install.sh
```

Здесь xxx - номер сборки WCS

5. Активируйте Вашу лицензию

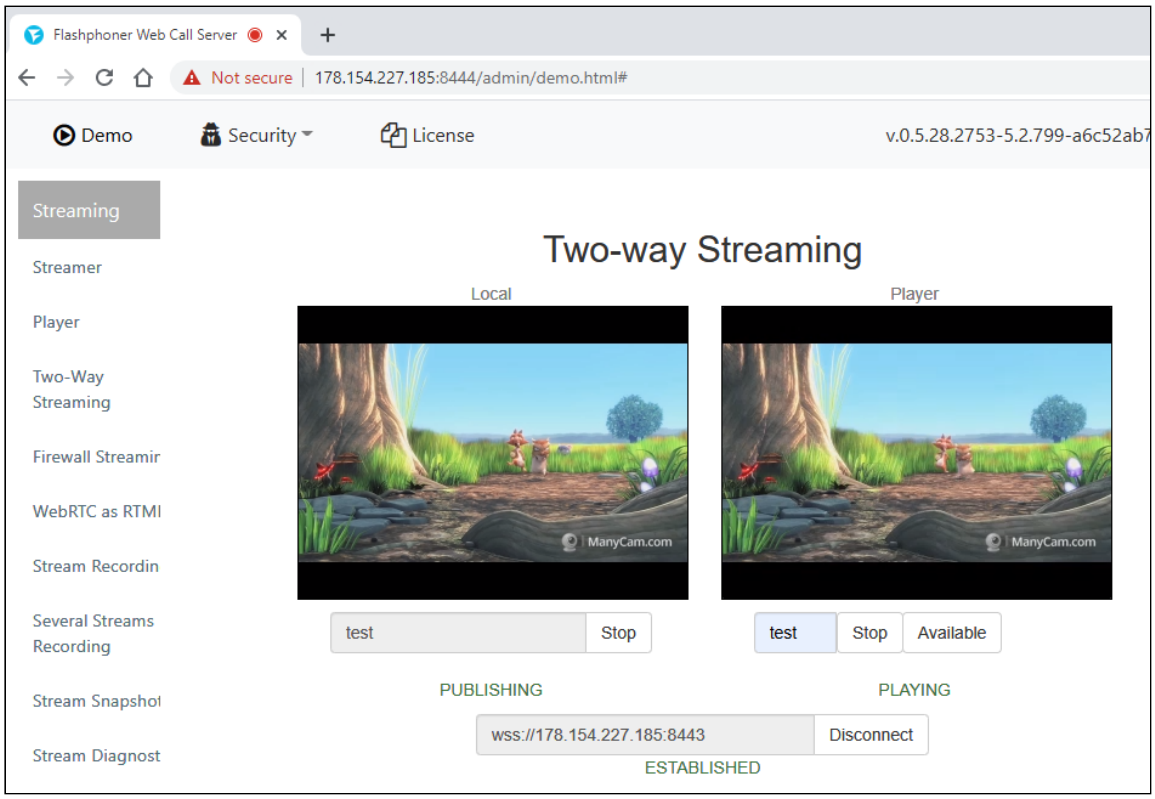
```
cd /usr/local/FlashphonerWebCallServer/bin
sudo ./activation.sh
```

Запуск и тест WCS

1. Запустите WCS

```
sudo systemctl start webcallserver
```

2. Войдите в веб-интерфейс сервера, откройте пример Two Way Streaming, опубликуйте и проиграйте поток test



Параметры пользователя admin по умолчанию

Yandex.Cloud поддерживает два варианта получения данных о запущенном инстансе: Google Cloud API endpoints и AWS EC2 API endpoints. Поэтому, начиная со сборки 5.2.921, WCS по умолчанию определяет облачную среду как Amazon.

В свою очередь, одно из главных требований Amazon - это уникальный пароль администратора для каждого инстанса, поэтому в качестве пароля в облачной среде Amazon используется уникальный instanceId, доступный через API или в EC2 консоли.

Таким образом, при запуске WCS в Yandex.Cloud, начиная со сборки 5.2.921, для пользователя admin по умолчанию устанавливается пароль, равный instanceId. Однако, этот параметр может не отображаться в консоли Yandex.Clouid. Для того, чтобы узнать instanceId, подключитесь к серверу по SSH и используйте следующую команду

```
curl http://169.254.169.254/latest/meta-data/instance-id
```

Attachments:

- yandex_create_vm.png (image/png)
- yandex_create_vm_choose_name.png (image/png)
- yandex_create_vm_choose_name.png (image/png)
- yandex_create_vm_choose_cpu.png (image/png)
- yandex_create_vm_choose_os.png (image/png)
- yandex_create_vm_choose_hdd.png (image/png)
- yandex_create_vm_choose_network.png (image/png)
- yandex_create_vm_choose_user_and_create.png (image/png)
- yandex_vm_list.png (image/png)
- yandex_vm_ip_address.png (image/png)
- yandex_vm_ssh.png (image/png)
- yandex_vm_test.png (image/png)