

WCS в Yandex.Cloud

- [Развертывание сервера](#)
 - [Развертывание и запуск экземпляра виртуальной машины](#)
 - [Настройка межсетевое экрана](#)
 - [Установка и настройка WCS](#)
 - [Запуск и тест WCS](#)

Начиная со сборки [5.2.759](#), WCS может быть развернут в Yandex.Cloud как отдельный стоящий медиа сервер или часть CDN с низкой задержкой.

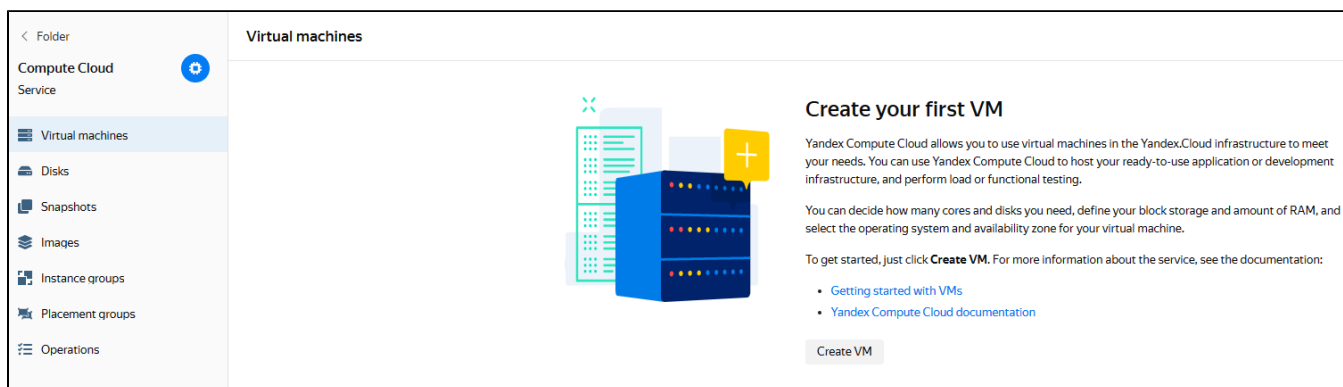
Для начала развертывания необходимы:

- 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%"

Computing resources

Platform ?

vCPU 2 80

Guaranteed vCPU performance ? 5% 20% 50% 100%
 For any task, including high-load services.

RAM 2 GB 32 GB

Additional Preemptible ?

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

Image/boot disk selection

Operating systems Container Solution Cloud Marketplace Custom

Ubuntu 20.04 ?	Windows Server 2019 Datacenter ?
Debian 10 ?	CentOS 7 ?
CoreOS 2303.4.0 ?	openSUSE 42.3 ?

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

Disks

Disk name	Type	Size	Max. IOPS ?	Max. bandwidth ?
CentOS 7 Boot	HDD	20 GB	—	—
		10 GB		4096 GB

Add disk

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

Network settings

Subnet ? default / default-ru-central1-a

Public IP Auto List No address

Advanced DDoS protection ?

Internal address Auto Manual

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

Access

Service account ? Create account

Login ? support

SSH key ?

```
DpQfXmfQfaFBVyyXihffdHoFGfdXx84BUV
Kz35hYonOwsDCvYmVhNTZt4oOYU7t7OxR
A5UjcWunPwzXFtikJDgIL0B25QuqObFP9NH
c+ggJmNA91dxUC9q1QY/GCJTIPdbYO9QO
WGmrlXDDrZxroWDqxfXnMX5CEoVelsgS56
WUnYeXqqTGG9mPnwohcsU41 support
```

Advanced Grant access to serial console ?

Create VM

и нажмите "Create VM"

8. Созданный сервер появится в списке


Virtual machines

Filter by name

All statuses

All availability zones

Table settings 13/14

Name	Status	OS	Platform	vCPU	vCPU performance	RAM	Preemptible	Disk size	Availability zone	Internal IPv4	Public IPv4
test-wcs	Running		Intel Cascade Lake	2	100%	2 GB	no	20 GB	ru-central1-a	10.130.0.20	178.154.227.185

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

Network

Network interface

...

Private IPv4 10.130.0.20

Public IPv4 178.154.227.185 

Subnet [default-ru-central1-a](#)

10. Подключитесь к серверу по ssh

```

$ ssh -i /g/.ssh/id_rsa_yandex support@178.154.227.185
The authenticity of host '178.154.227.185 (178.154.227.185)' can't be established.
ECDSA key fingerprint is SHA256:69SQ1JWPNe3+F7fHHx1K70gmN/hIohHce9NNsrWbVA0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '178.154.227.185' (ECDSA) to the list of known hosts.
[support@test-wcs ~]$ uname
Linux
[support@test-wcs ~]$ uname -a
Linux test-wcs.ru-central1.internal 3.10.0-1127.el7.x86_64 #1 SMP Tue Mar 31 23:36:51 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
[support@test-wcs ~]$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                 2
On-line CPU(s) list:   0,1
Thread(s) per core:    2
Core(s) per socket:    1
Socket(s):              1
NUMA node(s):          1
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  85
Model name:             Intel Xeon Processor (Cascadelake)
Stepping:               6
CPU MHz:                2095.068
BogoMIPS:               4190.13
Hypervisor vendor:     KVM
Virtualization type:   full
L1d cache:              32K
L1i cache:              32K
L2 cache:                4096K
L3 cache:                16384K
NUMA node0 CPU(s):     0,1
Flags:                  fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht sys
x16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch in
dx smap clflushopt clwb avx512cd avx512bw avx512vl xsaveopt xsavec xgetbv1 arat pku ospke avx512_vnni md_clear spec_ctrl intel_
[support@test-wcs ~]$ free
              total        used        free      shared  buff/cache   available
Mem:           1881860      99072      1662108         556       120680       1647320
Swap:              0              0              0
[support@test-wcs ~]$

```

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

```

#!/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/GPL/openjdk-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

Streaming

Streamer

Player

Two-Way Streaming

Firewall Streamir

WebRTC as RTMI

Stream Recordin

Several Streams Recording

Stream Snapshot

Stream Diagnost

Two-way Streaming

Local



Player



test Stop

test Stop Available

PUBLISHING

PLAYING

wss://178.154.227.185:8443 Disconnect

ESTABLISHED