

**Установка Wi-Fi-контроллера в режиме
резервирования Active/Standby
QWC-VC**





Оглавление

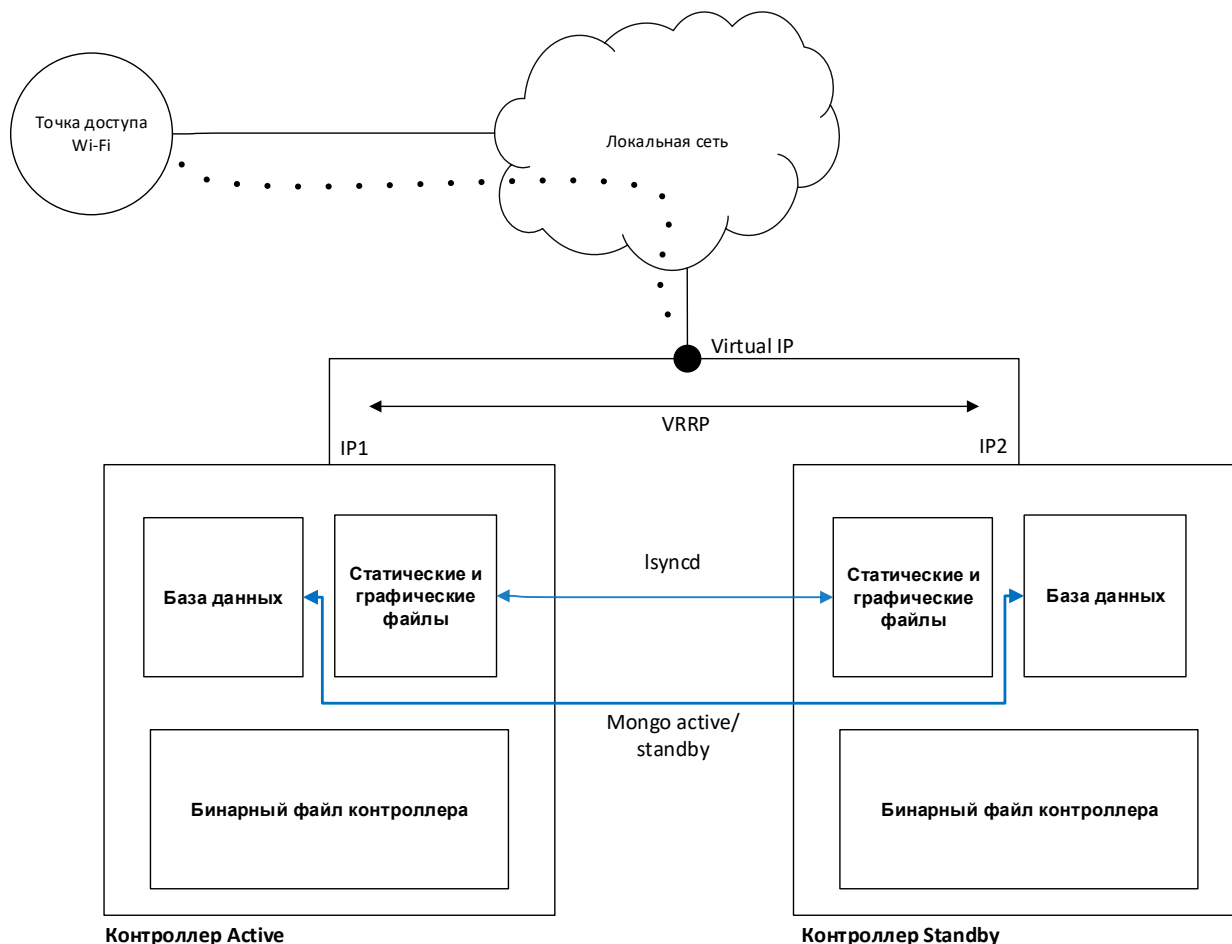
1. ВВЕДЕНИЕ	3
2. СИСТЕМНЫЕ ТРЕБОВАНИЯ	4
2.1. Требования к аппаратной платформе	4
2.2. Требования к операционной системе	4
3. УСТАНОВКА WI-FI-КОНТРОЛЛЕРА	5
4. ОБЩАЯ ИНФОРМАЦИЯ	11
4.1. Замечания и предложения	11
4.2. Гарантия и сервис	11
4.3. Техническая поддержка	11
4.4. Электронная версия документа	11



1. ВВЕДЕНИЕ

Данное руководство содержит всю необходимую информацию для установки и обновления Wi-Fi-контроллера QWC-VC в режиме Active/Standby.

Типовая схема Active/Standby представлена ниже.



Резервирование по схеме 1+1 основано на использовании протокола VRRP и плавающего сетевого адреса (VIP). В качестве программного обеспечения используется *keepalived*.

Keepalived осуществляет следующие функции:

- Проверку состояния серверов.
- Автоматическое переключение ресурсов в случае отказа основного сервера с выполнением скриптов пользователя.

В случае резервирования платформы управления, мониторинга и авторизации Keepalived запускает или останавливает сервисы, необходимые для работы.



2. СИСТЕМНЫЕ ТРЕБОВАНИЯ

2.1. Требования к аппаратной платформе

Для подбора необходимого сервера или виртуальной машины для установки Wi-Fi-контроллера в зависимости от предполагаемого числа точек доступа, которое будет обслуживаться данным контроллером, воспользуйтесь таблицей.

Технические требования к серверу	До 1 000 точек доступа	До 2 500 точек доступа	До 5 000 точек доступа
Ядра процессора	4+ (2,5 ГГц)	8+ (2,5 ГГц)	16+ (2,5 ГГц)
Оперативная память, ГБ	8+	16+	32+
Жесткий диск, ГБ	100+	128+	500+
Сетевой интерфейс	Gigabit Ethernet	Gigabit Ethernet	Gigabit Ethernet
Операционная система	Ubuntu Server 22.04	Ubuntu Server 22.04	Ubuntu Server 22.04

2.2. Требования к операционной системе

Для установки виртуального контроллера QWC-VC операционная система должна удовлетворять следующим требованиям, приведенным в таблице.

Версия ядра ОС Linux	Linux Kernel \geq 4.4
Версия Docker	\geq 17.05
Версия docker-compose	\geq 1.20
Версия Веб-браузера клиента	Google Chrome



3. УСТАНОВКА WI-FI-КОНТРОЛЛЕРА

1. Загрузите бинарный файл контроллера, содержащий в имени –НА- с сайта https://www.qtech.ru/catalog/wireless/lan_controllers/qwc_wm/#documentation
2. Установите Ubuntu Server 22.04 на два физических сервера или виртуальные машины. Hostname должны быть разными. Выделите Virtual IP в той же подсети, что и контроллеры.

Контроллер хранит данные следующим образом:

```
/etc/wimark – данные конфигурации;  
/usr/share/wimark – база данных.
```

В соответствии из этого сделайте разбиение на разделы.

1. Установите docker.

```
sudo curl -sSl https://get.docker.com | sh
```
2. Установите docker-compose.

```
sudo curl -sL https://github.com/docker/compose/releases/download/v2.11.2/docker-compose-linux-x86_64 \  
-o /usr/local/bin/docker-compose  
sudo chmod +x /usr/local/bin/docker-compose
```
3. Установите keepalived.

```
sudo add-apt-repository ppa:hakamur/keepalived  
sudo apt-get update  
sudo apt-get install keepalived
```
4. Для корректной работы keepalived необходимо в файл /etc/sysctl.conf добавить следующие опции, позволяющие системе обрабатывать пакеты с виртуального IP адреса.

```
$ sudo su -  
# echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf  
# echo "net.ipv4.ip_nonlocal_bind = 1" >> /etc/sysctl.conf; sysctl -p  
# iptables -P FORWARD ACCEPT  
# logout
```
5. Для корректной работы сервиса lsyncd (репликация изображений карт и другой статистики) необходимо, чтобы можно было попасть на соседний хост контроллера по ssh по id_rsa ключу пользователя, т.е. без пароля.

Генерируем ключи на каждом хосте:

```
ssh-keygen
```



Копируем ключ на удаленный сервер (изменить user и remote_host):

```
cat ~/.ssh/id_rsa.pub | ssh user@remote_host "mkdir -p ~/.ssh && cat >> ~/.ssh/authorized_keys"
```

6. Загрузить бинарный файл контроллера на контроллер и запустить установщик.
`sudo bash *HA*`

7. Контроллер установился, но не запустился. Предоставленный бинарный файл загружает docker-образы, размещает необходимые статические файлы в /usr/share/wimark, а также создает, размещает конфигурации master и backup вариантов работы в /etc/wimark.

8. На каждом контроллере необходимо отредактировать файлы `.env`

```
sudo su -  
nano /etc/wimark/master/.env  
nano /etc/wimark/backup/.env  
logout
```

Пример для Контроллера 1.

```
wlcuser@wlc-cluster-01:/$ cat /etc/wimark/master/.env  
HOSTNAME=wlc-cluster-01          #hostname сервера  
HOST_UUID=49054ae593f145a390e6b274cdc83e33 #содержимое /etc/machine-id  
SECOND_IP=10.130.0.196          #ip-адрес второго контроллера  
VIP=10.130.0.210                #виртуальный IP-адрес  
USERNAME=wlcuser                #имя пользователя из п.7
```

```
wlcuser@wlc-cluster-01:/$ cat /etc/wimark/backup/.env  
HOSTNAME=wlc-cluster-01  
HOST_UUID=49054ae593f145a390e6b274cdc83e33  
VIP=10.130.0.210
```

Пример для Контроллера 2.

```
wlcuser@wlc-cluster-02:/$ cat /etc/wimark/master/.env  
HOSTNAME=wlc-cluster-02  
HOST_UUID=ef7be027bef34760962f69250c7aff7  
SECOND_IP= 10.130.0.150  
VIP= 10.130.0.210  
USERNAME=wlcuser
```

```
wlcuser@wlc-cluster-02:/$ cat /etc/wimark/backup/.env  
HOSTNAME=wlc-cluster-02
```



```
HOST_UUID=ef7be027bef34760962f69250c7aff7  
VIP=10.130.0.210
```

9. Создаем конфигурационные файлы /etc/wimark/master-backup.sh на каждом контроллере и делаем их исполняемыми:

```
sudo su -  
nano /etc/wimark/master-backup.sh  
chmod ug+x /etc/wimark/master-backup.sh  
logout
```

Вставляем в файл содержимое:

```
#!/bin/bash  
  
TYPE=$1  
NAME=$2  
STATE=$3  
  
LOG_STR="$(date) $STATE"  
  
case $STATE in  
  "MASTER")  
    echo "MASTER" > ./pa.lock  
    rm -f /usr/share/wimark/volumes/mongo/*.lock  
    cd /etc/wimark/backup; docker-compose down  
    cd /etc/wimark/master; docker-compose up -d  
    LOG_STR="$LOG_STR: start master"  
    ;;  
  "BACKUP")ls  
    echo "BACKUP" > ./pa.lock  
    sleep 10  
    [ $(cat ./pa.lock) != "BACKUP" ] && exit 0  
    rm -f /usr/share/wimark/volumes/mongo/*.lock  
    cd /etc/wimark/master; docker-compose down  
    cd /etc/wimark/backup; docker-compose up -d  
    LOG_STR="$LOG_STR: start backup"  
    ;;  
  "FAULT")
```



```
        LOG_STR="$LOG_STR: fault"
        ;;
    *)
        LOG_STR="$LOG_STR: unknown"
        ;;
esac

echo $LOG_STR >> /var/log/wimark-keepalived.log
```

10. Создаем файлы конфигурации /etc/keepalived/keepalived.conf на хостах:

```
sudo nano /etc/keepalived/keepalived.conf
```

и копируем содержимое (необходимо изменить параметры **interface** и **virtual_ipaddress**).
Для контроллера, который будет active:

```
global_defs {
    script_user root
}

vrrp_instance WIM2 {
    # iface for listeting interface: eth0
    interface ens160

    # state of current server: MASTER or BACKUP
    state MASTER

    # id for unique VRRP segment
    virtual_router_id 67

    # preempt - for change MASTER always to first node (if up)
    # nopreempt - stay in position after reconnecting
    #nopreempt

    # some other needed parameters
    garp_master_delay 1
    priority 200
```




```
advert_int 1

# authentication method and password to use
authentication {
    auth_type PASS
    auth_pass wimark123
}

# virtual address to promote
virtual_ipaddress {
    10.130.0.210/24 dev ens160
}

# script to run when STATE is changing -- main logic, DO NOT change
notify /etc/wimark/master-backup.sh
}
```

Для контроллера, который будет standby:

```
global_defs {
    script_user root
}

vrrp_instance WIM1 {
    # iface for listeting interface: eth0
    interface ens160

    # state of current server: MASTER or BACKUP
    state MASTER

    # id for unique VRRP segment
    virtual_router_id 67

    # preempt - for change MASTER always to first node (if up)
    # nopreempt - stay in position after reconnecting
    #nopreempt
}
```



```
# some other needed parameters
garp_master_delay 1
priority 100
advert_int 1

# authentication method and password to use
authentication {
    auth_type PASS
    auth_pass wimark123
}

# virtual address to promote
virtual_ipaddress {
    10.130.0.210/24 dev ens160
}

# script to run when STATE is changing -- main logic, DO NOT change
notify /etc/wimark/master-backup.sh
}
```

11. Запустить и проверить работу keepalived

```
sudo systemctl enable keepalived
sudo systemctl start keepalived
sudo systemctl status keepalived
```

```
ping $VIP
```

12. Проверить работу платформы (все в состоянии running):

```
sudo docker ps -a
```

13. WEB UI доступен по адресу:

```
https://$VIP:443
```

```
логин: wimark
```

```
пароль: wimark
```



4. ОБЩАЯ ИНФОРМАЦИЯ

4.1. Замечания и предложения

Мы всегда стремимся улучшить нашу документацию и помочь вам работать лучше, поэтому мы хотим услышать вас. Мы всегда рады обратной связи, в особенности:

- ошибки в содержании, непонятные или противоречащие места в тексте;
- идеи по улучшению документации, чтобы находить информацию быстрее;
- неработающие ссылки и замечания к навигации по документу.

Если вы хотите написать нам по поводу данного документа, то используйте, пожалуйста, форму обратной связи на qtech.ru.

4.2. Гарантия и сервис

Процедура и необходимые действия по вопросам гарантии описаны на сайте QTECH в разделе «Поддержка» -> «[Гарантийное обслуживание](#)».

Ознакомиться с информацией по вопросам тестирования оборудования можно на сайте QTECH в разделе «Поддержка» -> «[Взять оборудование на тест](#)».

Вы можете написать напрямую в службу сервиса по электронной почте sc@qtech.ru.

4.3. Техническая поддержка

Если вам необходимо содействие в вопросах, касающихся нашего оборудования, то можете воспользоваться нашей автоматизированной системой запросов технического сервис-центра helpdesk.qtech.ru.

Телефон Технической поддержки +7 (495) 477-81-18 доб. 0

4.4. Электронная версия документа

Дата публикации 21.11.2022



https://files.qtech.ru/upload/wireless/QWC-VC/QWC-VC_HA_install_guide.pdf