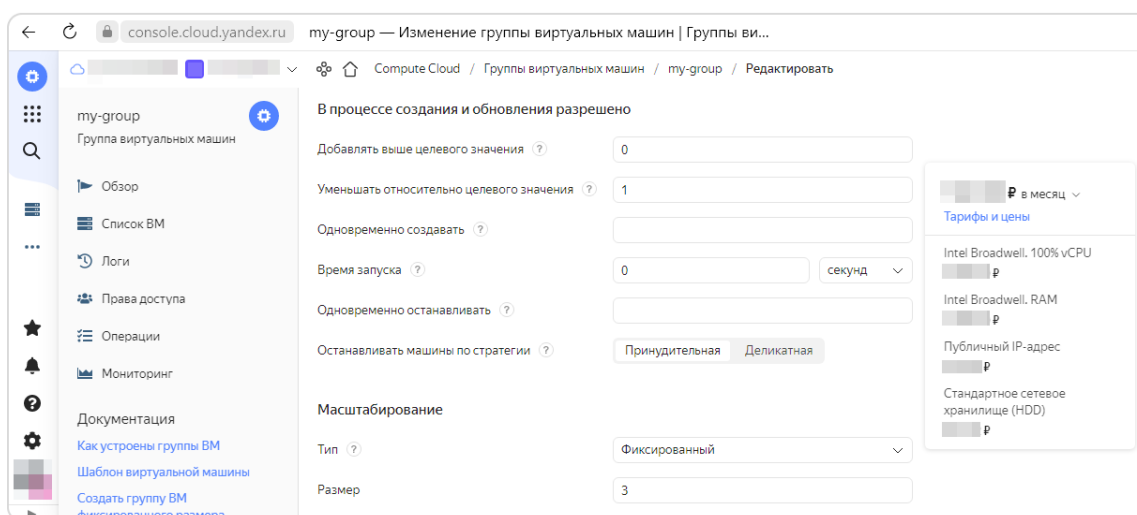


Практическая работа. Обновление приложения

Первый вариант обновления

1. Если вы работаете в консоли управления, измените шаблон VM и выберите образ с ОС Ubuntu 20.04. Убедитесь, что параметры политики развёртывания такие: группу нельзя расширять, а уменьшать можно только на одну VM.



Политика развёртывания группы виртуальных машин (вариант 1)

Если вы работаете в командной строке, в спецификации `specification.yaml` измените параметр `image_id` (например с `fd8s2gbn4d5k2rcf12d9` на `fd8ju9iqf6g5bcq77jns`) и запустите обновление группы:

Скопировать код

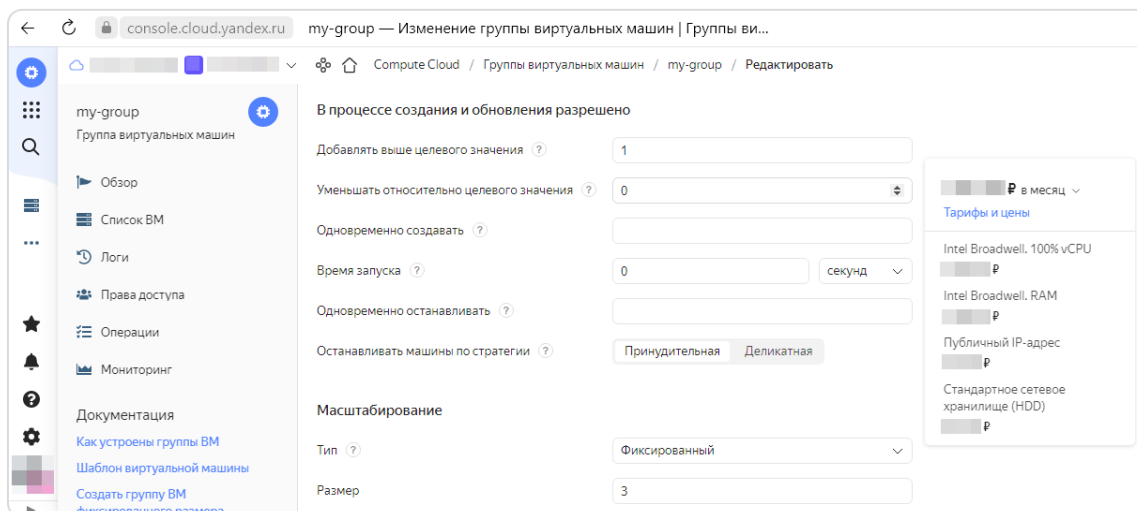
```
yc compute instance-group update \  
  --name my-group \  
  --file <путь_к_файлу_specification.yaml>
```

2. В консоли управления на странице группы VM перейдите на вкладку **Список VM** и проследите, как меняются статусы машин.
3. Сначала вы увидите статус `Running outdated`. Это означает, что машины работают со старой версией приложения.
4. Затем одна из машин начинает обновляться: для неё закрывается трафик (статус `Closing traffic`), она останавливается (статус `Stopping instance`), обновляется (статус `Updating instance`), затем трафик снова открывается (статус `Opening traffic`), и наконец статус меняется на `Running actual`. Обновление выполнено.
5. Затем то же самое последовательно выполняется для остальных машин в группе.

Порядок обновления зависит от политики развёртывания. Мы запретили увеличивать размер группы и указали, что одновременно неработоспособной может быть только одна машина. Именно так и произошло обновление: машины по одной выводились из строя, обновлялись и запускались снова.

Второй вариант обновления

6. Теперь давайте изменим настройки политики развёртывания. Если вы работаете в консоли управления, измените шаблон VM и выберите образ с Ubuntu и NGINX, созданный ранее и помещённый в Container Registry. Измените параметры развёртывания. Теперь группу можно расширять на одну VM, а уменьшать нельзя:



Политика развёртывания группы виртуальных машин (вариант 2)

Если вы работаете в командной строке, в спецификации

`specification.yaml` измените параметр `image_id` (например, снова с `fd8ju9iqf6g5bcq77jns` на `fd8s2gbn4d5k2rcf12d9`). Параметры

обновления измените так:

Скопировать код

```
deploy_policy:
  max_unavailable: 0
  max_expansion: 1
```

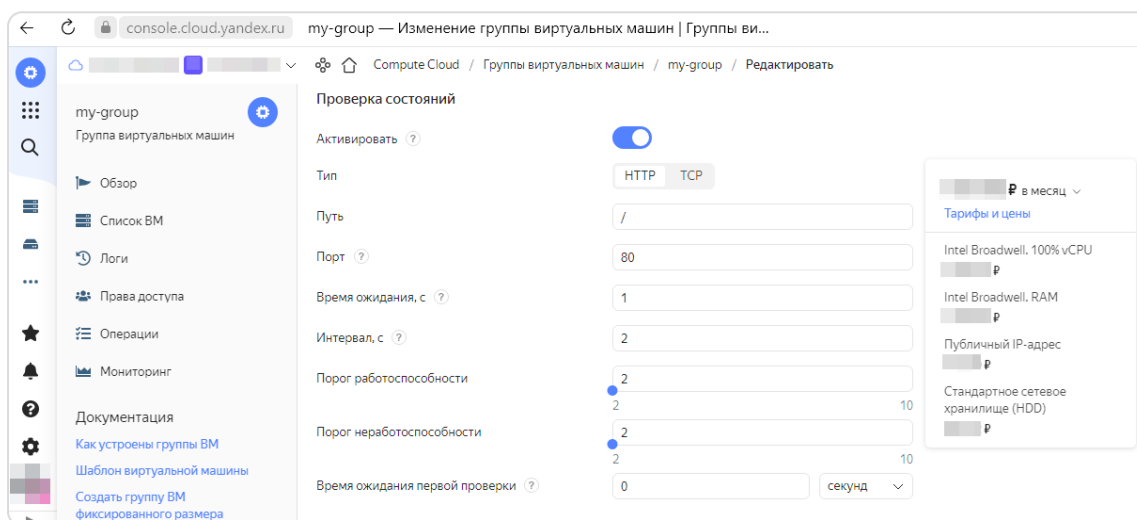
Запустите обновление группы.

7. В консоли управления на странице группы VM перейдите на вкладку **Список VM** и проследите, как меняются статусы машин.
8. Сначала вы увидите статус `Running outdated`. Затем создаётся новая машина (статус `Creating instance`), для неё открывается трафик (статус `Opening traffic`), статус машины меняется на `Running actual`, при этом одна из «устаревших» VM выводится из строя (статусы `Closing traffic` и `Stopping instance`).
9. Затем то же самое последовательно выполняется для остальных машин в группе.

Практическая работа. Сбой приложения

Последний сценарий, который мы рассмотрим, это сбой приложения. Ситуация, когда сама VM работоспособна, но по каким-то причинам произошла ошибка в приложении. Это может быть потеря соединения с базой данных или какой-то баг в запущенном приложении (например утечка памяти). Давайте симулируем такой сценарий. На наших виртуальных машинах запущен только веб-сервер NGINX, давайте остановим его. Но сначала включим проверку состояния VM.

1. В консоли управления откройте вкладку **Обзор** для вашей группы виртуальных машин, нажмите кнопку **Изменить** и активируйте проверку состояний. Сохраните изменения.



2. В браузере откройте страницу с внешним IP-адресом балансировщика, привязанного к вашей группе, и посмотрите, на какую из машин выводится трафик. Узнайте внешний IP-адрес этой машины.
3. В новой вкладке браузера откройте IP-адрес этой виртуальной машины и убедитесь, что выводится приветственная страница, т. е. сервер доступен.

4. Помните, когда вы меняли файл конфигурации для группы машин, вы добавили в него пользователя `my-user`? Теперь он вам пригодится — из консоли зайдите на ВМ от его имени:

Скопировать код

```
ssh my-user@<внешний_IP-адрес_ВМ>
```

5. Посмотрите список запущенных процессов:

Скопировать код

```
ps axu
```

6. Убедитесь, что в списке есть процессы `nginx`:

```
systemd+ 383 0.0 0.5 24536 12012 ?        Ss   11:23   0:00 /lib/systemd/systemd-resol
systemd+ 450 0.0 0.3 27256 6272 ?        Ss   11:23   0:00 /lib/systemd/systemd-netwo
root     529 0.0 0.3 238160 7360 ?        Ssl  11:23   0:00 /usr/lib/accounts-service/a
root     531 0.0 0.1 9412 2912 ?        Ss   11:23   0:00 /usr/sbin/cron -f
message+ 532 0.0 0.2 7512 4544 ?        Ss   11:23   0:00 /usr/bin/dbus-daemon --sys
root     536 0.0 0.1 81924 3632 ?        Ssl  11:23   0:00 /usr/sbin/irqbalance --for
root     538 0.0 0.9 31888 18308 ?       Ss   11:23   0:00 /usr/bin/python3 /usr/bin/
syslog   539 0.0 0.2 224344 4680 ?        Ssl  11:23   0:00 /usr/sbin/rsyslogd -n -iNO
root     541 0.0 0.3 17296 7872 ?        Ss   11:23   0:00 /lib/systemd/systemd-login
root     551 0.0 0.1 8200 2304 ttys0    Ss+  11:23   0:00 /sbin/agetty -o -p -- \u -
root     562 0.0 0.0 8428 1820 tty1     Ss+  11:23   0:00 /sbin/agetty -o -p -- \u -
root     575 0.0 0.3 12172 7392 ?        Ss   11:23   0:00 sshd: /usr/sbin/sshd -D [l
root     579 0.0 1.0 110520 20688 ?       Ssl  11:23   0:00 /usr/bin/python3 /usr/shar
root     584 0.0 0.3 233556 7892 ?        Ssl  11:23   0:00 /usr/lib/policykit-1/polki
root     1300 0.0 0.0 55280 1496 ?        Ss   11:24   0:00 nginx: master process /usr
www-data 1301 0.0 0.2 55844 5192 ?        S    11:24   0:01 nginx: worker process
www-data 1302 0.0 0.2 55844 5192 ?        S    11:24   0:00 nginx: worker process
root     1429 0.0 0.0 0 0 ?        I    11:29   0:00 [kworker/u4:0-events_unbou
root     1476 0.0 0.0 0 0 ?        I    11:38   0:00 [kworker/1:0-events]
root     1480 0.0 0.0 0 0 ?        I    11:39   0:00 [kworker/u4:2-events_power
root     1561 0.0 0.0 0 0 ?        I    11:42   0:00 [kworker/0:1-events]
root     1567 0.0 0.0 0 0 ?        I    11:46   0:00 [kworker/u4:1-events_unbou
root     1575 0.0 0.3 12172 6884 ?        Ss   11:49   0:00 sshd: [accepted]
root     1578 0.0 0.4 13796 9036 ?        Ss   11:49   0:00 sshd: my-user [priv]
my-user  1591 0.1 0.4 18892 9212 ?        Ss   11:49   0:00 /lib/systemd/systemd --use
my-user  1592 0.0 0.1 104636 4040 ?        S    11:49   0:00 (sd-pam)
root     1593 0.0 0.0 0 0 ?        I    11:49   0:00 [kworker/0:0-events]
root     1594 0.0 0.0 0 0 ?        I    11:49   0:00 [kworker/0:2-memcg_kmem_ca
my-user  1617 0.0 0.2 13928 6020 ?        R    11:49   0:00 sshd: my-user@pts/0
my-user  1618 0.0 0.0 2608 1676 pts/0    Ss   11:49   0:00 -sh
my-user  1634 0.0 0.1 11488 3412 pts/0    R+   11:50   0:00 ps axu
$ |
```

6. Теперь остановите эти процессы, чтобы сделать сервер недоступным:

Скопировать код

```
sudo killall nginx
```

7. В браузере обновите страницу балансировщика. Вы увидите, что теперь трафик направляется на другую виртуальную машину группы. Это означает, что Instance Group обнаружил сбой приложения и переключил трафик.
8. Теперь обновите страницу виртуальной машины, на которой вы остановили NGINX. Убедитесь, что сервер теперь недоступен.
9. Откройте список машин вашей группы и проследите, как меняется состояние одной из машин.
Сначала будет закрыт трафик (статус `Closing traffic`), затем виртуальная машина будет остановлена (статус `Stopping instance`), а затем перезапущена (статус `Running actual`).
10. Убедитесь, что веб-сервер на этой VM снова доступен.

Мы проверили четыре основных сценария сбоев и убедились, что Yandex Cloud автоматически обрабатывает их и восстанавливает работоспособность группы.

Теперь вы можете удалить группу виртуальных машин, в этом курсе она больше не понадобится.