Теория.

Принципы отказоустойчивости

Вы наверняка время от времени встречали в новостях сообщения о том, что какой-то крупный интернет-магазин или востребованный сервис вышел из строя и был недоступен в течение нескольких часов или даже суток. Такие ситуации приводят к финансовым потерям (недополученной прибыли или даже штрафам) и становятся ударом по репутации провайдера.
Если вы работаете с юридическими лицами, обязательства по обеспечению доступности обычно фиксируются в виде SLA — Service Level Agreements, соглашении об уровне сервиса. Например, вы гарантируете, что сервис будет доступен 99,95% времени. Если вы работаете с обычными пользователями, формальные SLA могут не заключаться, но вы и сами заинтересованы в том, чтобы ваше приложение работало без сбоев.
Не существует универсального решения, которое устраняло бы все проблемы, связанные с отказоустойчивостью. Но если сочетать различные подходы со стороны разрабатываемого приложения и настройки инфраструктуры, то можно подобрать удачные решения для каждого конкретного случая и минимизировать риски отказа работы системы.
Мы уже затрагивали отдельные вопросы отказоустойчивости в предыдущих уроках: балансировка нагрузки, зоны доступности, автоматическое масштабирование и автоматическое восстановление — все это помогает поддерживать непрерывную работу системы.

Как можно обеспечивать отказоустойчивость

Есть несколько классических подходов к обеспечению отказоустойчивости.

- **Масштабирование**. Система должна быть готова быстро предоставить дополнительные ресурсы при повышении нагрузки и уметь освобождать лишние ресурсы при снижении нагрузки. Этот принцип заложен в облачной архитектуре.

- **Избыточность и резервирование.** Наличие независимых дублирующих компонентов повышает надёжность системы: например, если один диск выйдет из строя, данные можно быстро перенести на запасной диск.
  Отказоустойчивые системы часто строят, создавая несколько копий приложения или его компонентов: баз данных, очередей сообщений и т.д. Реализация может строиться двумя способами: так, чтобы взаимодействие происходило сразу с двумя компонентами одновременно, или так, чтобы постоянно работал только один компонент, а второй находился в ожидании.
- **Мониторинг**. Для быстрого реагирования нужно отслеживать состояние всех компонентов системы и анализировать историю событий. В небольших системах это можно делать вручную. В крупных или критически важных системах необходим автоматический мониторинг с уведомлениями при приближении к критическим значениям параметров. Такой мониторинг называют проактивным — он позволяет не исправлять проблемы, а не допускать их появления. Мониторингу мы посвятим отдельную тему.
- **Реакция на сбои**. Например, система может сама перенести данные с вышедшего из строя диска на резервный или развернуть дополнительные виртуальные машины при повышении нагрузки. Подобные действия тоже входят в проактивный мониторинг и их тоже можно автоматизировать.

Понимайте вашу систему

Чтобы выбрать оптимальное решение, хорошо изучите устройство вашей системы. Мы говорим именно «системы», а не «приложения» или «сервиса», потому что слабым звеном может оказаться не код, а часть инфраструктуры. Например, это могут быть сбои в работе сети или проблемы на стороне провайдера.
Вам нужно понимать слабые места системы — **точки отказа**. Точкой отказа называется компонент, при выведении которого из строя перестает корректно работать вся система. Выход из строя точки отказа часто приводит к неработоспособности других компонентов, что в свою очередь ведёт к отказу следующих. Такие ситуации называют **каскадными сбоями**, или эффектом домино, и они приводят к самым масштабным авариям.

Отказоустойчивость в облаке

При проектировании отказоустойчивых приложений важно понимать, что представляют собой отдельные сущности облачной инфраструктуры в реальном мире.
Например, понятие **зона доступности** — достаточно условное и определяется провайдером инфраструктуры. Какой-то провайдер может назвать три соседние стойки в дата-центре тремя разными зонами доступности.
В Yandex Cloud зоны доступности — это дата-центры в разных регионах России. За счет такого географического разделения обеспечивается более высокая надёжность. Но это и накладывает определенные ограничения — при отказе некоторые сущности физически не смогут мигрировать в другую зону доступности (например кеши записи при отказе диска). Значит, вам нужно настраивать репликацию на более высоком уровне, например, на уровне приложения.

Рассчитывайте целесообразность

Высокая доступность приложения или сервиса требует инвестиций. Когда вы планируете меры по повышению отказоустойчивости, старайтесь соотносить затраты и отдачу от них.
Например, если ваше приложение просто формирует какой-то ежемесячный отчет, вряд ли стоит добиваться для него времени работы 99,999%. И уж точно не стоит уделять большое внимание отказоустойчивости сред разработки и тестирования.
Вот пример оценки вложений в отказоустойчивость для небольшого интернет-магазина: стоит ли бороться за повышение времени доступности с 99% до 99,9%? Судя по этим расчетам, инвестиции окупаются только если выручка магазина составляет не менее 40 тыс. руб. в день.

Упущенная выручка - пример b2c компании

Uptime	Время недоступности в день	Минут недоступности в месяц	Затраты в месяц	Ежедневная выручка	Выручка в минуту	Потеря выручки в месяц
99	14m	7h 18m 17.5s	4 037 ₽	40 000 ₽	27,78 ₽	12 166 ₽
99,9%	43s	43m 49.7s	15 332 ₽	40 000 ₽	27,78 ₽	597 ₽
		Дополнительные инвестиции в отказоустойчивость		11 295 ₽		
		Экономия на отсутствии финансовых потерь		11 569,44 ₽		
		Ежедневная выручка, при которой экономия выше чем инвестиции		40 000 ₽		

Практическая работа. Сбой виртуальной машины

Давайте посмотрим, как принципы построения отказоустойчивых систем реализованы в Yandex Cloud. В практических работах этой темы вы проверите четыре основных сценария отказов:

- сбой виртуальной машины,
- сбой всей зоны доступности,
- обновление приложения
- сбой приложения.

Вы сымитируете эти отказы и понаблюдаете, как Yandex Cloud обеспечивает доступность приложения и восстанавливает инфраструктуру после сбоев.
Начнем с самого простого сценария — сбоя виртуальной машины.

1. Создайте группу из трёх ВМ в трёх зонах доступности под балансировщиком нагрузки. Используйте образ с ОС Ubuntu 18.04 (потом мы обновим его на более свежую версию ОС).

Используйте спецификацию `specification.yaml` из практической работы по CLI Yandex Cloud, но адаптируйте её для того, чтобы на ней можно было проверить разные сценарии сбоев.
Во-первых, будут задействованы все три зоны доступности, поэтому нужно немного исправить блок `allocation_policy`:
Скопировать код

```
allocation_policy:
    zones:
        - zone_id: ru-central1-a
        - zone_id: ru-central1-b
        - zone_id: ru-central1-c
```

Также пропишите подсети для каждой зоны (не забывайте подставлять идентификаторы ваших подсетей):
Скопировать код

```
   network_interface_specs:
        - network_id: <идентификатор_сети>
          subnet_ids:
            - <идентификатор_подсети_№1>
            - <идентификатор_подсети_№2>
            - <идентификатор_подсети_№3>
          primary_v4_address_spec: { one_to_one_nat_spec: { ip_version:
IPV4 }}
```

Во-вторых, в секции `#cloud-config` укажите пользователя, которого нужно создать для входа в виртуальные машины по SSH (это понадобится позднее, на одной из следующих практических работ):

Скопировать код

```
       users:
          - name: my-user
            groups: sudo
            lock_passwd: true
            sudo: 'ALL=(ALL) NOPASSWD:ALL'
            ssh-authorized-keys:
              - ssh-rsa AAAAB3Nza...
```

Обновленный файл спецификации specification.yaml

```
name: my-group
service_account_id: <идентификатор_сервисного_аккаунта>

instance_template:
    platform_id: standard-v1
    resources_spec:
        memory: 2g
        cores: 2
    boot_disk_spec:
        mode: READ_WRITE
        disk_spec:
            image_id: <идентификатор_образа_Ubuntu_18.04>
            type_id: network-hdd
            size: 32g
    network_interface_specs:
        - network_id: <идентификатор_сети>
          subnet_ids:
            - <идентификатор_подсети_№1>
            - <идентификатор_подсети_№2>
            - <идентификатор_подсети_№3>
          primary_v4_address_spec: { one_to_one_nat_spec: { ip_version:
IPV4 }}
    scheduling_policy:
        preemptible: false
    metadata:
      user-data: |-
        #cloud-config
          users:
            - name: my-user
              groups: sudo
              lock_passwd: true
              sudo: 'ALL=(ALL) NOPASSWD:ALL'
              ssh-authorized-keys:
                - <содержимое_публичной_части_SSH-ключа>
          package_update: true
          runcmd:
            - [ apt-get, install, -y, nginx ]
```

```
            - [/bin/bash, -c, 'source /etc/lsb-release; sed -i
"s/Welcome to nginx/It is $(hostname) on $DISTRIB_DESCRIPTION/"
/var/www/html/index.nginx-debian.html']

deploy_policy:
    max_unavailable: 1
    max_expansion: 0
scale_policy:
    fixed_scale:
        size: 3
allocation_policy:
    zones:
        - zone_id: ru-central1-a
        - zone_id: ru-central1-b
        - zone_id: ru-central1-c

load_balancer_spec:
    target_group_spec:
        name: my-target-group
```

Создайте группу по новой спецификации:
Скопировать код

```
yc compute instance-group create --file
<путь_к_файлу_specification.yaml>
```

Если ранее вы удаляли балансировщик нагрузки, создайте его снова и привяжите к целевой группе:
Скопировать код

```
yc load-balancer network-load-balancer create \
  --region-id ru-central1 \
  --name my-load-balancer \
  --listener name=my-listener,external-ip-version=ipv4,port=80 \
  --target-group
target-group-id=<идентификатор_целевой_группы>,healthcheck-name=test-he
alth-check,healthcheck-interval=2s,healthcheck-timeout=1s,healthcheck-u
nhealthythreshold=2,healthcheck-healthythreshold=2,healthcheck-http-por
t=80
```

В консоли управления убедитесь, что ресурсы созданы. Проверьте вывод по внешнему IP-адресу балансировщика — должна отображаться приветственная страница с идентификатором одной из виртуальных машин группы.

2. Начните отслеживать состояние виртуальных машин группы и целевой группы балансировщика:

Скопировать код

```
while true; do \
yc compute instance-group \
  --id <идентификатор_группы_ВМ> list-instances; \
yc load-balancer network-load-balancer \
  --id <идентификатор_балансировщика> target-states \
  --target-group-id <идентификатор_целевой_группы>; \
sleep 5; done
```

Информация выводится в виде таблиц:

```
+----------------------+--------------------------+----------------+--
----------+----------------------+----------------+
|     INSTANCE ID      |           NAME           |  EXTERNAL IP   |
INTERNAL IP |         STATUS         | STATUS MESSAGE |
+----------------------+--------------------------+----------------+--
----------+----------------------+----------------+
| ef34nv4tp3ha8gl6p3df | cl1m5ksvljnq5frekghi-uzex | 84.201.148.207 |
10.128.0.42 | RUNNING_ACTUAL [1m54s] |                |
| ef3nquhoicdq0ccl0tlq | cl1m5ksvljnq5frekghi-iduv | 84.201.171.248 |
10.128.0.9  | RUNNING_ACTUAL [13m]   |                |
| ef3oio9su52imaod7rad | cl1m5ksvljnq5frekghi-ixac | 84.252.132.4   |
10.128.0.37 | RUNNING_ACTUAL [6h]    |                |
+----------------------+--------------------------+----------------+--
----------+----------------------+----------------+

+----------------------+-------------+---------+
|      SUBNET ID       |   ADDRESS   | STATUS  |
+----------------------+-------------+---------+
| b0c4h992tbuodl5hudpu | 10.128.0.37 | HEALTHY |
| e2luooifg8ruecr7g6fk | 10.128.0.6  | HEALTHY |
| e9bn57jvjnbujnmk3mba | 10.128.0.9  | HEALTHY |
+----------------------+-------------+---------+
```

3. Сбой виртуальной машины может произойти из-за падения физического хоста, на котором она запущена. Иногда виртуальную машину могут удалить случайно, по ошибке. Чтобы сымитировать сбой, удалим одну из виртуальных машин в группе через консоль управления.

Если бы это была единственная машина, на которую поступает трафик, система стала бы недоступна. Но у нас система развернута на нескольких виртуальных машинах, поэтому трафик будет перенаправлен на две оставшиеся. Через несколько секунд будет обнаружена проблема, и виртуальная машина будет выведена из-под балансировки. Об этом говорит статус `UNHEALTHY`.

```
+----------------------+--------------------------+----------------+--
----------+----------------------+----------------+
|     INSTANCE ID      |           NAME           |  EXTERNAL IP   |
INTERNAL IP |        STATUS        | STATUS MESSAGE |
```

```
+----------------------+--------------------------+----------------+--
----------+----------------------+----------------+
| ef330no5frc5de91v77n | cl1m5ksvljnq5frekghi-uzex | 84.201.147.33  |
10.128.0.6  | RUNNING_ACTUAL [15m] |                |
| ef3nquhoicdq0ccl0tlq | cl1m5ksvljnq5frekghi-iduv | 84.201.171.248 |
10.128.0.9  | RUNNING_ACTUAL [32m] |                |
| ef3oio9su52imaod7rad | cl1m5ksvljnq5frekghi-ixac | 84.252.132.4   |
10.128.0.37 | RUNNING_ACTUAL [6h]  |                |
+----------------------+--------------------------+----------------+--
----------+----------------------+----------------+

+----------------------+-------------+-----------+
|      SUBNET ID       |   ADDRESS   |  STATUS   |
+----------------------+-------------+-----------+
| b0c4h992tbuodl5hudpu | 10.128.0.37 | HEALTHY   |
| e2luooifg8ruecr7g6fk | 10.128.0.6  | UNHEALTHY |
| e9bn57jvjnbujnmk3mba | 10.128.0.9  | HEALTHY   |
+----------------------+-------------+-----------+
```

4. Далее подсеть перейдет в статус `DRAINING` — ресурс удаляется, и с него снимается трафик. Балансировщик перестает передавать трафик этому ресурсу.

```
+----------------------+--------------------------+----------------+--
----------+----------------------+----------------+
|     INSTANCE ID      |           NAME           |  EXTERNAL IP   |
INTERNAL IP |        STATUS        | STATUS MESSAGE |
+----------------------+--------------------------+----------------+--
----------+----------------------+----------------+
| ef330no5frc5de91v77n | cl1m5ksvljnq5frekghi-uzex | 84.201.147.33  |
10.128.0.6  | CLOSING_TRAFFIC [0s] |                |
| ef3nquhoicdq0ccl0tlq | cl1m5ksvljnq5frekghi-iduv | 84.201.171.248 |
10.128.0.9  | RUNNING_ACTUAL [33m] |                |
| ef3oio9su52imaod7rad | cl1m5ksvljnq5frekghi-ixac | 84.252.132.4   |
10.128.0.37 | RUNNING_ACTUAL [6h]  |                |
+----------------------+--------------------------+----------------+--
----------+----------------------+----------------+

+----------------------+-------------+----------+
|      SUBNET ID       |   ADDRESS   |  STATUS  |
+----------------------+-------------+----------+
| b0c4h992tbuodl5hudpu | 10.128.0.37 | HEALTHY  |
| e2luooifg8ruecr7g6fk | 10.128.0.6  | DRAINING |
| e9bn57jvjnbujnmk3mba | 10.128.0.9  | HEALTHY  |
+----------------------+-------------+----------+


+----------------------+--------------------------+----------------+--
----------+----------------------+----------------+
```

```
|     INSTANCE ID      |           NAME            |  EXTERNAL IP   |
INTERNAL IP |        STATUS        | STATUS MESSAGE |
+----------------------+---------------------------+----------------+--
----------+----------------------+----------------+
| ef330no5frc5de91v77n | cl1m5ksvljnq5frekghi-uzex | 84.201.147.33  |
10.128.0.6  | CLOSING_TRAFFIC [9s] |                |
| ef3nquhoicdq0ccl0tlq | cl1m5ksvljnq5frekghi-iduv | 84.201.171.248 |
10.128.0.9  | RUNNING_ACTUAL [33m] |                |
| ef3oio9su52imaod7rad | cl1m5ksvljnq5frekghi-ixac | 84.252.132.4   |
10.128.0.37 | RUNNING_ACTUAL [6h]  |                |
+----------------------+---------------------------+----------------+--
----------+----------------------+----------------+

+----------------------+-------------+----------+
|      SUBNET ID       |   ADDRESS   |  STATUS  |
+----------------------+-------------+----------+
| b0c4h992tbuodl5hudpu | 10.128.0.37 | HEALTHY  |
| e9bn57jvjnbujnmk3mba | 10.128.0.9  | HEALTHY  |
+----------------------+-------------+----------+
```

5. После этого Instance Group начнет пересоздавать удалённую виртуальную машину. Процесс восстановления может занять некоторое время. Понаблюдаем за ним.

Сначала новая виртуальная машина появится в группе в статусе `CREATING_INSTANCE`.

```
+----------------------+---------------------------+----------------+--
----------+--------------------------+----------------+
|     INSTANCE ID      |           NAME            |  EXTERNAL IP   |
INTERNAL IP |         STATUS           | STATUS MESSAGE |
+----------------------+---------------------------+----------------+--
----------+--------------------------+----------------+
| ef330no5frc5de91v77n | cl1m5ksvljnq5frekghi-uzex | 84.201.147.33  |
10.128.0.6  | CREATING_INSTANCE [-1s] |                |
| ef3nquhoicdq0ccl0tlq | cl1m5ksvljnq5frekghi-iduv | 84.201.171.248 |
10.128.0.9  | RUNNING_ACTUAL [33m]    |                |
| ef3oio9su52imaod7rad | cl1m5ksvljnq5frekghi-ixac | 84.252.132.4   |
10.128.0.37 | RUNNING_ACTUAL [6h]     |                |
+----------------------+---------------------------+----------------+--
----------+--------------------------+----------------+

+----------------------+-------------+----------+
|      SUBNET ID       |   ADDRESS   |  STATUS  |
+----------------------+-------------+----------+
| b0c4h992tbuodl5hudpu | 10.128.0.37 | HEALTHY  |
| e9bn57jvjnbujnmk3mba | 10.128.0.9  | HEALTHY  |
+----------------------+-------------+----------+
```

6. Далее виртуальная машина будет открыта для трафика (статус `OPEN_TRAFFIC`). Балансировщик начнет процесс включения машины в список доступных машин.

```
+----------------------+--------------------------+----------------+--
----------+----------------------+---------------------------------+
|     INSTANCE ID      |           NAME           |  EXTERNAL IP   |
INTERNAL IP |        STATUS        |         STATUS MESSAGE          |
+----------------------+--------------------------+----------------+--
----------+----------------------+---------------------------------+
| ef374t9ghea78p471gal | cl1m5ksvljnq5frekghi-uzex | 84.252.135.153 |
10.128.0.32 | OPENING_TRAFFIC [-1s] | Adding target(s)               |
|                      |                          |                |
|                       | 10.128.0.32 to target group    |
|                      |                          |                |
|                       | b7rh0bhm9f82dglb2p9r           |
| ef3nquhoicdq0ccl0tlq | cl1m5ksvljnq5frekghi-iduv | 84.201.171.248 |
10.128.0.9  | RUNNING_ACTUAL [34m]  |                                 |
| ef3oio9su52imaod7rad | cl1m5ksvljnq5frekghi-ixac | 84.252.132.4   |
10.128.0.37 | RUNNING_ACTUAL [7h]   |                                 |
+----------------------+--------------------------+----------------+--
----------+----------------------+---------------------------------+

+----------------------+-------------+---------+
|      SUBNET ID       |   ADDRESS   | STATUS  |
+----------------------+-------------+---------+
| b0c4h992tbuodl5hudpu | 10.128.0.37 | HEALTHY |
| e9bn57jvjnbujnmk3mba | 10.128.0.9  | HEALTHY |
+----------------------+-------------+---------+


+----------------------+--------------------------+----------------+--
----------+----------------------+---------------------------------+
|     INSTANCE ID      |           NAME           |  EXTERNAL IP   |
INTERNAL IP |        STATUS        |         STATUS MESSAGE          |
+----------------------+--------------------------+----------------+--
----------+----------------------+---------------------------------+
| ef374t9ghea78p471gal | cl1m5ksvljnq5frekghi-uzex | 84.252.135.153 |
10.128.0.32 | OPENING_TRAFFIC [1s] | Adding target(s)               |
|                      |                          |                |
|                      | 10.128.0.32 to target group    |
|                      |                          |                |
|                      | b7rh0bhm9f82dglb2p9r           |
| ef3nquhoicdq0ccl0tlq | cl1m5ksvljnq5frekghi-iduv | 84.201.171.248 |
10.128.0.9  | RUNNING_ACTUAL [34m] |                                 |
| ef3oio9su52imaod7rad | cl1m5ksvljnq5frekghi-ixac | 84.252.132.4   |
10.128.0.37 | RUNNING_ACTUAL [7h]  |                                 |
+----------------------+--------------------------+----------------+--
----------+----------------------+---------------------------------+

+----------------------+-------------+---------+
|      SUBNET ID       |   ADDRESS   | STATUS  |
```

```
+----------------------+-------------+---------+
| b0c4h992tbuodl5hudpu | 10.128.0.37 | HEALTHY |
| e2luooifg8ruecr7g6fk | 10.128.0.32 | INITIAL |
| e9bn57jvjnbujnmk3mba | 10.128.0.9  | HEALTHY |
+----------------------+-------------+---------+

+----------------------+--------------------------+----------------+--
----------+----------------------+---------------------------------+
|     INSTANCE ID      |           NAME           |  EXTERNAL IP   |
INTERNAL IP |        STATUS        |         STATUS MESSAGE          |
+----------------------+--------------------------+----------------+--
----------+----------------------+---------------------------------+
| ef374t9ghea78p471gal | cl1m5ksvljnq5frekghi-uzex | 84.252.135.153 |
10.128.0.32 | OPENING_TRAFFIC [18s] | Awaiting HEALTHY state for     |
|                      |                          |                |
|                       | target(s) 10.128.0.32. Elapsed |
|                      |                          |                |
|                       | time: 3s.                      |
| ef3nquhoicdq0ccl0tlq | cl1m5ksvljnq5frekghi-iduv | 84.201.171.248 |
10.128.0.9  | RUNNING_ACTUAL [34m]  |                                  |
| ef3oio9su52imaod7rad | cl1m5ksvljnq5frekghi-ixac | 84.252.132.4   |
10.128.0.37 | RUNNING_ACTUAL [7h]   |                                  |
+----------------------+--------------------------+----------------+--
----------+----------------------+---------------------------------+

+----------------------+-------------+----------+
|      SUBNET ID       |   ADDRESS   |  STATUS  |
+----------------------+-------------+----------+
| b0c4h992tbuodl5hudpu | 10.128.0.32 | INITIAL  |
| b0c4h992tbuodl5hudpu | 10.128.0.37 | INACTIVE |
| b0c4h992tbuodl5hudpu | 10.128.0.9  | HEALTHY  |
+----------------------+-------------+----------+

+----------------------+--------------------------+----------------+--
----------+-------------------------+---------------------------------+
|     INSTANCE ID      |           NAME           |  EXTERNAL IP   |
INTERNAL IP |         STATUS          |         STATUS MESSAGE
|
+----------------------+--------------------------+----------------+--
----------+-------------------------+---------------------------------+
| ef374t9ghea78p471gal | cl1m5ksvljnq5frekghi-uzex | 84.252.135.153 |
10.128.0.32 | OPENING_TRAFFIC [1m32s] | [NLB unhealthy]; Awaiting
|
|                      |                          |                |
|                         | HEALTHY state for target(s)    |
|                      |                          |                |
|                         | 10.128.0.32. Elapsed time: 1m  |
|                      |                          |                |
|                         | 17s.                           |
```

```
| ef3nquhoicdq0ccl0tlq | cl1m5ksvljnq5frekghi-iduv | 84.201.171.248 |
10.128.0.9  | RUNNING_ACTUAL [35m]    |
|
| ef3oio9su52imaod7rad | cl1m5ksvljnq5frekghi-ixac | 84.252.132.4   |
10.128.0.37 | RUNNING_ACTUAL [7h]     |
|
+----------------------+---------------------------+----------------+--
----------+------------------------+---------------------------------+

+----------------------+-------------+---------+
|      SUBNET ID       |   ADDRESS   | STATUS  |
+----------------------+-------------+---------+
| b0c4h992tbuodl5hudpu | 10.128.0.37 | HEALTHY |
| e2luooifg8ruecr7g6fk | 10.128.0.32 | HEALTHY |
| e9bn57jvjnbujnmk3mba | 10.128.0.9  | HEALTHY |
+----------------------+-------------+---------+
```

7. И в завершение подсеть перейдет в статус `HEALTHY`, а машина — в статус `RUNNING_ACTUAL`, и трафик будет снова разделен между тремя машинами.

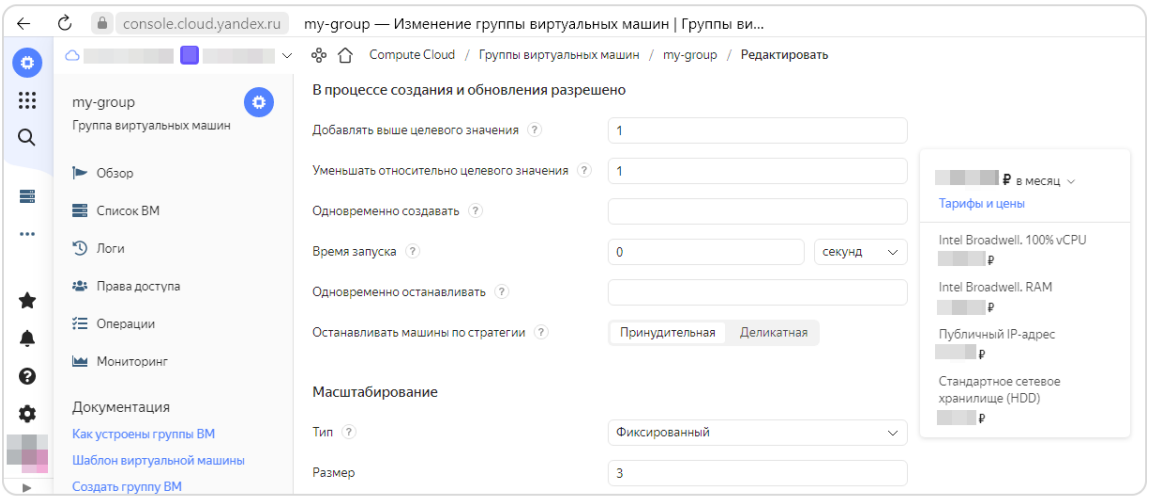
```
+----------------------+---------------------------+----------------+--
----------+----------------------+----------------+
|     INSTANCE ID      |           NAME            |  EXTERNAL IP   |
INTERNAL IP |        STATUS        | STATUS MESSAGE |
+----------------------+---------------------------+----------------+--
----------+----------------------+----------------+
| ef374t9ghea78p471gal | cl1m5ksvljnq5frekghi-uzex | 84.252.135.153 |
10.128.0.32 | RUNNING_ACTUAL [-1s] |                |
| ef3nquhoicdq0ccl0tlq | cl1m5ksvljnq5frekghi-iduv | 84.201.171.248 |
10.128.0.9  | RUNNING_ACTUAL [35m] |                |
| ef3oio9su52imaod7rad | cl1m5ksvljnq5frekghi-ixac | 84.252.132.4   |
10.128.0.37 | RUNNING_ACTUAL [7h]  |                |
+----------------------+---------------------------+----------------+--
----------+----------------------+----------------+

+----------------------+-------------+---------+
|      SUBNET ID       |   ADDRESS   | STATUS  |
+----------------------+-------------+---------+
| b0c4h992tbuodl5hudpu | 10.128.0.37 | HEALTHY |
| e2luooifg8ruecr7g6fk | 10.128.0.32 | HEALTHY |
| e9bn57jvjnbujnmk3mba | 10.128.0.9  | HEALTHY |
+----------------------+-------------+---------+
```

Восстановление произошло автоматически без ручного вмешательства.
Обратите внимание! Эту группу виртуальных машин мы будем использовать и в трех следующих практических работах, не удаляйте её. Если вы будете делать большой перерыв между практическими работами, вы можете остановить группу (чтобы не расходовать средства на балансе), а затем запустить её снова.

Практическая работа. Сбой зоны доступности

В этом сценарии рассмотрим ситуацию, когда произошел сбой сразу всей зоны доступности. Такие ситуации возникают крайне редко и могут быть связаны с какими-то масштабными стихийными бедствиями, однако и их стоит предусмотреть.
Посмотрим, как будет решаться проблема неожиданного выхода из строя зоны доступности.
В нашем примере используются все три зоны доступности — `ru-central1-a`, `ru-central1-b` и `ru-central1-c`. В каждой зоне располагается одна ВМ.

1. Установите такие настройки политики развертывания — пусть группу можно расширять на 1 ВМ и уменьшать на 1 ВМ:



2. Теперь в настройках группы виртуальных машин уберите одну зону доступности, например `ru-central1-c`. Переключитесь на вкладку **Список ВМ** и посмотрите, что будет происходить.
3. Для ВМ, которая располагалась в зоне `ru-central1-c`, отключается трафик (статус `Closing traffic`), а затем сама машина удаляется (статус `Deleting instance`). Одновременно в другой зоне доступности создаётся и запускается новая ВМ.

Остальные машины в группе продолжают работать без изменений.

Таким образом, даже при выходе из строя всей зоны доступности группа виртуальных машин продолжит работать и будет способна принимать прежнюю нагрузку.