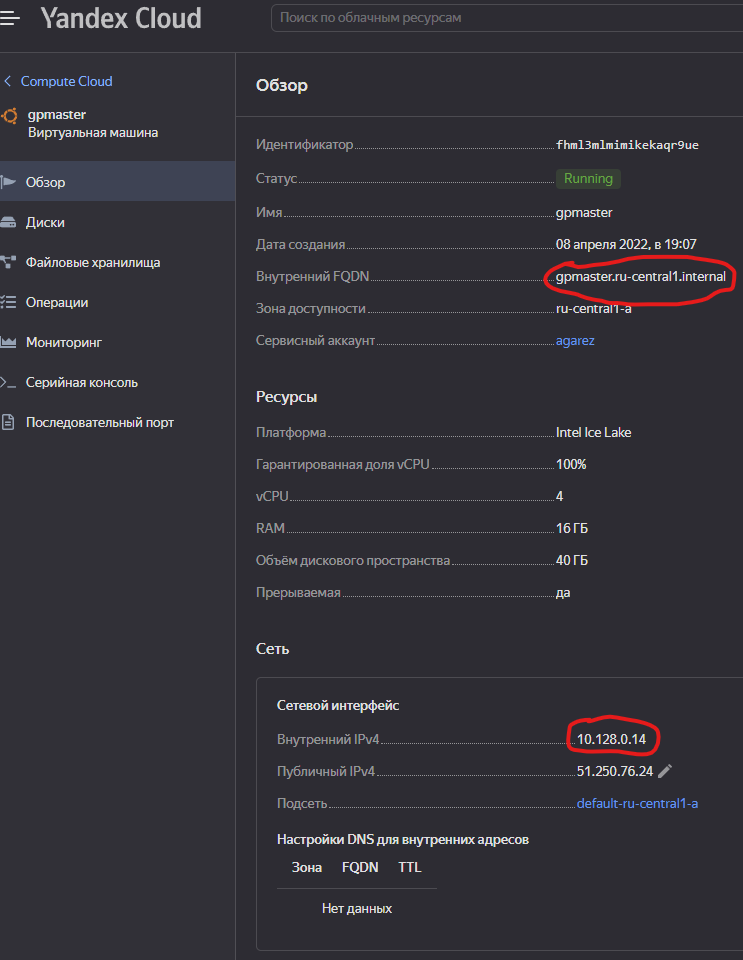
# Развертывание кластера GreenPlum в Yandex Cloud на ВМ

Полная официальная инструкция от pivotal - [по ссылке](https://greenplum.docs.pivotal.io/6-20/install_guide/install_guide.html). (довольно запутанная) Вот [эта](https://sandeepkattepogu.medium.com/greenplum-6-7-1-on-aws-4d368feea486) чуть понятнее, но покрывает не всё.

1. Создаем виртуальные машины под кластер
   1. 1 нужна под мастер и от 1 до 200 под сегменты
   2. Количество vCPU влияет на оптимальное число сегментов. Самое простое – 1 сегмент на 1 vCPU, учитывая mirror сегменты. Например, ВМ с 4 vCPU – 2 сегмента + 2 mirror сегмента.
   3. ОС должна быть из [списка поддерживаемых](https://greenplum.docs.pivotal.io/6-20/install_guide/platform-requirements.html), например Ubuntu 18.04 LTS
   4. Системные требования с офф сайта такие. Но у нас не промышленный сервер, можно обойтись меньшим количеством ОЗУ.
   5. Общий объем дисков SSD, выдаваемый под тестовый период – не больше 200 ГБ. Если хочется суммарно больше, нужно использовать HDD.

|  |  |
| --- | --- |
| Minimum CPU | Any x86\_64 compatible CPU |
| Minimum Memory | 16 GB RAM per server |
| Disk Space Requirements | * 150MB per host for Greenplum installation * Approximately 300MB per segment instance for metadata * Cap disk capacity at 70% full to accommodate temporary files and prevent performance degradation |
| Network Requirements | 10 Gigabit Ethernet within the array  NIC bonding is recommended when multiple interfaces are present  Greenplum Database can use either IPV4 or IPV6 protocols. |

1. Для каждой машины копируем имя хоста и внутренний айпишник из консоли Yandex cloud. В моём случае это 5 хостов.



|  |  |  |
| --- | --- | --- |
| Хост | Ip | назначение |
| gpmaster.ru-central1.internal | 10.128.0.14 | Master (mdw) |
| gpseghost1.ru-central1.internal | 10.128.0.33 | Segment host 1(sdw1) primary |
| gpseghost2.ru-central1.internal | 10.128.0.5 | Segment host 2(sdw2) primary |
| gpseghost3.ru-central1.internal | 10.128.0.22 | Segment host 3(sdw3) primary |
| gpseghost4.ru-central1.internal | 10.128.0.37 | Segment host 4(sdw4) primary |

1. На всех машинах:
   1. Обновляем установщик, ставим питон (он есть в зависимостях, но автоматом сам не ставится) и пакет с GP

sudo apt update && sudo apt upgrade -y -q && sudo apt install software-properties-common

sudo add-apt-repository ppa:greenplum/db

sudo apt update

sudo apt install -y --no-install-recommends python

sudo apt install -y greenplum-db-6

* 1. Добавляем в файл /etc/hosts

sudo nano /etc/hosts

10.128.0.14 mdw

10.128.0.33 sdw1

10.128.0.5 sdw2

10.128.0.22 sdw3

10.128.0.37 sdw4

* 1. В файле /etc/security/limits.conf добавляем

\* soft nofile 524288

\* hard nofile 524288

\* soft nproc 131072

\* hard nproc 131072

* 1. Вычисляем

Значение для kernel.shmall

echo $(expr $(getconf \_PHYS\_PAGES) / 2)

2053120

Значение для kernel.shmmax

echo $(expr $(getconf \_PHYS\_PAGES) / 2 \\* $(getconf PAGE\_SIZE))

8409579520

Значение для vm.min\_free\_kbytes

free

total

16424964

16424964 \* 0.03 = 492748

* 1. Указываем в /etc/sysctl.conf

kernel.shmall = 2053120

kernel.shmmax = 8409579520

kernel.shmmni = 4096

vm.overcommit\_memory = 2 # See Segment Host Memory

vm.overcommit\_ratio = 95 # See Segment Host Memory

vm.min\_free\_kbytes = 492748

net.ipv4.ip\_local\_port\_range = 10000 65535 # See Port Settings

kernel.sem = 250 2048000 200 8192

kernel.sysrq = 1

kernel.core\_uses\_pid = 1

kernel.msgmnb = 65536

kernel.msgmax = 65536

kernel.msgmni = 2048

net.ipv4.tcp\_syncookies = 1

net.ipv4.conf.default.accept\_source\_route = 0

net.ipv4.tcp\_max\_syn\_backlog = 4096

net.ipv4.conf.all.arp\_filter = 1

net.core.netdev\_max\_backlog = 10000

net.core.rmem\_max = 2097152

net.core.wmem\_max = 2097152

vm.swappiness = 10

vm.zone\_reclaim\_mode = 0

vm.dirty\_expire\_centisecs = 500

vm.dirty\_writeback\_centisecs = 100

vm.dirty\_background\_ratio = 3 # Так как меньше 64 GB оперативки

vm.dirty\_ratio = 10 # Так как меньше 64 GB оперативки

#vm.dirty\_background\_bytes = 1610612736 # Так как меньше 64 GB оперативки

#vm.dirty\_bytes = 4294967296 # Так как меньше 64 GB оперативки

* 1. Перегружаем

sudo sysctl -p

* 1. Создаем юзера gpadmin и группу, делаем ему консоль по умолчанию

sudo groupadd gpadmin

sudo useradd -d /home/gpadmin -s /bin/bash -m gpadmin -g gpadmin

sudo passwd gpadmin (12345)

* 1. Добавляем юзера gpadmin в группу и даем доступ к sudo

sudo usermod -a -G gpadmin gpadmin

sudo usermod -a -G sudo gpadmin

1. На мастере:
   1. Копируем на мастер приватный ключ от админского аккаунта (в моем случае – agarez) от всех ВМ, в моем случае он один.
   2. Генерим ключ для gpadmin. В качестве названия я выбрал sshmstr

ssh-keygen

* 1. Копируем приватный и публичные ключи от gpadmin на все машины

scp /home/agarez/sshmstr 10.128.0.33:/home/agarez/

scp /home/agarez/sshmstr.pub 10.128.0.33:/home/agarez/

scp /home/agarez/sshmstr 10.128.0.5:/home/agarez/

scp /home/agarez/sshmstr.pub 10.128.0.5:/home/agarez/

scp /home/agarez/sshmstr 10.128.0.22:/home/agarez/

scp /home/agarez/sshmstr.pub 10.128.0.22:/home/agarez/

scp /home/agarez/sshmstr 10.128.0. 37:/home/agarez/

scp /home/agarez/sshmstr.pub 10.128.0.37:/home/agarez/

1. На всех машинах:
   1. Переносим ключи из админского аккаунта в папку .ssh gpadmin и передаем права

sudo mkdir -p /home/gpadmin/.ssh

sudo cp sshmstr /home/gpadmin/.ssh

sudo cp sshmstr.pub /home/gpadmin/.ssh

sudo chown gpadmin /home/gpadmin/.ssh

sudo chown gpadmin /home/gpadmin/.ssh/sshmstr

sudo chown gpadmin /home/gpadmin/.ssh/sshmstr.pub

* 1. Переключаемся на gpadmin, устанавливаем ключам права доступа, добавляем в authorized\_keys и config

su gpadmin

cd $home

chmod 700 /home/gpadmin/.ssh

chmod 600 /home/gpadmin/.ssh/sshmstr

chmod 600 /home/gpadmin/.ssh/sshmstr.pub

cat /home/gpadmin/.ssh/sshmstr.pub >> /home/gpadmin/.ssh/authorized\_keys

chmod 600 /home/gpadmin/.ssh/authorized\_keys

echo IdentityFile /home/gpadmin/.ssh/sshmstr >> /home/gpadmin/.ssh/config

exit

sudo systemctl restart ssh

* 1. С каждой машины проверяем, что можно подключиться под gpadmin на все остальные машины по SSH не указывая доп параметры

su gpadmin

ssh [gpadmin@10.128.0.14](mailto:gpadmin@10.128.0.14)

exit

ssh [gpadmin@10.128.0.33](mailto:gpadmin@10.128.0.33)

exit

ssh [gpadmin@10.128.0.5](mailto:gpadmin@10.128.0.5)

exit

ssh [gpadmin@10.128.0.22](mailto:gpadmin@10.128.0.22)

exit

ssh [gpadmin@10.128.0.37](mailto:gpadmin@10.128.0.37)

exit

* 1. Здесь и дальше всё делаем под gpadmin
  2. На всех машинах запускаем добавление переменных среды (здесь для версии 6.21.0)

source /opt/greenplum-db-6.21.0/greenplum\_path.sh

* 1. Проверяем, что добавилось

which gpssh

Ответ: /opt/greenplum-db-6.21.0/bin/gpssh

* 1. Добавялем строчки к bashsrc на мастере

source /opt/greenplum-db-6.21.0/greenplum\_path.sh

echo "source /opt/greenplum-db-6.21.0/greenplum\_path.sh; export MASTER\_DATA\_DIRECTORY=/gpdata/master/gpseg-1; cd ~" >> ~/.bashrc; source ~/.bashrc

* 1. Меняем владельца директории с GP

sudo chown -R gpadmin:gpadmin $GPHOME

* 1. Создаем директории для данных и конфига

На мастере

sudo mkdir -p /home/gpadmin/gpdata/master;

sudo mkdir -p /home/gpadmin/gpdata/mirror;

sudo mkdir -p /home/gpadmin/gpconfig;

sudo chown -R gpadmin:gpadmin /gpdata

На сегментах

sudo mkdir -p /home/gpadmin/gpdata/primary;

sudo mkdir -p /home/gpadmin/gpdata/mirror;

sudp chown -R gpadmin:gpadmin /gpdata;

1. На мастере:
   1. Копируем дефолтные конфиги из примеров

sudo cp $GPHOME/docs/cli\_help/gpconfigs/gpinitsystem\_config /home/gpadmin/gpconfig/;

sudo cp $GPHOME/docs/cli\_help/gpconfigs/hostfile\_exkeys /home/gpadmin/gpconfig/

sudo cp $GPHOME/docs/cli\_help/gpconfigs/hostfile\_gpssh\_segonly /home/gpadmin/gpconfig/

* 1. В hostfile\_exkeys оставляем список всех хостов

mdw

sdw1

sdw2

sdw3

sdw4

* 1. В hostfile\_segonly оставляем список только сегментных хостов

sdw1

sdw2

sdw3

sdw4

* 1. В gpinitsystem\_config правим (повторения в DATA\_DIRECTORY – по числу сегментов на хост)

MASTER\_DIRECTORY=/gpdata/master

declare -a DATA\_DIRECTORY=(/gpdata/primary /gpdata/primary)

MACHINE\_LIST\_FILE=/home/gpadmin/gpconfig/hostfile\_gpssh\_segonly

* 1. Проверяем, что GP может ходить на сегменты без пароля и GP поставлен и добавлен в окружение на всех сегментах – везде всё должно быть одинаковое

gpssh -f /home/gpadmin/gpconfig/hostfile\_gpssh\_segonly -e ls -l $GPHOME

* 1. Запускаем инициализацию кластера

gpinitsystem -c /home/gpadmin/gpconfig/gpinitsystem\_config

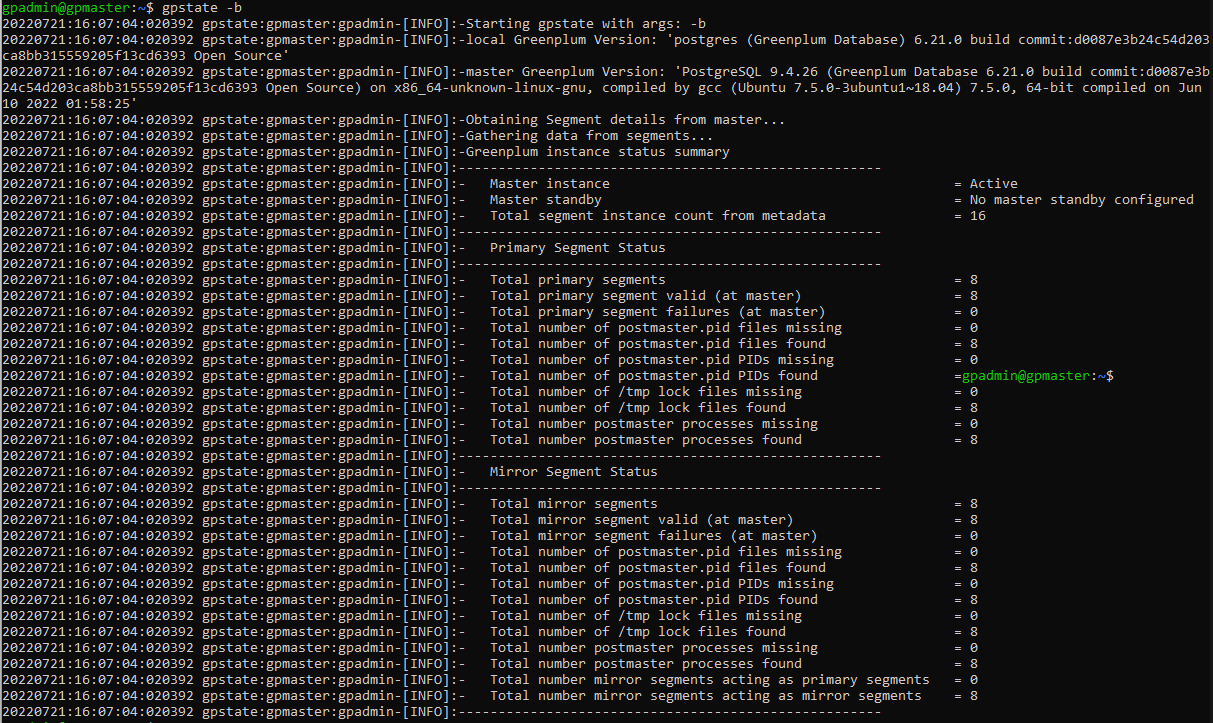
* 1. Когда спросит «Continue with Greenplum creation Yy|Nn (default=N):» говорим Y
  2. Надпись «Greenplum Database instance successfully created» в логе скажет об успешном создании инстанса
  3. Добавляем зеркала

gpaddmirrors -p 10000

Когда спросит папки, отвечаем /gpdata/mirror

* 1. Проверяем состояние кластера

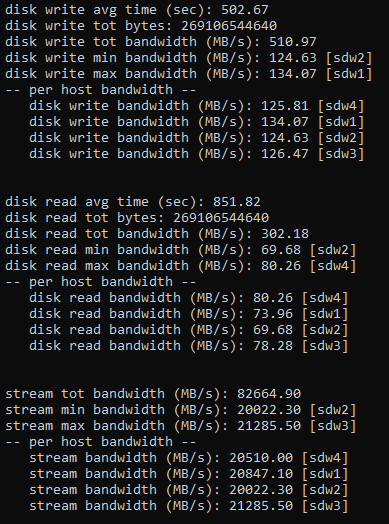
gpstate -b



* 1. Итого у нас кластер на 4-х хост-сегментах, 16 сегментах, 8 основных, 8 мастеров, по 1 сегменту на ядро.
  2. Смотрим производительность

gpcheckperf -f /home/gpadmin/gpconfig/hostfile\_gpssh\_segonly -r ds -D -d /gpdata/primary -d /gpdata/primary

Работает:



1. Если нужно, настраиваем pg\_hba.conf на мастере: $MASTER\_DATA\_DIRECTORY/pg\_hba.conf и перегружаем настройки: gpstop -u
2. Подключаемся к кластеру

psql -d postgres -U gpadmin

Успех!

1. Базовые команды для управления кластером:

Start the database cluster:

gpstart

Restart the database cluster:

gpstop -r

Reload the configuation: pg\_hba.conf and postgresql.conf file

gpstop -u

Stop the database cluster:

gpstop