

# DRE Advanced Media Platform STREAMER

## Руководство по установке

Индекс	STREAMER-IG
Конфиденциальность	Публичный - L0
Ревизия	1.0
Статус	Согласован

## Содержание

1. Аннотация .....	3
2. Общее описание .....	4
3. Минимальные системные требования .....	5
4. Необходимая конфигурация для установки .....	6
5. Установка инфраструктуры .....	7
6. Установка компонентов STREAMER .....	12
7. Развёртывание Serh .....	13
7.1. База .....	13
7.2. Переменные окружения в GitLab Variables .....	14
7.3. Развёртывание кластера .....	14
7.4. Создание пользователя и получение ключей .....	14
7.5. Обновление кластера: .....	15
8. Сторонние зависимости .....	16

## 1. Аннотация

Документ предназначен для технических специалистов, занимающихся установкой, настройкой и поддержкой сервиса. Документ рассчитан на инженеров, обладающих специальными навыками и знаниями в области инсталляции программного обеспечения.

Данный документ опубликован исключительно с целью изучения системных требований для установки продукта, а также ознакомления с последовательностью и деталями процесса установки. Реальная установка продукта производится с использованием внутренних репозиториев ООО "Цифра", доступ к которым предоставляется заказчику по запросу.

## 2. Общее описание

DRE Advanced Media Platform STREAMER (далее - STREAMER) - общее название группы сервисов, обеспечивающих вещание потокового (Live) контента, а так же VOD, рекламного и Catch-Up контента через интернет.

### 3. Минимальные системные требования

Для установки сервиса необходимо наличие не менее 4 серверов.

Сервера должны удовлетворять следующим требованиям:

1. Операционная система ubuntu-20.04-server-amd64 (с установленным пакетом sudo).
2. Многоядерный центральный процессор с тактовой частотой каждого ядра 2 ГГц (не менее 8 ядер).
3. Объем оперативной памяти 64 ГБ.
4. 4 жестких диска емкостью по 500 ГБ. Основной жесткий диск (системный) не менее 256 ГБ и RAID 1.
5. Минимум - один интерфейс Ethernet 10 Гбит/с, желательно 2 интерфейса по 10 Гбит/с. Первый интерфейс будет предназначен для внешнего трафика, а также для трафика kubernetes. Второй интерфейс будет предназначен для ребалансировки и восстановления кластера Serph.
6. Свободное место для папки временных файлов /tmp - 10 ГБ.

## 4. Необходимая конфигурация для установки

При конфигурации серверов следуйте следующим инструкциям:

1. Рекомендуется назвать ноды, на которых будет развернута система: k8s-ceph-node-1, k8s-ceph-node-2 или иначе, но не используя название master, slave-1, slave-2 и т.д. Нужно обязательно использовать разные hostname.
2. IP адреса должны быть заданы статически, без использования dhcp.
3. Установите kubernetes версии 1.21 через kubespray (или иным способом):
  - a. с сетью flannel , kube\_proxy\_mode: iptables, enable\_nodelocaldns: true;
  - b. с включенным helm 3.6.2;
  - c. с выключенным дашбордом (dashboard\_enabled: false);
  - d. выключенным ingress (ingress\_nginx\_enabled: false);
  - e. включенным metrics\_server\_enabled. Ссылка на пример предоставляется по запросу заказчика.
4. Установите метку (label):
  - a. на нодах, через которые будет доступен public ingress, - external;
  - b. на нодах, через которые будет доступен private ingress, - internal.
5. Установите кластер Ceph по инструкции (пункт 6 настоящего руководства) и создайте пользователя.
6. Проверьте работоспособность всех компонентов - Kubernetes, Ceph.
7. Для доступа сервисов к внешним ресурсам по именам (например, DRE Advanced Media Platform META DATA SERVER, источникам потоков, и проч) требуется настройка разбора имен DNS на всех узлах кластера. Для разных дистрибутивов ОС эти настройки могут отличаться, необходимо обращаться к документации используемого дистрибутива. Например, для Ubuntu 20.04, операционная система использует для разрешения имен сервис systemd-resolved, работающий как кэш имен. Он конфигурируется либо в файле настройки сетевой конфигурации, располагающемся в /etc/netplan/, либо непосредственно в конфигурационном файле /etc/systemd/resolved.conf

После внесения изменений в конфигурацию systemd-resolved следует рестартовать этот сервис командой `sudo service systemd-resolved restart`

По умолчанию kubelet - управляющий компонент кластерного узла, ищет конфигурацию разрешения имен по стандартному для UNIX пути /etc/resolv.conf, для корректной работы необходимо поменять этот путь в файле /etc/kubernetes/kubelet-config.yaml изменением (добавлением) параметра resolvConf: `"/run/systemd/resolve/resolv.conf"`

После внесения изменений в конфигурацию kubelet следует рестартовать этот сервис командой `sudo service kubelet restart`

Для применения указанных изменений в работающем кластере нужно рестартовать все поды coredns, nodelocaldns а также поды с параметром dnsPolicy: Default или работающие с сетью хоста (см <https://kubernetes.io/docs/concepts/services-networking/dns-pod-service/>).

## 5. Установка инфраструктуры

Настройте kubernetes для доступа runner к кластеру:

1. Перейдите в раздел Settings->CI/CD->Secret variables.
2. Создайте переменную kube\_config и в её значении пропишите содержимое файла .kube/config с любой мастер-ноды Kubernetes-кластера, зашифрованное кодировкой base64.

Установите через GitLab CI/CD следующую инфраструктуру и настройте, версии чартов переданы в проекте:

- dashboard
- ingress-release (необходима настройка PROD\_CERT и PROD\_KEY)
- ingress-int-release (необходима настройка PROD\_CERT и PROD\_KEY)
- keepalived-release (необходима настройка Ips и servers)
- ceph-csi-rbd (как дефолтное хранилище)
- prometheus-operator

```

repositories:
  - name: ngingress
    url: https://kubernetes.github.io/ingress-nginx
  - name: chartmuseum
    url: https://chartmuseum.gs-labs.tv/svc-dep/
  - name: dashboard
    url: https://kubernetes.github.io/dashboard/
  - name: incubator
    url: https://charts.helm.sh/incubator

releases:
  - name: dashboard
    chart: dashboard/kubernetes-dashboard
    namespace: kubernetes-dashboard
    version: 5.0.4
    values:
      - replicaCount: 4
        enableSkipLogin: true
        enableInsecureLogin: true
        extraArgs:
          - --enable-skip-login
          - --enable-insecure-login
        service:
          type: ClusterIP
          externalPort: 443
        resources:
          limits:
            cpu: 1000m
            memory: 1000Mi
          requests:
            cpu: 300m
            memory: 300Mi
        ingress:
          enabled: true
          annotations:
            kubernetes.io/ingress.class: nginxint
          paths:
            - /
          hosts:
            - k8s.streamer.test.gs-labs.tv
        rbac:
          create: true
  
```

```

    clusterAdminRole: true
  serviceAccount:
    create: true
    name: dashboard
  metricsScraper:
    enabled: true

- name: rbac
  namespace: kubernetes-dashboard
  chart: incubator/raw
  version: 0.2.4
  values:
    - resources:
      - apiVersion: rbac.authorization.k8s.io/v1
        kind: ClusterRoleBinding
        metadata:
          name: kubernetes-dashboard-admin
          namespace: kubernetes-dashboard
        roleRef:
          apiGroup: rbac.authorization.k8s.io
          kind: ClusterRole
          name: cluster-admin
        subjects:
          - kind: ServiceAccount
            name: dashboard
            namespace: kubernetes-dashboard

##### INGRESS #####
- name: ingress-release
  namespace: ingress
  chart: ngingress/ingress-nginx
  version: 4.0.16
  values:
    ### REQUIRED ###
    - controller:
      admissionWebhooks:
        enabled: false
      ingressClassResource:
        name: nginxprod
      defaultTls:
        crt: {{ env "PROD_CERT" }}
        key: {{ env "PROD_KEY" }}
    ### OPTIONAL ###
    # sysctls:
    # net.core.somaxconn: "65535"
    # net.ipv4.netfilter.ip_conntrack_max: "1048576"
    # net.netfilter.nf_conntrack_max: "1048576"
    # fs.file-max: "2097152"
    # net.ipv4.tcp_max_syn_backlog: "65535"
    # net.ipv4.tcp_syncookies: "1"
    # net.ipv4.tcp_timestamps: "1"
    # net.ipv4.ip_local_port_range: "1024 65535"
    config:
      ssl-redirect: "false"
      log-format-upstream: '{"timestamp":"$msec","origin": "nginx", "path": "$request_uri",
"time_request": $request_time, "query_string": "$query_string", "time_backend_connect":
"$upstream_connect_time", "time_backend_response": "$upstream_response_time", "host": "$host", "method":
"$request_method", "user_agent": "$http_user_agent", "http_status_code": $status, "bytes_sent":
$bytes_sent, "scheme": "$scheme", "request_proto": "$server_protocol", "connection_requests":
$connection_requests, "upstream_addr": "$upstream_addr", "upstream_http_server":
"$upstream_http_server", "x_forwarded_for": "$http_x_forwarded_for", "x_serial_number":
"$http_x_serial_number", "cid": "$http_x_correlation_id", "hwid": "$http_x_hwid",
"x_domain_code": "$http_x_domain_code", "client": "$remote_addr", "drm_token": "$http_drm_token", "x-drm-
token": "$http_x_drm_token"}'
      log-format-escape-json: "true"
      gzip-level: "2"
      proxy-buffering: "on"

```



```

    allow-backend-server-header: "true"
    enable-modsecurity: "false"
    skip-access-log-urls: "/healthz"
    worker-processes: "auto"
    proxy-stream-timeout: "600s"
    client-header-timeout: "60"
    client-body-timeout: "60"
    max-worker-connections: "65535"
    keep-alive: "75"
    limit-rate: "0"
    limit-rate-after: "0"
    limit-rps: "10"
    load-balance: "round_robin"
    ssl-protocols: "TLSv1 TLSv1.1 TLSv1.2"
    ssl-ciphers: "ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-
SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:
ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-
AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-
SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA:DHE-RSA-AES256-
SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:AES:CAMELLIA:
DES-CBC3-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-
SHA:!KRB5-DES-CBC3-SHA"
    # deployType: daemonset
    # nodeSelectorEnabled: "true"
    nodeSelector:
      ka: external
    metrics:
      enabled: true
      service:
        annotations:
          prometheus.io/port: "10254"
          prometheus.io/scrape: "true"
    podAnnotations:
      prometheus.io/port: "10254"
      prometheus.io/scrape: "true"

##### INGRESS INT #####
- name: ingress-int-release
  namespace: ingress-int
  chart: ngingress/ingress-nginx
  version: 4.0.16
  values:
    ### REQUIRED ###
    - controller:
      admissionWebhooks:
        enabled: false
      ingressClassResource:
        name: nginxint
      kind: DaemonSet
      hostNetwork: true
      defaultTls:
        crt: {{ env "PROD_CERT" }}
        key: {{ env "PROD_KEY" }}
    ### OPTIONAL ###
    # sysctls:
    #   net.core.somaxconn: "65535"
    #   net.ipv4.netfilter.ip_conntrack_max: "1048576"
    #   net.netfilter.nf_conntrack_max: "1048576"
    #   fs.file-max: "2097152"
    #   net.ipv4.tcp_max_syn_backlog: "65535"
    #   net.ipv4.tcp_syncookies: "1"
    #   net.ipv4.tcp_timestamps: "1"
    #   net.ipv4.ip_local_port_range: "1024 65535"
    config:
      ssl-redirect: "false"
      log-format-upstream: '{"timestamp": "%msec", "origin": "nginx", "path": "$request_uri",
"time_request": $request_time, "query_string": "$query_string", "time_backend_connect":
"$upstream_connect_time", "time_backend_response": "$upstream_response_time", "host": "$host", "method":

```

```
"$request_method", "user_agent": "$http_user_agent", "http_status_code": $status, "bytes_sent":
$bytes_sent, "scheme": "$scheme", "request_proto": "$server_protocol", "connection_requests":
connection_requests, "upstream_addr": "$upstream_addr", "upstream_http_server":
"$upstream_http_server", "x_forwarded_for": "$http_x_forwarded_for", "x_serial_number":
"$http_x_serial_number", "cid": "$http_x_correlation_id", "hwid": "$http_x_hwid",
"x_domain_code": "$http_x_domain_code", "client": "$remote_addr"}'
  log-format-escape-json: "true"
  gzip-level: "2"
  proxy-buffering: "on"
  allow-backend-server-header: "true"
  enable-modsecurity: "false"
  skip-access-log-urls: "/healthz"
  worker-processes: "auto"
  proxy-stream-timeout: "600s"
  client-header-timeout: "60"
  client-body-timeout: "60"
  max-worker-connections: "65535"
  keep-alive: "75"
  limit-rate: "0"
  limit-rate-after: "0"
  limit-rps: "10"
  load-balance: "round_robin"
  ssl-protocols: "TLSv1 TLSv1.1 TLSv1.2"
  ssl-ciphers: "ECDHE-RSA-AES128-GCM-SHA256:ECDSA-AES128-GCM-SHA256:ECDSA-AES256-GCM-
SHA384:ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA256:kEDH+AESGCM:
ECDHE-RSA-AES128-SHA256:ECDSA-AES128-SHA256:ECDSA-AES128-SHA:ECDSA-AES128-SHA:ECDSA-AES128-
SHA:ECDSA-AES256-SHA384:ECDSA-AES256-SHA384:ECDSA-AES256-SHA:ECDSA-AES256-SHA:DHE-RSA-AES128-
SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA:DHE-RSA-AES256-
SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:AES:
CAMELLIA:DES-CBC3-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!MD5:!PSK:!aECDH:!EDH-DSS-DES-CBC3-SHA:!EDH-RSA-DES-CBC3-
SHA:!KRB5-DES-CBC3-SHA"
  # deployType: daemonset
  # nodeSelectorEnabled: "true"
  nodeSelector:
    ka: internal
  metrics:
    enabled: true
  service:
    annotations:
      prometheus.io/port: "10254"
      prometheus.io/scrape: "true"
  podAnnotations:
    prometheus.io/port: "10254"
    prometheus.io/scrape: "true"
  defaultBackend:
    enabled: true
  nodeSelector:
    ka: internal

##### KEEPALIVED #####
- name: keepalived-release
  namespace: ingress-int
  chart: chartmuseum/keepalived-chart
  version: 0.22.1
  values:
    ### REQUIRED ###
  - Ips:
    - 192.168.14.136
  servers:
    - node4-str-sand#enol#192.168.14.117
    - node2-str-sand#enol#192.168.14.32
    - node3-str-sand#enol#192.168.14.36
  nodeSelector:
    ka: internal

##### CEPH-CSI-RBD #####
- name: ceph-csi-rbd
```

```
namespace: ceph-csi-rbd
chart: chartmuseum/ceph-csi-rbd
installed: true
version: 0.3.0
values:
- ceph-csi-rbd:
  csiConfig:
    - clusterID: {{ env "ceph_fsid" }}
      monitors:
        - {{ env "ceph_mon1" }}
        - {{ env "ceph_mon2" }}
        - {{ env "ceph_mon3" }}

- secret:
  userKey: {{ env "rbdkube_key" }}

- storageClass:
  clusterID: {{ env "ceph_fsid" }}
  defaultClass: true

##### CEPH-CSI-CEPHFS #####
- name: ceph-csi-cephfs
  namespace: ceph-csi-cephfs
  chart: chartmuseum/ceph-csi-cephfs
  version: 0.3.0
  values:
- ceph-csi-cephfs:
  csiConfig:
    - clusterID: {{ env "ceph_fsid" }}
      monitors:
        - '{{ env "rgwServer2" }}:6789'
        - '{{ env "rgwServer3" }}:6789'
        - '{{ env "rgwServer4" }}:6789'

- secret:
  adminKey: {{ env "cephfs_key" }}

- storageClass:
  clusterID: {{ env "ceph_fsid" }}
  defaultClass: false
```

## 6. Установка компонентов STREAMER

Чтобы установить компоненты, сделайте следующее:

1. Настройте kubeconfig для доступа runner к кластеру:
  - a. Перейдите в раздел Settings->CI/CD->Secret variables.
  - b. Создайте переменную kube\_config и в её значении пропишите содержимое файла .kube/config с любой мастер-ноды kubernetes-кластера, зашифрованное кодировкой base64.
2. Создайте в DRE Advanced Media Platform META DATA SERVER (далее - MDS) (пример: <http://uishah.srv2.test.gs-labs.tv/new/account-manager/services>) сервис с названием «streamer». Инструкция по созданию в руководствах MDS.
3. Измените в values каждого чарта необходимые параметры. Ссылка на пример (файл testing.yaml) предоставляется по запросу заказчика.
4. Установите через GitLab CI/CD все компоненты STREAMER с предоставленными стабильными версиями.
5. В переменных окружения GitLab укажите AWS\_ACCESS\_KEY\_ID и AWS\_SECRET\_ACCESS\_KEY пользователя STREAMER, полученные при установке Seph кластера в главном deployment репозитории.
6. В переменной окружения PLAYOUT\_ADDITIONAL\_SEGMENT\_URL задайте ссылку на дополнительный сегмент для виртуальных каналов (Playout-каналы), который позволяет корректно проигрывать поток при разрывах между контентом.
7. При необходимости, в переменной окружения PLAYOUT\_DISABLE\_ADDITIONAL\_SEGMENT\_AGENTS задайте через запятую типы клиентов (User-Agent), у которых будет отсутствовать дополнительный сегмент, необходимый для корректного вещания Playout-каналов. Значение по умолчанию отсутствует.
8. При необходимости, в переменной окружения PLAYOUT\_DISABLE\_DISCONTINUITY\_AGENTS задайте через запятую типы клиентов (User-Agent), у которых будут отсутствовать тэги *discontinuity*. Значение по умолчанию отсутствует.
9. Запустите джоб для развёртывания.

## 7. Развёртывание Ceph

### 7.1. База

1. Создайте форк (ссылка предоставляется по запросу заказчика).
2. В передаваемом репозитории в файле README.md. В файле описаны основные джобы для работы с Ceph кластером, а также указаны переменные окружения, которые нужно указывать для того или иного действия.
3. Измените в all.yml на нужные значения:
  - a. monitor\_interface
  - b. public\_network
  - c. cluster\_network
  - d. radosgw\_interface
4. Измените в inventory\_hosts:
  - a. k8s-ceph-node-1 ansible\_host=192.168.15.209 ip=192.168.15.209 ... (Имя k8s-ceph-node-1 должно совпадать с именем ноды)
  - b. Указать ноды, которые будут мониторами [mons]
  - c. Указать ноды со списком дисков, на которых будут развёртываться osd [osds]
  - d. Указать, на какой ноде будет установлена grafana [grafana-server]
  - e. Указать ноды на которых будет установлены rgw клиенты [rgws]
  - f. Указать ноды на которых будет установлены mds клиенты [mdss] (Необходимо для CephFS)

Измените в group\_vars в файлах mdss.yml, mgrs.yml, mons.yml, osds.yml, rgws.yml MemoryLimit в соответствии с указаниями ниже:

1. Для mds необходимо выставить значение 4G. В тестовой(не нагруженной) среде можно выставить значение 2G. Минимально возможное значение для тестовой среды 1G.
2. Для mgrs необходимо выставить значение 4G. В тестовой(не нагруженной) среде можно выставить значение 2G. Минимально возможное значение для тестовой среды 1G.
3. Для mons необходимо выставить значение 8G и выше(при лишней оперативной памяти, это значение лучше увеличить до 12/16G). В тестовой(не нагруженной) среде можно установить 4G. В большинстве случаев потребление этого демона будет меньше чем даже 4G, но выставлять значение опираясь на статистику потребления крайне не рекомендуется, т.к. при восстановлении или интенсивной записи /удалении/чтении потребление памяти сильно возрастает, и при нехватке работа будет некорректной.
4. Для osds необходимо выставить значение 8G и выше, в случае если диск больше чем 4ТБ, желательно использовать формулу: 2 гб памяти на 1 ТБ, в случае если диски меньше 4ТБ и не являются SSD, минимально возможное значение для продакшена 5G(для SSD 7G), крайне не рекомендуется в продакшене указывать значение меньше. В тестовой зоне(не нагруженной) можно установить лимиты 4G(хотя рекомендуется всё же 5G для корректной работы, даже если диски меньше 1ТБ), в совсем ограниченных тестовых зонах, надо изменять параметры цефа и тогда можно установить 2G, но такая система не будет держать нагрузки и работать полностью корректно в определённых ситуациях.
5. Для rgws необходимо значение 8G(при лишней оперативной памяти, это значение лучше увеличить до 12/16G). В тестовой(не нагруженной) зоне можно установить лимиты 4G.

Так же стоит учесть, что чем выше значение указанное для MemoryLimit тем лучше, особенно это важно для osds, mons и rgws. Так же возможно запустить кластер без создания этих конфигурационных файлов и под нагрузкой определить пиковые и средние значения, что бы потом корректно указать

лимиты, но стоит понимать, что в определённых ситуациях демоны ceph могут использовать всю доступную оперативную память в кластере(и удерживать её вплоть до перезапуска демонов) и тогда kubernetes будет работать некорректно.

## 7.2. Переменные окружения в GitLab Variables

1. Укажите `id_rsa`, через который можно подключиться к каждой ноде кластера, т.к. данный ключ используется для развёртывания кластера.
2. Укажите переменные окружения для разворачивания:
  - a. `grafana_admin_user`,
  - b. `grafana_admin_password`,
  - c. `dashboard_admin_user`,
  - d. `dashboard_admin_password`,
  - e. `system_access_key`,
  - f. `system_secret_key`. Генерация: `system_access_key -> cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 20 | head -n 1; system_secret_key -> cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 40 | head -n 1`

## 7.3. Развёртывание кластера

1. Перейдите в `ci/cd pipelines` и запустите джоб `deploy_cluster`
2. После выполнение джоб создайте пользователя и получите ключи:  
`radosgw-admin user create --uid=streamer --display-name="Streamer"`
3. Скопируйте и сохраните:  
`"access_key": "PG8329ZDOVUK5DB1FTLS",`  
`"secret_key": "xUBqz2Ye3ex4yX7gcuGzFJWs3osAxDAvi93cZCHJ"`
4. Выполните следующие команды:
  - `ceph osd set-require-min-compat-client luminous`
  - `ceph balancer mode upmap`
  - `ceph osd pool set russia-moscow.rgw.buckets.data target_size_ratio 0.90` (выполнить данную команду после того как произойдёт какая либо запись в s3, например: кечапа, вод контента или рекламы)

При возникновении предупреждения `"mons are allowing insecure global_id reclaim / AUTH_INSECURE_GLOBAL_ID_RECLAIM_ALLOWED"`, выполнить следующую команду: `ceph config set mon mon_warn_on_insecure_global_id_reclaim_allowed false`. Данное предупреждение не является критичным и можно не выполнять данную команду, но предупреждение останется в этом случае.

## 7.4. Создание пользователя и получение ключей

1. После развёртывания, проверьте работоспособность всех компонентов и создайте пользователя `streamer`:  
`radosgw-admin user create --uid=streamer --display-name="Streamer"`
2. Скопируйте и сохраните:  
`"access_key": "PG8329ZDOVUK5DB1FTLS",`  
`"secret_key": "xUBqz2Ye3ex4yX7gcuGzFJWs3osAxDAvi93cZCHJ"`

При релизе новой версии STREAMER необходимо узнать об обновлении ceph. Если в ceph есть изменения, обновите файл `all.yml` в соответствии с новым релизом, при этом оставьте свои изменения.

## 7.5. Обновление кластера:

1. Приведите конфигурацию развёртывания в соответствие с переданным релизом.
2. Перейдите в CI/CD pipelines и запустите джоб `update_cluster`.

## 8. Сторонние зависимости

Для работы сервиса должен быть обеспечен доступ к следующим системам:

- Брокер сообщений Pulsar из состава DRE Advanced Media Platform META DATA SERVER (далее - MDS);
- Сервис MDS версии не менее 6.2.1.
- DRE Advanced Media Platform SCRAMBLER.

© ООО "Цифра", 2019-2023

Документация "DRE Advanced Media Platform STREAMER. Руководство по установке" является объектом авторского права. Воспроизведение всего произведения или любой его части воспрещается без письменного разрешения правообладателя.