

DRE Remote Control Service

Руководство администратора

Индекс	2060-Cloud-AG
Конфиденциальность	Публичный - L0
Ревизия	1.0
Статус	Согласован

Содержание

1. Аннотация	3
2. Общее описание	4
3. Компоненты системы	5
3.1. Переменные среды	5
4. Настройка конфигурации	7
5. Пример файла helmfile-yaml	8
6. Мониторинг	10
6.1. Prometheus	10
6.2. Grafana	10
6.2.1. Логирование	10
6.2.2. Активные подключения	16
7. Диагностика	23
8. Базы данных	24

1. Аннотация

Документ предназначен для технических специалистов, занимающихся администрированием сервиса и обладающих навыками работы с компьютером на профессиональном уровне.

2. Общее описание

Сервис DRE Remote Control Service (далее в документе используется условное наименование - облачный сервис Cloud) предназначен для передачи данных и управляющих команд между: внешними системами, удаленными клиентскими приложениями, пользовательским оборудованием системы "Умный дом". Организует передачу информационных сообщений от пользовательского оборудования в клиентские приложения через отправку: sms и push уведомлений, электронных писем; перевод сообщений на поддерживаемые языки. Поддерживает прием и обработку голосовых команд от голосовых помощников, в т. ч. Яндекс.Алиса, Маруся, Салют и аналогичных. Включает в себя программный интерфейс для подключения внешних систем. Для организации канала передачи данных используется сеть Internet.

3. Компоненты системы

Cloud построен на микросервисной архитектуре, функционирует на основе взаимодействия между собой компонентов (сервисов) системы:

- API Gateway - сервис маршрутизации запросов.
- Monitor - следит за активностью подключений сервиса RPC и хранит их состояние.
- Storage - хранит файлы, необходимые для работы контроллера.
- Notifications - отвечает за трансляцию запросов устройства в уведомления, включая переводы на поддерживаемые языки.
- Voice Assistant - принимает запросы от голосового помощника Яндекс.Алиса - авторизует пользователя на управление устройством и транслирует голосовые команды в команды сервиса RPC.
- RPC - отвечает за удалённое управление - поддержка websocket соединения, исполнение запросов мобильных приложений.
- Kron - отвечает за своевременное выполнение асинхронных задач.
- Pechkin - отправляет сообщения по каналам SMS, Email, Push.
- Skill Alice - служит адаптером к API "Умный дом DREHOME&TV" от Яндекс - преобразует и обрабатывает запросы для внешнего сервиса УД Яндекс.
- Skill Marusia - преобразует и обрабатывает запросы для внешнего сервиса УД VK Group.
- Skill Salute - преобразует и обрабатывает запросы для внешнего сервиса УД Сбер.
- Control Hub - предоставляет высокоуровневое API для выполнения действий над "умными" устройствами и получения их состояния.
- Minio - Minio хранит системные файлы STB (метрики, файлы конфигураций), а также пользовательские файлы (например, фотографии), то есть работает с сервисами "умного дома". Данный сервис предоставляет Amazon S3 совместимое API. Это позволяет заменить minio на более сложное решение при необходимости. Отказоустойчивость minio достигается путем балансировки и алгоритма Erasure Code.
- NATS - брокер сообщений.
- Redis - хранилище данных для Pechkin, Kron, Voice Assistant, Control Hub - key-value кэш с возможностью указания даты истечения. Хранит информацию об активных подключениях для системы мониторинга. Отказоустойчивость реализуется с помощью Redis Sentinel.
- Cameron - сервис для удаленного управления видеорекамерой.

3.1. Переменные среды

Поведение приложений настраивается с помощью переменных сред. Для каждого микросервиса в проекте smart-home-cloud внутреннего git репозитория ООО "Цифра" (ссылка предоставляется по запросу заказчика) предоставляется шаблон развертывания (helm chart) в папке charts.

Файл values.yaml отвечает за настройку этого шаблона и соответственно микросервиса. Все что находится под ключом **env**, описывает переменные среды приложения. Например:

Поведение приложений настраивается с помощью переменных сред. Для каждого микросервиса в проекте внутреннего git репозитория ООО "Цифра" (ссылка предоставляется по запросу заказчика) в папке .helm /SERVICE_NAME предоставляется шаблон развертывания, при этом:

- переменные сред, отвечающие за настройку микросервиса приведены в файле **config.yaml**, который располагается в директории .helm/SERVICE_NAME/templates/**config.yaml**

- значения переменных сред, которые подставляются в файл `config.yaml` задаются в файле **values.yaml**, который располагается в директории `.helm/SERVICE_NAME/values.yaml`

Пример для сервиса cameron .helm/cameron/templates/config.yaml

```
{{- if .Values.cameron.enabled }}
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ template "name" . }}
  labels:
    {{ include "chartLabels" . | indent 4 }}
    component: cameron
data:
  # logging level, available options: trace, debug, info
  LOG_LEVEL: {{ .Values.cameron.log.level | quote }}
  # redis host and port
  REDIS_ADDR: {{ .Values.cameron.redis.addr | quote }}
  # database number (0-16) to be selected after connecting to the server
  REDIS_DB: {{ .Values.cameron.redis.db | quote }}
  # message broker address
  NATS_ADDR: {{ .Values.cameron.nats.addr | quote }}
  # is an address for server that encoding video stream
  SCRAMBLER_ADDR: {{ .Values.cameron.scrambler.addr | quote }}
  # sets video encryption mode, one of "wv_dash", "gs_aes", "afp_saes", "plain_hls", "plain_dash"
  # plain_hls - not encrypted at all hls stream, gs_aes - default encryption for our mobile apps
  SCRAMBLER_ENCRYPTION_PROFILE: {{ .Values.cameron.scrambler.encryptionProfile | quote }}

{{- end }}
```

Из этого файла можно узнать какие конфигурационные параметры доступны и как они используются.

Так же, благодаря тому, что мы храним их в git репозитории, можно смотреть какие произошли изменения в новой версии одним из способов:

1. Командой `git diff <tag>..HEAD`

Пример

```
git diff 2.3.0.1..HEAD
```

2. Меню Compare на gitlab (ссылка предоставляется по запросу заказчика).

4. Настройка конфигурации

Конфигурацию сервисов по умолчанию можно посмотреть в файле *default.yaml.gotmpl* проекта smart-home-cloud внутреннего git репозитория ООО "Цифра" (ссылка предоставляется по запросу заказчика).

5. Пример файла helmfile-yaml

Пример helmfile-yaml

```
---
releases:
- name: api-gateway
  namespace: {{ .Values.namespace }}
  chart: chartmuseum/smh-api-gateway
  version: {{ .Values | getOrNil "versions.apiGateway" }}
  installed: {{ .Values.apiGateway_val.enabled }}
  values:
    - {{ .Values.apiGateway_val | toYaml | nindent 8 }}

- name: rpc
  namespace: {{ .Values.namespace }}
  chart: chartmuseum/rpc
  version: {{ .Values | getOrNil "versions.rpc" }}
  installed: {{ .Values.rpc_val.enabled }}
  values:
    - {{ .Values.rpc_val | toYaml | nindent 8 }}

- name: control-hub
  namespace: {{ .Values.namespace }}
  chart: chartmuseum/control-hub
  version: {{ .Values.versions.controlHub }}
  installed: {{ .Values.controlHub_val.enabled }}
  values:
    - {{ .Values.controlHub_val | toYaml | nindent 8 }}

- name: skill-alice
  namespace: {{ .Values.namespace }}
  chart: chartmuseum/skill-alice
  version: {{ .Values.versions.skillAlice }}
  installed: {{ .Values.skillAlice_val.enabled }}
  values:
    - {{ .Values.skillAlice_val | toYaml | nindent 8 }}

- name: skill-marusya
  namespace: {{ .Values.namespace }}
  chart: chartmuseum/skill-alice
  version: {{ .Values.versions.skillAlice }}
  installed: {{ .Values.skillMarusya_val.enabled }}
  values:
    - {{ .Values.skillMarusya_val | toYaml | nindent 8 }}

- name: skill-salute
  namespace: {{ .Values.namespace }}
  chart: chartmuseum/skill-salute
  version: {{ .Values.versions.skillSalute }}
  installed: {{ .Values.skillSalute_val.enabled }}
  values:
    - {{ .Values.skillSalute_val | toYaml | nindent 8 }}

- name: cameron
  namespace: {{ .Values.namespace }}
  chart: chartmuseum/cameron
  version: {{ .Values.versions.cameron }}
  installed: {{ .Values.cameron_val.enabled }}
  values:
    - {{ .Values.cameron_val | toYaml | nindent 8 }}

- name: geocache
  namespace: {{ .Values.namespace }}
```



```
chart: chartmuseum/geocache
version: {{ .Values.versions.geocache }}
installed: {{ .Values.geocache_val.enabled }}
values:
  - {{ .Values.geocache_val | toYaml | nindent 8 }}

- name: voice-assistant
  namespace: {{ .Values.namespace }}
  chart: chartmuseum/voice-assistant
  version: {{ .Values | getOrNil "versions.voiceAssistant" }}
  installed: {{ .Values.voiceAssistant_val.enabled }}
  values:
    - {{ .Values.voiceAssistant_val | toYaml | nindent 8 }}

- name: pechkin
  namespace: {{ .Values.namespace }}
  chart: chartmuseum/pechkin
  version: {{ .Values.versions.pechkin }}
  installed: {{ .Values.pechkin_val.enabled }}
  values:
    - {{ .Values.pechkin_val | toYaml | nindent 8 }}

- name: kron
  namespace: {{ .Values.namespace }}
  chart: chartmuseum/kron
  version: {{ .Values.versions.kron }}
  installed: {{ .Values.kron_val.enabled }}
  labels:
    app: kron
  values:
    - {{ .Values.kron_val | toYaml | nindent 8 }}

- name: notifications
  namespace: {{ .Values.namespace }}
  chart: chartmuseum/notifications
  version: {{ .Values | getOrNil "versions.notifications" }}
  installed: {{ .Values.notifications_val.enabled }}
  values:
    - {{ .Values.notifications_val | toYaml | nindent 8 }}

- name: storage
  namespace: {{ .Values.namespace }}
  chart: chartmuseum/storage
  version: {{ .Values | getOrNil "versions.storage" }}
  installed: {{ .Values.storage_val.enabled }}
  values:
    - {{ .Values.storage_val | toYaml | nindent 8 }}

# it's a bit dirty now, but we plan to use helm instead helmfile, so it should be fixed
repositories:
  - name: chartmuseum
    url: https://chartmuseum.gs-labs.tv/svc-dep/
```

6. Мониторинг

Cloud не имеет пользовательского интерфейса, поэтому наблюдение происходит через внешние инструменты Prometheus и Grafana.

Инструменты Prometheus и Grafana, упоминаемые в данной главе, являются сторонними по отношению к Cloud продуктами. Описание их структуры, принципов действия, процедур установки и т.п. выходит за рамки настоящего документа.

6.1. Prometheus

Cloud не имеет особых метрик для Prometheus, поэтому сбор этих метрик настраивать не требуется. Однако, Kubernetes компоненты предоставляют большой набор метрик, который можно использовать для мониторинга, например, метрики nginx ingress.

6.2. Grafana

Grafana выполняет несколько задач в Cloud:

1. Отображение Prometheus метрик
2. Отображение активных соединений приставок и приложений (пп. 5.2.2)
3. Отображение логов компонентов

6.2.1. Логирование

Лог-файлы компонентов системы могут быть получены через стандартный механизм доступа к логам в среде kubernetes при помощи команды `kubectl logs <pod name>`.

Логи компонентов отображаются в Grafana путём подключения как DataSource системы Grafana Loki или Elasticsearch.

Настройка сбора логов остаётся на усмотрение администратора, но т.к. на тестовых зонах используется Grafana Loki ввиду его простоты, то ниже представлена конфигурация дашборда для Grafana Loki.

Для корректной работы переменных должен быть установлен prometheus. Продвинутая работа с Loki доступна из меню Explore в Grafana

logs.json

```
{
  "annotations": {
    "list": [
      {
        "$$hashKey": "object:75",
        "builtIn": 1,
        "datasource": "-- Grafana --",
        "enable": true,
        "hide": true,
        "iconColor": "rgba(0, 211, 255, 1)",
        "name": "Annotations & Alerts",
        "target": {
          "limit": 100,

```

```

        "matchAny": false,
        "tags": [],
        "type": "dashboard"
    },
    "type": "dashboard"
}
]
},
"description": "Loki logs panel with prometheus variables ",
"editable": true,
"gnetId": 12019,
"graphTooltip": 1,
"id": 31,
"iteration": 1654083794606,
"links": [],
"panels": [
    {
        "datasource": "Loki",
        "fieldConfig": {
            "defaults": {
                "color": {
                    "mode": "palette-classic"
                },
                "custom": {
                    "axisLabel": "",
                    "axisPlacement": "hidden",
                    "barAlignment": 0,
                    "drawStyle": "bars",
                    "fillOpacity": 59,
                    "gradientMode": "none",
                    "hideFrom": {
                        "legend": false,
                        "tooltip": false,
                        "viz": false
                    },
                    "lineInterpolation": "linear",
                    "lineWidth": 2,
                    "pointSize": 5,
                    "scaleDistribution": {
                        "type": "linear"
                    },
                    "showPoints": "never",
                    "spanNulls": true,
                    "stacking": {
                        "group": "A",
                        "mode": "none"
                    },
                    "thresholdsStyle": {
                        "mode": "off"
                    }
                },
                "links": [],
                "mappings": [],
                "thresholds": {
                    "mode": "absolute",
                    "steps": [
                        {
                            "color": "green",
                            "value": null
                        },
                        {
                            "color": "red",
                            "value": 80
                        }
                    ]
                }
            },
            "unit": "short"
        },
    },

```

```

    "overrides": [
      {
        "matcher": {
          "id": "byName",
          "options": "{}"
        },
        "properties": [
          {
            "id": "displayName",
            "value": "entries"
          }
        ]
      }
    ]
  },
  "gridPos": {
    "h": 4,
    "w": 24,
    "x": 0,
    "y": 0
  },
  "id": 6,
  "options": {
    "legend": {
      "calcs": [],
      "displayMode": "hidden",
      "placement": "bottom"
    },
    "tooltip": {
      "mode": "multi"
    }
  },
  "pluginVersion": "8.1.5",
  "targets": [
    {
      "expr": "sum(count_over_time({namespace=\"$namespace\", app=~\"$app.*\"} |~ ` $search` [500ms]))",
      "instant": false,
      "range": true,
      "refId": "A"
    }
  ],
  "timeFrom": null,
  "timeShift": null,
  "type": "timeseries"
},
{
  "datasource": "Loki",
  "gridPos": {
    "h": 25,
    "w": 24,
    "x": 0,
    "y": 4
  },
  "id": 2,
  "maxDataPoints": "",
  "options": {
    "dedupStrategy": "none",
    "enableLogDetails": true,
    "prettifyLogMessage": false,
    "showCommonLabels": false,
    "showLabels": false,
    "showTime": true,
    "sortOrder": "Descending",
    "wrapLogMessage": true
  },
  "targets": [
    {
      "expr": "{namespace=\"$namespace\", app=~\"$app.*\"} |~ ` $search`",

```

```

        "refId": "A"
    }
],
"timeFrom": null,
"timeShift": null,
"title": "Logs Panel",
"type": "logs"
},
{
  "datasource": null,
  "gridPos": {
    "h": 3,
    "w": 24,
    "x": 0,
    "y": 29
  },
  "id": 4,
  "options": {
    "content": "",
    "mode": "html"
  },
  "pluginVersion": "8.1.5",
  "timeFrom": null,
  "timeShift": null,
  "transparent": true,
  "type": "text"
}
],
"refresh": false,
"schemaVersion": 30,
"style": "dark",
"tags": [],
"templating": {
  "list": [
    {
      "allValue": null,
      "current": {
        "selected": false,
        "text": "smart-home-cloud-dev",
        "value": "smart-home-cloud-dev"
      },
      "datasource": "Prometheus",
      "definition": "label_values(kube_pod_info, namespace)",
      "description": null,
      "error": null,
      "hide": 0,
      "includeAll": false,
      "label": null,
      "multi": false,
      "name": "namespace",
      "options": [],
      "query": {
        "query": "label_values(kube_pod_info, namespace)",
        "refId": "Prometheus-namespace-Variable-Query"
      },
      "refresh": 1,
      "regex": "/smart-home-cloud|ingress/",
      "skipUrlSync": false,
      "sort": 0,
      "tagValuesQuery": "",
      "tagsQuery": "",
      "type": "query",
      "useTags": false
    },
    {
      "allValue": ".*",
      "current": {
        "selected": false,

```

```

        "text": "All",
        "value": "$__all"
    },
    "datasource": "Prometheus",
    "definition": "label_values(namespace_workload_pod:kube_pod_owner:relabel{namespace=~\"$namespace\"},
workload)",
    "description": null,
    "error": null,
    "hide": 0,
    "includeAll": true,
    "label": null,
    "multi": true,
    "name": "app",
    "options": [],
    "query": {
        "query": "label_values(namespace_workload_pod:kube_pod_owner:relabel{namespace=~\"$namespace\"},
workload)",
        "refId": "StandardVariableQuery"
    },
    "refresh": 1,
    "regex": "",
    "skipUrlSync": false,
    "sort": 0,
    "tagValuesQuery": "",
    "tagsQuery": "",
    "type": "query",
    "useTags": false
},
{
    "current": {
        "selected": false,
        "text": "",
        "value": ""
    },
    "description": null,
    "error": null,
    "hide": 0,
    "label": null,
    "name": "search",
    "options": [
        {
            "selected": true,
            "text": "",
            "value": ""
        }
    ],
    "query": "",
    "skipUrlSync": false,
    "type": "textbox"
}
]
},
"time": {
    "from": "now-30m",
    "to": "now"
},
"timepicker": {
    "refresh_intervals": [
        "5s",
        "10s",
        "30s",
        "1m",
        "5m",
        "15m",
        "30m",
        "1h",
        "2h",
        "1d"
    ]
}

```

```
]
},
"timezone": "",
"title": "Loki Dashboard",
"uid": "liz0yRCZz",
"version": 46
}
```

6.2.2. Активные подключения

Список активных подключений настраивается в Grafana при помощи DataSource PostgreSQL, которые подключается к БД сервиса monitor.

connections.json

```
{
  "annotations": {
    "list": [
      {
        "builtIn": 1,
        "datasource": "-- Grafana --",
        "enable": true,
        "hide": true,
        "iconColor": "rgba(0, 211, 255, 1)",
        "name": "Annotations & Alerts",
        "type": "dashboard"
      }
    ]
  },
  "editable": true,
  "gnetId": null,
  "graphTooltip": 0,
  "id": 1,
  "iteration": 1594219261043,
  "links": [],
  "panels": [
    {
      "columns": [],
      "datasource": "$namespace",
      "fieldConfig": {
        "defaults": {
          "custom": {}
        },
        "overrides": []
      },
      "fontSize": "100%",
      "gridPos": {
        "h": 13,
        "w": 24,
        "x": 0,
        "y": 0
      },
      "id": 8,
      "pageSize": null,
      "showHeader": true,
      "sort": {
        "col": 0,
        "desc": true
      },
      "styles": [
        {
          "alias": "Expires at",
          "align": "auto",
          "dateFormat": "YYYY-MM-DD HH:mm:ss",
          "pattern": "time",
          "type": "date"
        },
        {
          "alias": "Role",
          "align": "auto",
          "colorMode": null,
          "colors": [
            "rgba(245, 54, 54, 0.9)",

```



```

        "rgba(237, 129, 40, 0.89)",
        "rgba(50, 172, 45, 0.97)"
    ],
    "dateFormat": "YYYY-MM-DD HH:mm:ss",
    "decimals": 2,
    "mappingType": 1,
    "pattern": "role",
    "preserveFormat": false,
    "rangeMaps": [],
    "thresholds": [
        ""
    ],
    ],
    "type": "string",
    "unit": "short",
    "valueMaps": []
},
{
    "alias": "Controller ID",
    "align": "auto",
    "colorMode": null,
    "colors": [
        "rgba(245, 54, 54, 0.9)",
        "rgba(237, 129, 40, 0.89)",
        "rgba(50, 172, 45, 0.97)"
    ],
    "dateFormat": "YYYY-MM-DD HH:mm:ss",
    "decimals": 2,
    "mappingType": 1,
    "pattern": "controller_id",
    "thresholds": [],
    "type": "string",
    "unit": "short"
},
{
    "alias": "IP",
    "align": "auto",
    "colorMode": null,
    "colors": [
        "rgba(245, 54, 54, 0.9)",
        "rgba(237, 129, 40, 0.89)",
        "rgba(50, 172, 45, 0.97)"
    ],
    "dateFormat": "YYYY-MM-DD HH:mm:ss",
    "decimals": 2,
    "mappingType": 1,
    "pattern": "ip",
    "thresholds": [],
    "type": "string",
    "unit": "short"
},
{
    "alias": "Client ID",
    "align": "auto",
    "colorMode": null,
    "colors": [
        "rgba(245, 54, 54, 0.9)",
        "rgba(237, 129, 40, 0.89)",
        "rgba(50, 172, 45, 0.97)"
    ],
    "dateFormat": "YYYY-MM-DD HH:mm:ss",
    "decimals": 2,
    "mappingType": 1,
    "pattern": "client_id",
    "thresholds": [],
    "type": "string",
    "unit": "short"
},
{

```



```

    "hide": 0,
    "includeAll": false,
    "label": null,
    "multi": false,
    "name": "namespace",
    "options": [],
    "query": "postgres",
    "refresh": 1,
    "regex": "",
    "skipUrlSync": false,
    "type": "datasource"
  },
  {
    "allValue": null,
    "current": {
      "tags": [],
      "text": "controller + remote",
      "value": [
        "controller",
        "remote"
      ]
    },
    "hide": 0,
    "includeAll": false,
    "label": null,
    "multi": true,
    "name": "role",
    "options": [
      {
        "selected": true,
        "text": "controller",
        "value": "controller"
      },
      {
        "selected": true,
        "text": "remote",
        "value": "remote"
      }
    ],
    "query": "controller,remote",
    "skipUrlSync": false,
    "type": "custom"
  },
  {
    "current": {
      "text": "",
      "value": ""
    },
    "hide": 0,
    "label": null,
    "name": "controllerId",
    "options": [
      {
        "text": "",
        "value": ""
      }
    ],
    "query": "",
    "skipUrlSync": false,
    "type": "textbox"
  },
  {
    "current": {
      "text": "",
      "value": ""
    },
    "hide": 0,
    "label": null,

```

```

    "name": "ip",
    "options": [
      {
        "text": "",
        "value": ""
      }
    ],
    "query": "",
    "skipUrlSync": false,
    "type": "textbox"
  },
  {
    "current": {
      "text": "",
      "value": ""
    },
    "hide": 0,
    "label": null,
    "name": "clientId",
    "options": [
      {
        "text": "",
        "value": ""
      }
    ],
    "query": "",
    "skipUrlSync": false,
    "type": "textbox"
  },
  {
    "current": {
      "text": "",
      "value": ""
    },
    "hide": 0,
    "label": null,
    "name": "domainId",
    "options": [
      {
        "text": "",
        "value": ""
      }
    ],
    "query": "",
    "skipUrlSync": false,
    "type": "textbox"
  }
]
},
"time": {
  "from": "now-6h",
  "to": "now"
},
"timepicker": {
  "refresh_intervals": [
    "10s",
    "30s",
    "1m",
    "5m",
    "15m",
    "30m",
    "1h",
    "2h",
    "1d"
  ]
},
"timezone": "",
"title": "Active Connections",

```

```
"uid": "5OzsNzLZk",
"version": 6
}
```

График количества подключений по вебсокету в течение интервала времени можно получить, добавив следующий дашборд:

connections-graph.json

```
{
  "id": 10,
  "gridPos": {
    "h": 7,
    "w": 24,
    "x": 0,
    "y": 0
  },
  "type": "timeseries",
  "title": "Time series",
  "pluginVersion": "8.1.5",
  "fieldConfig": {
    "defaults": {
      "custom": {
        "drawStyle": "line",
        "lineInterpolation": "linear",
        "barAlignment": 0,
        "lineWidth": 1,
        "fillOpacity": 39,
        "gradientMode": "none",
        "spanNulls": false,
        "showPoints": "always",
        "pointSize": 5,
        "stacking": {
          "mode": "none",
          "group": "A"
        },
        "axisPlacement": "auto",
        "axisLabel": "",
        "scaleDistribution": {
          "type": "linear"
        },
        "hideFrom": {
          "tooltip": false,
          "viz": false,
          "legend": false
        },
        "thresholdsStyle": {
          "mode": "off"
        },
        "lineStyle": {
          "fill": "solid"
        }
      },
      "color": {
        "mode": "palette-classic"
      },
      "thresholds": {
        "mode": "absolute",
        "steps": [
          {
            "color": "green",
            "value": null
          },
          {
            "color": "red",

```

```
        "value": 80
      }
    ]
  },
  "mappings": [],
  "overrides": [],
},
"options": {
  "tooltip": {
    "mode": "single"
  },
  "legend": {
    "displayMode": "hidden",
    "placement": "bottom",
    "calcs": []
  }
},
"targets": [
  {
    "exemplar": true,
    "expr": "sum(cloud_connected_controllers{namespace=\"$namespace\"})",
    "instant": false,
    "interval": "",
    "legendFormat": "",
    "refId": "A"
  }
],
"datasource": null
}
```

7. Диагностика

Ввиду того, что Cloud - это приложение, разработанное для Kubernetes, диагностика системы происходит путём анализа логов и состояния Kubernetes pods. В ситуациях, когда возникают неожиданные ошибки, необходимо обратиться к разработчику для консультации

8. Базы данных

В продукте Cloud присутствует несколько сервисов, хранящих информацию в базах PostgreSQL. Как правило, каждый сервис владеет отдельной БД и имеет доступ только к ней.

Более подробная информация о базах данных сервисов с конфигурациями дашбордов предоставляется заказчику по запросу.

© ООО "Цифра", 2023

Документация "DRE Remote Control Service. Руководство администратора" является объектом авторского права. Воспроизведение всего произведения или любой его части воспрещается без письменного разрешения правообладателя.