КИБЕРПРОТЕКТ

КИБЕР Бэкап

Версия 19.0



Открытый API

Редакция: 29.10.2026

© ООО «Киберпротект», 2026

ООО «Киберпротект» является правообладателем данного документа.

Все права защищены.

Распространение измененных версий данного руководства, а также переработанных материалов, входящих в данное руководство, запрещено без явного разрешения владельца авторских прав.

ДОКУМЕНТ ПРЕДОСТАВЛЯЕТСЯ «КАК ЕСТЬ». ДОКУМЕНТ НЕ ПРЕДПОЛАГАЕТ ОБЯЗАТЕЛЬСТВ И/ИЛИ ГАРАНТИЙ ПРАВООБЛАДАТЕЛЯ ОТНОСИТЕЛЬНО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, НАСКОЛЬКО ТАКОЕ ОГРАНИЧЕНИЕ ОТВЕТСТВЕННОСТИ ДОПУСКАЕТСЯ ЗАКОНОМ, ВКЛЮЧАЯ, СРЕДИ ПРОЧЕГО, ГАРАНТИИ КОММЕРЧЕСКОЙ ЦЕННОСТИ, УДОВЛЕТВОРИТЕЛЬНОГО КАЧЕСТВА, ПРИГОДНОСТИ ДЛЯ КОНКРЕТНОЙ ЦЕЛИ.

Оглавление

Обзор **1**

Архитектура 1

Модель тенантов 3

Виды интеграции 4

Интеграция с сервисами управления **5**

Об API 5

Политика версионирования и политика End of life 6

О документации 6

Установка окружения Python 6

Запуск оболочки Python и настройка ее сеанса 8

Авторизация 8

Аутентификация на платформе Кибер Бэкап 9

Доступность конечных точек для различных ролей пользователей 12

Запросы и ответы API 14

Параметры запросов API 17

Коды ответа API 20

Проверка кодов ответов API 20

Коды состояния HTTP 22

Сортировка и постраничные ответы 23

Постраничные ответы 23

Сортировка 25

Запросы API, инициирующие длительные операции, и ожидание их окончания 25

API REST-ресурсы и операции с ними 29

Ресурс 29

Получение списка ресурсов 34

Получение подробной информации о ресурсе 35

Группа 39

Создание группы 40

Обновление группы 42

Удаление группы 43

Добавление ресурсов в статическую группу или удаление их из нее 44

Политика 45

Параметры политики защиты 47

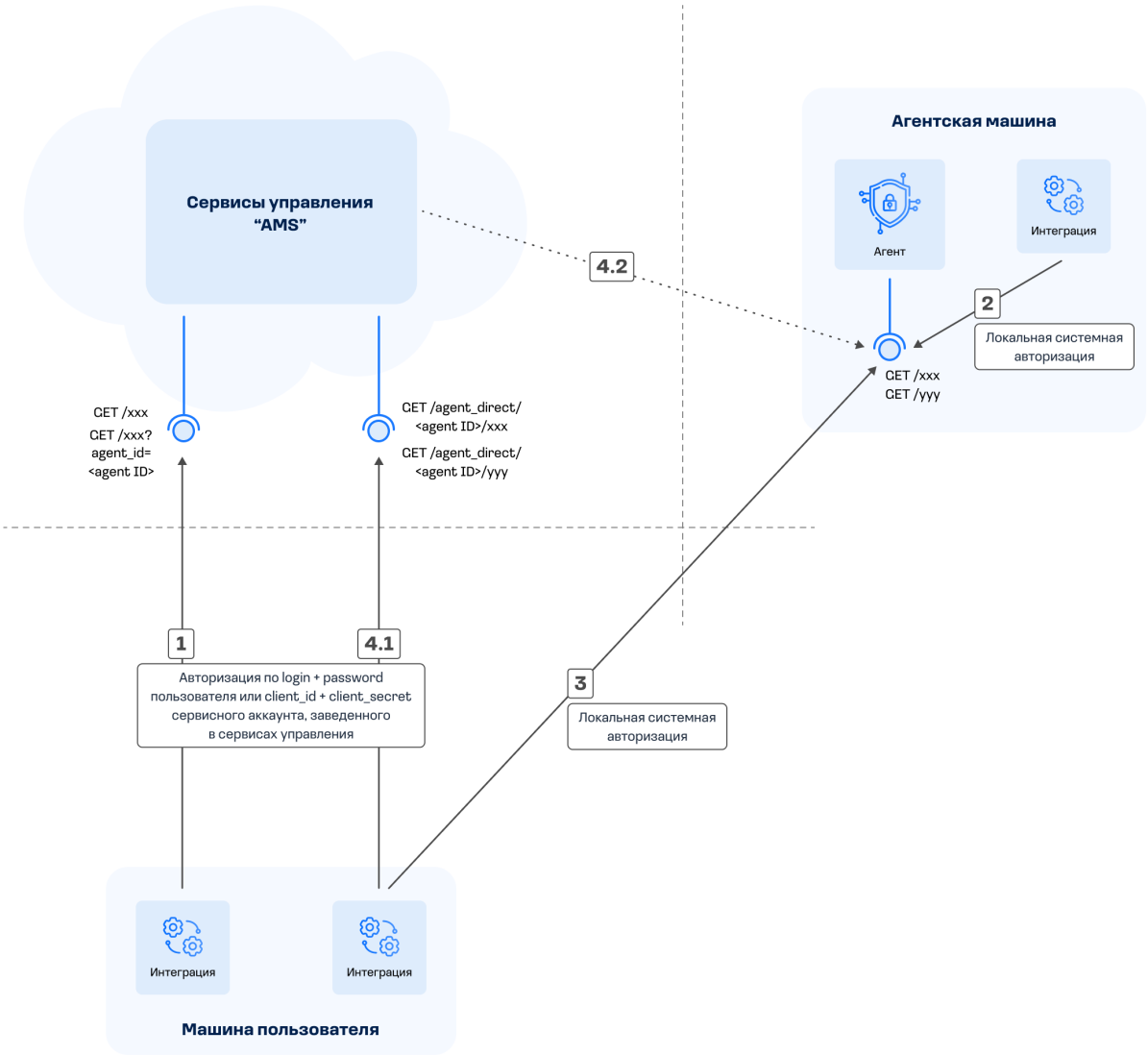
Создание плана резервного копирования 70

Получение перечня планов резервного копирования . . . 74
Получение информации о плане резервного копирования . . . 86
Редактирование параметров плана резервного копирования . . . 91
Включение/выключение плана резервного копирования . . . 93
Удаление плана резервного копирования . . . 94
Применение политики . . . 95
Получение списка применений плана резервного копирования . . . 95
Применение ранее созданного плана резервного копирования к ресурсу . . . 98
Отзыв ранее созданного плана резервного копирования с ресурса . . . 100
Запуск плана резервного копирования . . . 101
Реквизиты для входа . . . 103
Создание реквизитов для входа . . . 105
Получение информации о реквизитах для входа . . . 106
Удаление реквизитов для входа . . . 107
Архив резервных копий . . . 108
Просмотр архивов резервных копий . . . 109
Удаление архива резервных копий . . . 113
Резервная копия . . . 114
Просмотр резервных копий . . . 115
Удаление резервных копий . . . 118
Задача . . . 120
Получение списка задач . . . 121
Получение информации об отдельной задаче . . . 125
Действие . . . 127
Получение списка действий . . . 128
Получение информации о действии . . . 130
Справочники . . . 132
Справочник API . . . 132
Справочник по атрибутам ресурсов . . . 132
Справочник по настройкам политик . . . 132
Список изменений . . . 133
Интеграция с агентами . . . 134
Параметры командной строки для установщика агентов . . . 134
Параметры командной строки для регистрационной утилиты . . . 141

Обзор

Архитектура

Схема взаимодействия через API



Информационные потоки на схеме:

1 — Запросы открытого API к функциям сервисов управления продукта.

2 — Запросы к API агентов от интеграционных сервисов на агентской машине.

3 — Запросы к API агентов от интеграционных сервисов на машине управления.

4.1 — Запросы открытого API к функциям агентов.

4.2 — Запросы к API агентов от сервисов управления продукта.

Назначение открытого API

Открытый API обеспечивает авторизацию (см. *Авторизация*), а также доступ к функциям сервисов управления продукта и к функциям агентов, защищающих рабочие нагрузки (виртуальные или физические машины, приложения и т. п.).

Доступ к функциям сервисов управления позволяет управлять следующими ресурсами:

- *Ресурс*;
- *Политика*;
- *Применение политики*;
- *Реквизиты для входа*;
- *Задача*;
- *Действие*.

Доступ к функциям агентов позволяет выполнять следующие действия:

- Резервное копирование
  - Виртуальных и физических машин;
  - Дисков;
  - Файлов;
  - Приложений;
  - Баз данных;
  - Почтовых ящиков;
  - Конфигураций серверов и отдельных виртуальных машин.
- Восстановление из резервных копий.
- Операции с ленточными хранилищами, архивами, резервными копиями.
- Операции с задачами.
- Операции администрирования.

Примечание

Управление функциями агентов запланировано в следующих версиях документации открытого API.

Модель тенантов

Платформа Кибер Бэкап использует мультитенантную архитектуру для того, чтобы обеспечить распределенное владение для данных, контроль доступа, лицезирование и настройку для организации и ее физических или логических частей.

Ниже перечислены типы тенантов, присутствующие в платформе Кибер Бэкап.

Они связаны отношением родители-дети в следующем порядке:

1. **Customer** (**Клиент**).

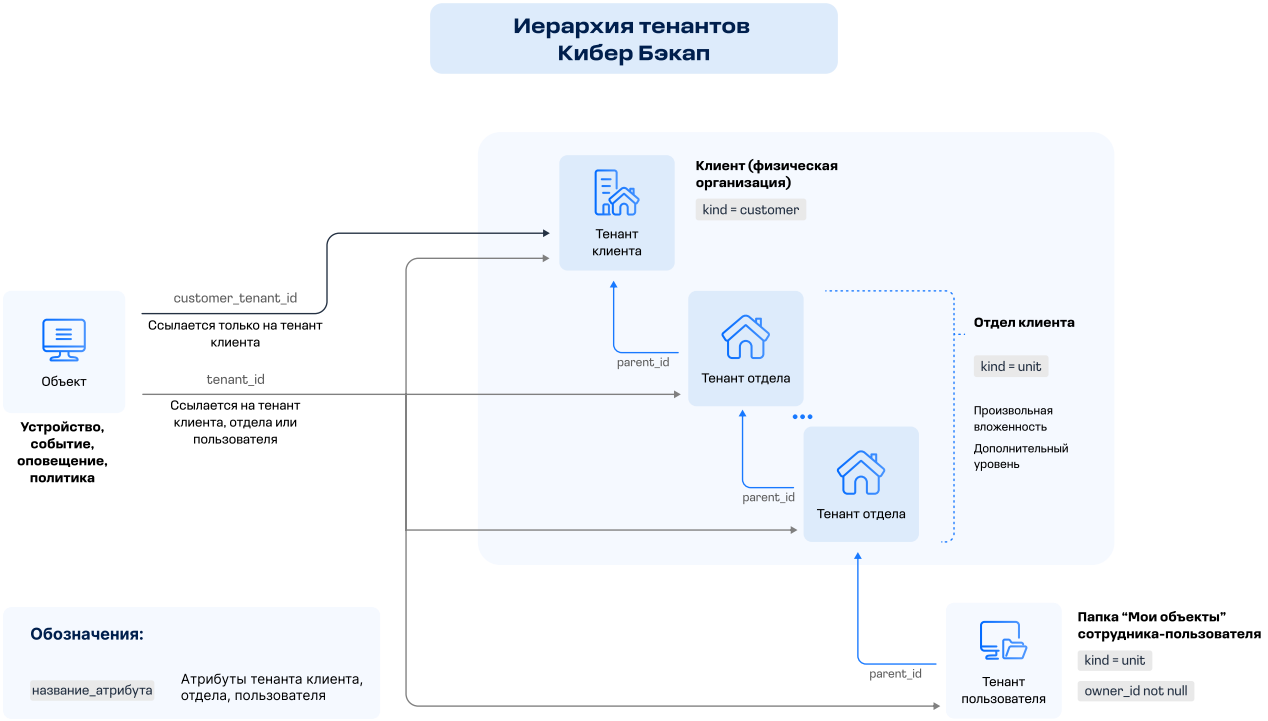
   Клиентская организация, которая приобретает и использует услуги.

2. **Unit** (**Отдел**).

   Логическая единица для управления группами и отделами в рамках клиентской организации.

Пример

Пример иерархии тенантов клиента и отдела.



Виды интеграции

Интеграция с сервисами управления, API сервисов управления

Интеграция с сервисами управления обеспечивает авторизацию (см. *Авторизация*) и доступ к управлению следующими ресурсами:

- *Ресурс*;
- *Политика*;
- *Применение политики*;
- *Реквизиты для входа*;
- *Задача*;
- *Действие*.

Подробнее см. в разделе *Интеграция с сервисами управления*.

Интеграция с агентами

Интеграция с агентами обеспечивает доступ к функциям защиты рабочих нагрузок (виртуальные или физические машины, приложения и т. п.).

Подробнее об интеграции см. в разделе *Интеграция с агентами*.

Интеграция с сервисами управления

Об API

Варианты установки продукта

Кибер Бэкап может быть установлен в следующих вариантах:

- Кибер Бэкап в облаке вендора;
- Кибер Бэкап в приватном облаке;
- Кибер Бэкап в локальном варианте (on-premise).

Коммуникационный протокол

Для API сервисов управления используется протокол HTTP или HTTPS (в зависимости от вида установки продукта):

- Кибер Бэкап Облачный (в облаке вендора) — всегда используется HTTPS;
- Кибер Бэкап — HTTP или HTTPS, доступно конфигурирование заказчиком.

Стиль API

API сервисов управления построен на стиле REST.

Структура URL API

Авторизация

Адрес для авторизации:`(http|https)://<installation address>/idp/token`.

Управление ресурсами

- `(http|https)://<installation address>/api/<api version>/<rest resource>` — URL-адрес управления ресурсами, в котором:
  - `installation address` — адрес, по которому размещён установленный продукт;
  - `api version` — используемая версия API;
  - `rest resource` — название ресурса.

Политика версионирования и политика End of life

Версии API

Номер версии API находится в URL:

```
(http|https)://<installation address>/api/<api version>/<rest resource>
```

Пример:

```
/api/v1/
```

Введение новых версий API

Изменения в API подразделяются на следующие:

- ”минорные” — незначительные изменения, которые не приводят к изменению версии API;
- ”мажорные” — значительные изменения, меняющие версию API.

Поддержка старых версий, политика End of life (EOL)

API сервисов управления поддерживает 3 последних мажорных версии: 1 актуальная и 2 архивных.

О документации

Примеры запросов в руководстве по API даны для curl. Примеры с кодом даны на языке Python (версия 3.7), поскольку это простой, понятный и широко используемый язык.

Для чтения примеров требуются базовые знания Python. Вам будет нетрудно понимать примеры кода, даже если вы не слишком хорошо знакомы с Python.

Хороший способ начать работу с Python — это ознакомиться с руководством по Python.

Установка окружения Python

Установка окружения Python в Windows

1. Скачайте установщик последней версии Python 3.
2. Запустите установщик.
3. Выберите **Add Python 3.x to PATH**.
4. Щелкните **Install Now**.

Произойдет установка Python и pip (утилита для локальной установки и настройки пакетов Python) в вашу пользовательскую папку и добавление папок с исполняемыми файлами Python в ваш пользовательский путь.

5. Запустите командную строку.
6. Установите пакет requests, необходимый для отправки запросов к API:

```
pip install requests
```

Установка окружения Python в Mac OS X

1. Запустите терминал.
2. Установите Python 3 и pip (утилита для локальной установки и настройки пакетов Python).
3. Установите пакет requests, необходимый для отправки запросов к API:

```
pip install requests
```

Установка окружения Python в Linux

1. Запустите терминал.
2. Проверьте, установлен ли у вас уже Python 3:

```
python --version
```

Если Python 3 не установлен, установите его через менеджер пакетов вашего дистрибутива. В зависимости от дистрибутива Linux наименования команд и пакетов, необходимых для установки, отличаются.

- В дистрибутивах Debian и основанных на них, таких как Ubuntu, используйте команду `apt`:

```
sudo apt-get install python3
```

- В дистрибутивах Red Hat и основанных на них используйте команду `yum`:

```
sudo yum install python3
```

- В дистрибутивах SUSE и основанных на них используйте команду `zypper`:

```
sudo zypper install python3
```

3. Установите pip (утилита для локальной установки и настройки пакетов Python).

4. Установите пакет requests, необходимый для отправления запросов к API:

```
pip install requests
```

Запуск оболочки Python и настройка ее сеанса

1. Убедитесь, что настройка окружения Python (см. *О документации*) выполнена корректно.
2. Запустите командную строку.
3. Запустите интерактивную оболочку Python с помощью следующей команды:

```
> python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

   Теперь вы сможете запускать команды Python и фрагменты кода.

4. Загрузите модули, необходимые для запуска команд в этой инструкции:

```
>>> import requests  # used for sending requests to the API
>>> import json      # used for manipulating JSON data
>>> import pprint    # used for formatting the output of JSON objects received in API¬
↪responses
```

Авторизация

Запросы к конечной точке открытого API должны выполняться в соответствии со схемой аутентификации Bearer, если не указано иное. Эта схема требует указания токена в заголовке `Authorization`. Токен — это уникальная зашифрованная строка, генерируемая облачной платформой. Токены устраняют необходимость в передаче учетных данных пользователя с запросами.

Токен выдается конечной точкой `/idp/token` системой управления учетными записями открытого API.

Чтобы сгенерировать токен, ознакомьтесь с разделом *Аутентификация на платформе Кибер Бэкап*.

Пример:

Запрос, отправленный через curl для API, имеет следующую форму:

```
curl -X POST -s <installation address>/api/<api version>/<endpoint> \
    --header "Authorization: Bearer¬
↪8770b34b74f9e4d9424eff50c38182bb4ae7f5596582ae61900b1b6a23e3ec58"
```

Если, например, срок действия токена, указанного в заголовке `Authorization`, истек или ваша учетная запись отключена, API вернет код состояния 401 (см. *Коды состояния HTTP*).

Важно

По соображениям безопасности время истечения срока действия токена установлено равным двум часам. По истечении этого времени API вернет код состояния 401 (см. *Коды состояния HTTP*).

Аутентификация на платформе Кибер Бэкап

Пройдя аутентификацию на платформе Кибер Бэкап с помощью токена доступа, интеграция получит право выполнять операции в тенанте и его субтенантах в соответствии с ролью пользователя, для которого сформирован токен.

Примечание

Перед началом работы запустите оболочку Python и настройте ее сеанс (см. *Запуск оболочки Python и настройка ее сеанса*). В интерактивной оболочке должны быть доступны модули **requests** и **pprint**.

Выполнение аутентификации на платформе Кибер Бэкап

1. Задайте переменные `username`, `password` и `datacenter_url`. В переменной `datacenter_url` укажите URL-адрес установки Кибер Бэкап без пути `/idp/token`:

   ```
   >>> username = '<имя пользователя>'
   >>> password = '<пароль>'
   >>> datacenter_url = '<URL-адрес установки продукта>'.rstrip('/')
   ```

2. Сформируйте тело запроса. Для `grant_type=password` поля `username` и `password` являются обязательными и передаются в теле запроса в формате `application/x-www-form-urlencoded`:

   ```
   >>> token_request_data = {
   ...     'grant_type': 'password',
   ...     'username': username,
   ...     'password': password,
   ... }
   ```

   Если для учетной записи настроена двухфакторная аутентификация TOTP, добавьте актуальный одноразовый код:

```
>>> token_request_data['totp_code'] = '<одноразовый код TOTP>'
```

3. Отправьте POST-запрос на конечную точку `/idp/token`. Модуль `requests` сам установит заголовок `Content-Type: application/x-www-form-urlencoded` при передаче словаря через параметр `data`:

```
>>> response = requests.post(
...     f'{datacenter_url}/idp/token',
...     data=token_request_data,
...     timeout=30,
... )
```

4. Проверьте код состояния и обработайте ошибку до извлечения токена:

```
>>> response.status_code
200
>>> if response.status_code != 200:
...     try:
...         error_info = response.json()
...     except ValueError:
...         error_info = response.text
...     raise RuntimeError(
...         f'Не удалось получить токен '
...         f'(HTTP {response.status_code}): {error_info}'
...     )
```

Код 200 означает, что аутентификация выполнена успешно и платформа выдала токены. При ошибке сервер возвращает JSON-объект с полями `error` и `error_description`. Например, ошибка `totp_required` означает, что в тело запроса нужно добавить `totp_code`.

5. Преобразуйте JSON из тела успешного ответа в словарь и сохраните его в переменной `token_info`:

```
>>> token_info = response.json()
>>> pprint.pprint(token_info)
{'access_token': 'eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...',
 'expires_in': 600,
 'expires_on': 1562910964,
 'id_token': 'eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjI5ZT...',
 'refresh_token': 'OhYc9oko5ej9',
 'token_type': 'bearer'}
```

Поле `refresh_token` является необязательным и может отсутствовать в ответе.

6. Создайте словарь `auth` для заголовка `Authorization` в последующих API-запросах:

```
>>> auth = {'Authorization': 'Bearer ' + token_info['access_token']}
>>> auth
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

   Передавайте этот словарь в каждом запросе к защищенному API. Например:

```
>>> response = requests.get(
...     f'{datacenter_url}/api/v1/tenants',
...     headers=auth,
...     timeout=30,
... )
```

Получение токена с помощью curl

Получить токен можно без запуска оболочки Python. Для этого отправьте POST-запрос на конечную точку `/idp/token` и передайте параметры в формате `application/x-www-form-urlencoded`.

Пример для Linux

```
curl --request POST '<URL-адрес установки продукта>/idp/token' \
  --header 'Content-Type: application/x-www-form-urlencoded' \
  --data-urlencode 'grant_type=password' \
  --data-urlencode 'username=<имя пользователя>' \
  --data-urlencode 'password=<пароль>'
```

Параметр `--data-urlencode` обеспечивает URL-кодирование значений перед их отправкой. Если для учетной записи настроена двухфакторная аутентификация TOTP, добавьте к команде параметр с актуальным одноразовым кодом:

```
--data-urlencode 'totp_code=<одноразовый код TOTP>'
```

Пример для Windows (командная строка cmd)

```
curl --request POST ”<URL-адрес установки продукта>/idp/token” ^
  --header ”Content-Type: application/x-www-form-urlencoded” ^
  --data-urlencode ”grant_type=password” ^
  --data-urlencode ”username=<имя пользователя>” ^
  --data-urlencode ”password=<пароль>”
```

Запрос через curl можно также передать одной строкой, без разбивки на строки.

Предупреждение

При передаче пароля непосредственно в командной строке не рекомендуется использовать пробелы, одинарные и двойные кавычки, обратную косую черту, обратную кавычку, знаки доллара и восклицания, амперсанд, вертикальную черту, точку с запятой, угловые и круглые скобки, знак процента и знак вставки. Командная оболочка может интерпретировать эти символы до запуска `curl`, из-за чего сервер получит измененный пароль. Если такие символы уже входят в пароль, не удаляйте и не заменяйте их: передайте пароль через переменную окружения или файл и обязательно используйте `--data-urlencode`.

Теперь интеграция может обращаться к API Кибер Бэкап в пределах прав, заданных в токене доступа.

Справочные материалы

- Описание конечной точки /idp/token.
- Описание гранта Resource Owner Password Credentials.

Доступность конечных точек для различных ролей пользователей

	Конечная точка	Администратор	Только для чтения	Оператор мониторинга	Аудитор
Токен	Создание токена `POST /idp/token`	Да	Да	Да	Да
Ресурс	Чтение списка устройств `GET /api/v1/resources`	Да	Да	Да	Нет
	Чтение информации о ресурсе `GET /api/v1/resources/<resource_ID>`	Да	Да	Да	Нет
Политика	Создание планов `POST /api/v1/policies`	Да	Нет	Нет	Нет
	Получение списка планов `GET /api/v1/policies`	Да	Да	Да	Нет
	Получение информации о плане `GET /api/v1/policies/<policy_ID>`	Да	Да	Да	Нет
	Редактирование плана `PATCH /api/v1/policies/<policy_ID>`	Да	Нет	Нет	Нет

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

	Конечная точка	Администратор	Только для чтения	Оператор мониторинга	Аудитор
	Включение и отключение плана `PATCH /api/v1/policy_applications/enabled`	Да	Нет	Нет	Нет
	Удаление плана `DELETE /api/v1/policies/<policy_ID>`	Да	Нет	Нет	Нет
Применение политики	Применение плана `POST /api/v1/policy_applications`	Да	Нет	Нет	Нет
	Отзыв плана `DELETE /api/v1/policy_applications`	Да	Нет	Нет	Нет
	Получение списка применённых планов `GET /api/v1/policy_applications`	Да	Да	Да	Нет
	Запуск плана `POST /api/v1/policy_applications:run`	Да	Нет	Нет	Нет
Реквизиты для входа	Создание реквизитов `POST /api/v1/credentials`	Да	Нет	Нет	Нет
	Получение информации о реквизитах `GET /api/v1/credentials/<credentials_ID>`	Да	Да	Да	Нет
	Удаление реквизитов `DELETE /api/v1/credentials/<credentials_ID>`	Да	Да	Да	Нет
Задача	Получение списка задач `GET /api/v1/tasks`	Да	Да	Да	Да
	Получение информации о задаче `GET /api/v1/tasks/<task_ID>`	Да	Да	Да	Да
Действие	Получение списка действий `GET /api/v1/activities`	Да	Да	Да	Да
	Получение информации о действии `GET /api/v1/activities/<activitiy_ID>`	Да	Да	Да	Да

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

	Конечная точка	Администратор	Только для чтения	Оператор мониторинга	Аудитор
Группа ресурсов	Создание группы `POST /api/v1/groups`	Да	Нет	Нет	Нет
	Обновление группы `PUT /api/v1/groups/<group_ID>`	Да	Нет	Нет	Нет
	Удаление группы `DELETE /api/v1/groups/<group_ID>`	Да	Нет	Нет	Нет
	Добавление и удаление ресурсов `PATCH /api/v1/groups/<group_ID</resources`	Да	Нет	Нет	Нет
Архив	Просмотр архивов `GET /api/v1/archives`	Да	Да	Да	Нет
	Удаление архивов `DELETE /api/v1/archives?vault_id=<vault_ID>&archive_id=<archive_ID>&machine_id=<machine_ID>`	Да	Нет	Нет	Нет
Резервная копия	Просмотр резервных копий `GET /api/v1/backups`	Да	Да	Да	Нет
	Удаление резервных копий `DELETE /api/v1/backups?vault_id=<vault_ID>&archive_id=<archive_ID>&backup_id=<backup_ID>&machine_id=<machine_ID>`	Да	Нет	Нет	Нет

Запросы и ответы API

Связь с конечными точками осуществляется через текстовые сообщения в формате HTTP, при этом само тело сообщения может быть пустым или иметь специальный формат.

Формат указывается в заголовке `Content-Type` сообщения.

Тип сообщения запроса

- Пустое тело сообщения.

  Пример:

```
GET /credentials/e5afb5e8-84b6-415b-969d-bc10d19f3301 HTTP/1.1
Host: https://installaddress.ru:433/api/v1
Authorization: Bearer dXNlcjpwYXNz...
```

- Тело сообщения в формате JSON (в основном).

  Пример:

```
POST /credentials HTTP/1.1
Host: https://installaddress.ru:443/api/v1
Content-Type: application/json
Authorization: Bearer dXNlcjpwYXNz...
{
    ”scope”: {
        ”type”: ”folder”
        },
    ”description”: ”новое описание”,
    ”identity”: ”user_name”,
    ”secret”: ”user_pass”
}
```

- Тело сообщения в форме HTML.

  Пример:

```
POST /idp/token HTTP/1.1
Host: https://installaddress.ru:443
Content-Type: application/x-www-form-urlencoded
Authorization: Bearer dXNlcjpwYXNz

grant_type=password
```

Тип сообщения ответа

- Пустое тело сообщения.

  Пример:

```
HTTP/1.1 204 No Content
<other headers...><empty line>
```

- Тело сообщения в формате JSON (в основном).

  Пример:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Cache-Control: no-store
Pragma: no-cache
<other headers...>

{
    ”access_token”: ”eyJhbGciOiJSUzI1NiIsImVhc...”,
    ”token_type”: ”bearer”,
    ”expires_in”: 86399,
    ”id_token”: ”eyJhbGciOiJSUzI1NiIsImtpZCI6I...”,
    ”scope”: null
}
```

- Тело сообщения в форме HTML

  Пример:

```
HTTP/1.1 200 OK
Content-Type: text/html<other headers...>

<!DOCTYPE html>
<html>
<head>
<meta charset=”UTF-8”>
</head>
<h5>Please wait...</h5>
</body>
</html>
```

В некоторых случаях URL-адреса конечных точек могут быть дополнены параметрами запроса.

Разработчики редко создают HTTP-сообщения самостоятельно: это действие выполняет программное обеспечение, веб-браузер, прокси или веб-сервер. Хорошими инструментами для работы с такими сообщениями являются curl, PowerShell или Postman.

Например, чтобы получить информацию о тенанте из конечной точки `/tenants/<tenant ID>` с помощью `curl`, используйте следующую команду:

```
curl -isX GET https://installaddress.ru/api/v1/credentials/<your credentials ID> \
     -H ”Authorization: Bearer <your token>”
```

Описания параметров `curl`, используемых в этом примере, приведены на странице руководства по curl.

Результат ответа может быть следующим:

```
HTTP/1.1 200 OK
Server: nginx
Date: Mon, 17 Jun 2019 06:55:25 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 573
<other headers...>

{
    ”scope”: {
        ”type”: ”folder”,
        ”subtype”: ””
    },
    ”description”: ”новое описание”,
    ”identity”: ”user_name”,
    ”id”: ”b12e9914-ecea-4869-8e55-3e62a6dd8bb8”,
    ”tenant”: ”8f2ce031-3a88-448a-b6b6-84fc8728eab4”,
    ”version”: 0,
    ”secret_autogenerated”: true,
    ”created_at”: ”2025-11-05T14:15:50Z”,
    ”updated_at”: ”2025-11-05T14:15:50Z”
}
```

Полный набор конечных точек, их описаний, поддерживаемых ими HTTP-методов, принимаемых ими данных запроса и генерируемых ими ответов описаны в справочнике API.

Параметры запросов API

Параметры запросов используются для ограничения выборки запрошенных ресурсов, поиска и фильтрации.

Операция	Описание	Рекомендуемая методика	Пример
Фильтрация (соответствие)	Отфильтровать набор ресурсов или ресурс точно соответствующие значению.	Оператор не требуется.	`?vaults=1aaahd`
Фильтрация (not)	Отфильтровать набор ресурсов или ресурс с использованием выражения ”не равно” для атрибута.	Используется оператор **ne** для атрибута.	`?vaults=ne(43ddaaef)`
Фильтрация (больше или равно)	Отфильтровать набор ресурсов или ресурс с использованием выражения ”больше или равно” для атрибута.	Используется оператор **ge** для атрибута.	`?enqueue_time=ge(2009-11-↪10T23:00:00Z)`
Фильтрация (больше)	Отфильтровать набор ресурсов или ресурс с использованием выражения ”больше” для атрибута.	Используется оператор **gt** для атрибута.	`?enqueue_time=gt(2009-11-↪10T23:00:00Z)`
Фильтрация (меньше или равно)	Отфильтровать набор ресурсов или ресурс с использованием выражения ”меньше или равно” для атрибута.	Используется оператор **le** для атрибута.	`?enqueue_time=le(2009-11-↪10T23:00:00Z)`
Фильтрация (меньше)	Отфильтровать набор ресурсов или ресурс с использованием выражения ”меньше” для атрибута.	Используется оператор **lt** для атрибута.	`?enqueue_↪time=lt(1568822944000000ns)`

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Операция	Описание	Рекомендуемая методика	Пример
Фильтрация (внутри интервала)	Отфильтровать набор ресурсов с использованием условия ”внутри интервала” для атрибута.	Используется оператор **range** для атрибута.	`?enqueue_time=range(2009-11-10T23:00:00Z,2009-11-11T23:00:00Z)`
Фильтрация (вне интервала)	Отфильтровать набор ресурсов с использованием условия ”вне интервала” для атрибута.	Используется оператор **xrange** для атрибута.	`?enqueue_time=xrange(2009-11-10T23:00:00Z,2009-11-11T23:00:00Z)`
Фильтрация (”и”) для значений одного атрибута	Отфильтровать набор ресурсов с использованием условия ”и” для значений одного атрибута.	Используется оператор **and** для атрибута. Синтаксис: `field=and(v1,v2,v3,и т.д.)`. Оператор **and** возможно использовать с операторами **hlike**, **like**, **tlike**.	`?resources.name=and(hlike(hit),like(jar))`
Фильтрация (”или”) для значений одного атрибута	Отфильтровать набор ресурсов с использованием условия ”или” для значений одного атрибута.	Используется оператор **or** для атрибута. Синтаксис: `field=or(v1,v2,v3,и т.д.)`.	`?resource.name=or(Critical data,CEO laptop)` Будут выбраны ресурсы со значениями `Critical data` или `CEO laptop`.
Фильтрация (”содержит”)	Отфильтровать набор ресурсов с использованием условия ”содержит” для атрибута.	Используется оператор **like** для атрибута. Синтаксис: `field=like(value)`.	`?resource.name=like(lap)` Будут выбраны ресурсы со значениями `laptop`, `thunderclap` и т. д.
Фильтрация (начинается с)	Отфильтровать набор ресурсов с использованием условия ”начинается с” для атрибута.	Используется оператор **hlike** для атрибута. Синтаксис: `field=hlike(value)`.	`?resource.name=hlike(CEO)` Будут выбраны ресурсы со значениями `CEO laptop`, `CEO's machines` и т. д.

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Операция	Описание	Рекомендуемая методика	Пример
Фильтрация (заканчивается)	Отфильтровать набор ресурсов с использованием условия ”заканчивается” для атрибута.	Используется оператор **tlike** для атрибута. Синтаксис: `field=tlike(value)`.	`?resource.name=tlike(top)` Будут выбраны ресурсы со значениями `laptop`, `tabletop` и т. д.
Сортировка (по возрастанию, по убыванию)	Переопределить порядок сортировки по умолчанию набора ресурсов.	Для сортировки используется атрибут **order** с префиксами **asc** для сортировки по возрастанию и **desc** для сортировки по убыванию. Синтаксис: `order=asc(f1), desc(f2)`.	`?order=asc(date_of_birth),desc(zip_↪code)` Данный пример запрашивает данные с сортировкой по возрастанию по атрибуту `date_of_birth` в первую очередь и далее по `zip_code` по убыванию.

Коды ответа API

Проверка кодов ответов API

Ответам API присваиваются определенные числовые коды, которые позволяют быстро определить, был ли запрос к конечной точке выполнен успешно или нет, и если нет, то почему.

В отдельных главах API приведены коды ошибок и их подробные описания для каждой конечной точки API.

Проверка кодов ответов

Всем ответам API присваиваются конкретные числовые коды состояний HTTP, которые позволяют быстро определить, был ли запрос к конечной точке выполнен успешно, а если нет, то почему он не был выполнен. Вы можете проверить коды состояний с помощью curl, добавив параметр`--include`:

```
curl -X POST -s <Cyberprotect product URL>/<Cyberprotect product API location>/status \
     --header Authorization: Bearer <your token> \
     --data ”{...}”
     --include
```

Коды успеха

Код будет содержаться в первой строке вывода ответа.

Для успешного выполнения запроса платформа также может предоставить объект JSON в теле ответа, содержащий информацию об операции.

Например:

```
HTTP/1.1 200 OK
<...>
{
  ”created_by”: ”f90e086b...”,
  ”last_access_at”: ”2020-05-22T10:13:28+00:00”,
  ”tenant_id”: ”e5afb5e8...”,
  ...
}
```

Коды ошибок

Для получения кода ошибки платформа может включить в тело ответа объект JSON, содержащий информацию о конкретной ошибке. Эти коды специфичны для каждого API и конечной точки.

Значение	Описание
`error.code`	Код платформы или HTTP-код (см. *Коды состояния HTTP*). Коды и описания платформы подробно описаны в отдельных главах, посвященных API.
`error.message`	Краткое сообщение с указанием причины ошибки.
`error.context`	Объект с дополнительной информацией об ошибке.
`error.domain`	Предметная область, в которой произошла ошибка. Возможные значения приведены в таблице далее.

Возможные значения `error.domain` в зависимости от кода ошибки:

Код ошибки	domain
400 BadRequestError	General
404 NotFoundError	
406	
409 DataConflictError	
415	
422 Issue	
500 InternalServerError	

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Код ошибки	domain
401 UnauthorizedRequestError	Access
401 AuthenticationTimeoutError	
403 AccessDeniedError	

Например:

```
HTTP/1.1 400 Bad Request
<...>
{
  ”error”:
    {
        ”code”:1006,
        ”message”:”It is prohibited to delete a non-disabled tenant.”,
        ”context”:
              {
                    ”id”:”1415032”
              },
        ”domain”:”General”
   }
}
```

Коды состояния HTTP

Ниже приводится сводка кодов состояния HTTP, возвращаемых API, и описание каждого из них. В справочной документации по API приведен список кодов ошибок API и описаний для каждой конечной точки.

Код	Описание
20x	Запрос к конечной точке был выполнен успешно, и тело ответа может содержать объект JSON с результатами.
400	Запрос к конечной точке завершился ошибкой, и тело ответа содержит объект JSON с подробными сведениями об ошибке.
401	Не удается обработать запрос к конечной точке, поскольку срок действия токена, указанного в заголовке `Authorization`, истек или он недействителен.
403	Не удается обработать запрос к конечной точке, поскольку ваша учетная запись не имеет права доступа к этой конечной точке.
404	Запрос к несуществующей конечной точке или в службе не найдены запрошенные данные.

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Код	Описание
405	В запросе используется метод, который не поддерживается конечной точкой.
409	Другой объект того же типа уже существует.
415	В запросе используется формат, который не поддерживается конечной точкой. Эта проблема может возникнуть, если в запросе был указан неправильный заголовок или заголовок `Content-Type` отсутствовал.
50x	Проблема в работе службы. Мы рекомендуем повторить запрос дважды с интервалом в 1-2 секунды. Если ошибка не устраняется, обратитесь к администратору компании или повторите попытку позже.

Сортировка и постраничные ответы

Постраничные ответы

Некоторые конечные точки могут выдавать большое количество результатов, что приводит к чрезмерному объему интернет-трафика и нагрузке на сервер.

Чтобы уменьшить количество результатов, эти конечные точки используют механизм постраничного разбиения, позволяющий им возвращать результаты на нескольких страницах.

Размер страницы определяется параметром `limit` строки запроса.

Если этот параметр явно не задан, конечная точка вернет количество результатов по умолчанию, указанное для параметра строки запроса `limit` в справочнике API.

Например, если имеется 99 результатов, а для параметра `limit` строки запроса задано значение 50 результатов, API разделит результаты на две страницы, отобразив первую страницу с 50 результатами и указателем, ведущим на вторую страницу с оставшимися 49 результатами.

Пример подобного запроса, выполняемого через curl:

```
curl -isX GET -G <product URL>/<product API location>/<endpoint> \
     --data ”limit=50” \
     -H Authorization: Bearer <your token>
```

Когда будет доступна следующая или предыдущая страница, загрузка будет включать токен страницы. Токен страницы — это строка, которая кодирует параметры запроса и позволяет перейти к следующей или предыдущей странице результатов.

Примечание

Некоторые конечные точки могут поддерживать переход только в прямом направлении, остальные — в обоих направлениях.

Когда следующая страница с результатами будет доступна, ответ будет содержать токен страницы `after`:

```
HTTP/1.1 200 OK
Server: nginx
Content-Type: application/json
<other headers...>

{"items":[...],"paging":{"cursors":{"after":"0A8AAAAAAAAAAA=="}}, ...}
```

Чтобы перейти к следующей странице, конечные точки могут принимать параметр строки запроса `after`, который должен использоваться вместе с параметром строки запроса `limit`. Обратите внимание, что токен страницы `after` должен быть закодирован в URL:

```
curl -isX GET -G <product URL>/<product API location>/<endpoint> \
     --data "limit=50" \
     --data-urlencode "after=0A8AAAAAAAAAAA==" \
     -H Authorization: Bearer <your token>
```

Когда вы перейдете на следующую страницу результатов, ответ также будет содержать токен страницы `before`, который позволит вам вернуться на предыдущую страницу:

```
HTTP/1.1 200 OK
Server: nginx
Content-Type: application/json
<other headers...>

{"items":[...],"paging":{"cursors":{"before":"AAAAAAAAAAAAAA=="}}, ...}
```

Для перехода на предыдущую страницу конечные точки могут принимать параметр строки запроса `before`, который должен использоваться вместе с параметром строки запроса `limit`. Обратите внимание, что токен страницы `before` должен быть закодирован в URL:

```
curl -isX GET -G <product URL>/<product API location>/<endpoint> \
     --data "limit=50" \
     --data-urlencode "before=AAAAAAAAAAAAAA==" \
     -H Authorization: Bearer <your token>
```

Сортировка

Параметры запроса СЛЕДУЕТ использовать только с целью ограничения набора ресурсов или в качестве критериев поиска или фильтрации.

Хотя в соответствии со стандартами ограничения по URL-адресам не существует, тем не менее разные браузеры и серверы устанавливают ограничения по URL-адресам. Поэтому при большом количестве фильтруемых элементов, пожалуйста, обратите внимание на рекомендацию по использованию большого количества фильтруемых элементов.

Параметры разбивки на страницы ДОЛЖНЫ соответствовать рекомендациям по разбивке на страницы.

Порядок сортировки по умолчанию СЛЕДУЕТ рассматривать как неопределенный и недетерминированный. В этом случае сервер может выбрать для возврата результатов порядок сортировки по умолчанию.

Если требуется явный порядок сортировки, СЛЕДУЕТ использовать параметр запроса `order` со следующим общим синтаксисом: `asc|desc(field_name),asc|desc(field_name2)`, где `asc` означает возрастание, а `desc` — убывание.

Пример: `/accounts?order=asc(date_of_birth),desc(zip_code)`. В этом примере сервер запрашивает выдачу учетных записей, отсортированных сначала по параметру `date_of_birth` в порядке возрастания, а затем по параметру `zip_code` в порядке убывания.

Подробнее о фильтрации ресурсов описано на странице *Параметры запросов API*.

Запросы API, инициирующие длительные операции, и ожидание их окончания

Ряд сценариев выполняется в асинхронном режиме, так как их выполнение занимает длительное время. При запуске сценария в ответе может предоставляться идентификатор задачи или активности. Для получения статуса выполнения сценария необходимо выполнить запрос по идентификатору полученной задачи, активности или другой эндпоинт (см. примеры). Примеры запросов описаны ниже.

Примеры запросов

Запрос для запуска плана резервного копирования

Метод и путь конечной точки:

```
POST /policy_applications:run
```

Заголовки:

```
Accept: application/json
Content-Type: application/json
Authorization: Bearer¬
↪eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjRlZDM4ZWRlLTFkZTQtNDhmNC1hM2EyLTQ3ZGViNDdiZjEwZCJ9.
↪..
```

Тело:

```
{
”state”: ”running”,
”policy_id”: ”53b86e12-3c65-4721-bf31-92ddebfa769f”,
”context_ids”: [”5b6ff2dc-9095-4718-bc93-d25c70aaa130”]
}
```

Ответ

Код:

```
202
```

Описание:

```
Accepted for asynchronous run invocation
```

Поллинг: статус выполнения плана резервного копирования

Запрос для проверки статуса выполнения

Метод и путь конечной точки:

```
GET /policy_applications?policy_id=53b86e12-3c65-4721-bf31-92ddebfa769f&context_id=5b6ff2dc-9095-
↪4718-bc93-d25c70aaa130
```

Заголовки:

```
Accept: application/json
Authorization: Bearer¬
↪eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjRlZDM4ZWRlLTFkZTQtNDhmNC1hM2EyLTQ3ZGViNDdiZjEwZCJ9.
↪..
```

Ответ: резервное копирование в процессе

Код:

```
200
OK
```

Тело запроса (резервное копирование в процессе):


```
{
    "items": [
        [
            {
                "id": "296a5dba-d31d-4026-a21c-2cff2ecb120f",
                "created_at": "2024-11-28T09:53:05.499539125Z",
                "updated_at": "2024-11-29T06:22:51.484848419Z",
                "deleted_at": null,
                "enabled": true,
                "DEPLOYMENT": {
                    "STATE": "deployed"
                },
                "RUNNING": {
                    "STATE": "running",
                    "PROGRESS": "20"
                },
                "last_success_time": "2024-11-28T11:13:45Z",
                "last_runtime": "2024-11-29T06:22:51Z",
                "last_activity": "policy.backup.machine",
                "next_run_time": "2024-11-29T10:01:14Z",
                "next_activity": "policy.backup.machine",
                "STATUS": "running",
                "agent_id": "bc083994-98f4-4365-97a5-7662e18ba0b2",
                "context": {
                    "id": "5b6ff2dc-9095-4718-bc93-d25c70aaa130",
                    "type": "resource.machine"
                },
                "policy": {
                    "id": "53b86e12-3c65-4721-bf31-92ddebfa769f",
                    "type": "policy.backup.machine",
                    "name": "ol_test_plan_02"
                },
                "tenant_id": "17189",
```


(продолжается на следующей странице)

(продолжение с предыдущей страницы)


```
                "context_tenant_id": "17190",
                "origin_contexts": [
                    "5b6ff2dc-9095-4718-bc93-d25c70aaa130"
                ]
            }
        ]
    ],
    "paging": {
        "cursors": {
            "total": 1
        }
    }
}
```


Ответ: резервное копирование завершено

Код:

```
200
OK
```

Тело запроса (резервное копирование завершено):


```
{
    "items": [
        [
            {
                "id": "296a5dba-d31d-4026-a21c-2cff2ecb120f",
                "created_at": "2024-11-28T09:53:05.499539125Z",
                "updated_at": "2024-11-29T06:24:06.873787017Z",
                "deleted_at": null,
                "enabled": true,
                "DEPLOYMENT": {
                    "STATE": "deployed"
                },
                "RUNNING": {
                    "STATE": "idle"
                },
                "last_success_time": "2024-11-29T06:23:49Z",
                "last_runtime": "2024-11-29T06:22:38Z",
```


(продолжается на следующей странице)

(продолжение с предыдущей страницы)


```
                "last_activity": "policy.backup.machine",
                "next_run_time": "2024-11-29T10:01:14Z",
                "next_activity": "policy.backup.machine",
                "STATUS": "ok",
                "agent_id": "bc083994-98f4-4365-97a5-7662e18ba0b2",
                "context": {
                    "id": "5b6ff2dc-9095-4718-bc93-d25c70aaa130",
                    "type": "resource.machine"
                },
                "policy": {
                    "id": "53b86e12-3c65-4721-bf31-92ddebfa769f",
                    "type": "policy.backup.machine",
                    "name": "ol_test_plan_02"
                },
                "tenant_id": "17189",
                "context_tenant_id": "17190",
                "origin_contexts": [
                    "5b6ff2dc-9095-4718-bc93-d25c70aaa130"
                ]
            }
        ]
    ],
    "paging": {
        "cursors": {
            "total": 1
        }
    }
}
```


API REST-ресурсы и операции с ними

Ресурс

Ресурс (также известный как рабочая нагрузка) — это источник данных, такой как физические и виртуальные хосты, базы данных, почтовые ящики, веб-сайты и т.д., к которому можно применить план защиты.

Ресурсы служат контекстами, к которым применяются и для которых выполняются правила или параметры политики.

Ресурс может быть атомарным или групповым. Атомарный ресурс может входить в несколько групповых ресурсов. Групповой ресурс можно определить статически (явно задав его участие в группе) или динамически (с помощью набора запросов к ресурсам).

Все операции с ресурсами находятся в конечной точке `/resources`.

Структура объекта JSON ресурса

Описание структуры объекта JSON ресурса доступно по ссылке.

Имя	Тип значения	Описание
`id`	строка UUID	Уникальный идентификатор (UUID) ресурса.
`created_at`	строка	Дата и время создания ресурса согласно стандарту ISO 8601.
`updated_at`	строка	Дата и время последнего обновления ресурса согласно стандарту ISO 8601.
`deleted_at`	строка	Дата и время мягкого удаления (установки флага ”удалён”) ресурса согласно стандарту ISO 8601 или `null` в случае, если ресурс не удалён.
`tenant_id`	строка UUID	Уникальный идентификатор (UUID) тенанта владельца ресурса.
`external_id`	строка UUID	Уникальный идентификатор (UUID) ресурса из внешней системы.
`type`	строка	Тип ресурса в формате `type.scope`, где `type` — это `resource`. Значения `scope` приведены после таблицы.
`settings_schema`	строка	Версия схемы настроек.
`agent_id`	строка	Идентификатор агента.
`group_condition`	строка	Список ресурсов, используемых внутри группы при её создании. Параметр должен включать: • для группы доменов – тип ресурса, домен, экземпляр; • для группы экземпляров – тип ресурса и экземпляр.
`allowed_member_types`	массив строк	Список типов ресурсов, которые можно добавить в группу.
`parent_group_ids`	массив строк UUID	Перечень уникальных идентификаторов (UUID) родительских групп.
`name`	строка	Наименование ресурса.

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Имя	Тип значения	Описание
`user_defined_name`	строка	Наименование ресурса, определённое пользователем.
`tag`	строка	Определённое пользователем строковое значение для фильтрации ресурсов.
`attributes`	объект	Массив атрибутов ресурса. Доступные базовые пространства имён: • `default` • `legacy` • `tenant` • `agent` • `cyberfit` • `backups` • `hwi` • `workload_isolation`

Значения `scope`:

- `machine`
- `datastore.vmwesx`
- `datastore_cluster.vmwesx`
- `virtual_appliance.vmwesx`
- `virtual_application.vmwesx`
- `virtual_center.vmwesx`
- `virtual_cluster.vmwesx`
- `virtual_data_center.vmwesx`
- `virtual_folder.vmwesx`
- `virtual_machine.vmwesx`
- `virtual_network.vmwesx`
- `virtual_resource_pool.vmwesx`
- `virtual_host.vmwesx`
- `virtual_machine.vmww`
- `virtual_cluster.mshyperv`
- `virtual_machine.mshyperv`
- `virtual_host.mshyperv`
- `virtual_appliance.mshyperv`
- `virtual_network.mshyperv`
- `virtual_folder.mshyperv`
- `virtual_data_center.mshyperv`
- `datastore.mshyperv`
- `virtual_machine.parallelsw`
- `virtual_host.parallelsw`
- `virtual_cluster.pcs`
- `virtual_machine.msvpc`
- `virtual_machine.msvs`
- `virtual_machine.pcs`
- `virtual_machine.xen`
- `virtual_machine.kvm`
- `virtual_machine.kvmsrv`
- `virtual_machine.rhev`
- `virtual_host.vmww`
- `virtual_host.msvpc`
- `virtual_host.msvs`
- `virtual_host.pcs`

- virtual_host.xen
- virtual_host.kvm
- virtual_host.kvmsrv
- virtual_host.rhev
- virtual_machine.hci
- virtual_host.hci
- virtual_cluster.hci
- virtual_machine.scale
- virtual_host.scale
- virtual_cluster.scale
- group.computers
- group.all
- group.vcenter
- group.datacenter
- group.hosts_and_clusters
- group.cluster
- group.host
- group.virtual_application
- group.resource_pool
- group.virtual_folder
- group.vms_and_templates
- group.vms_folder
- mssql_server
- mssql_aag_database
- mssql_aag_group
- mssql_database
- mssql_database_folder
- mssql_instance
- mssql_system_database
- exchange
- msexchange_database
- msexchange_storage_group
- msexchange_mailbox.msexchange
- msexchange_mailbox.office365

Структура объекта attributes

Описание атрибутов пространства имён доступно по ссылке.

Имя	Тип значения	Описание
name	строка	Наименование атрибута (пространство имён).
kvs	массив объектов	Массив объектов вида ”ключ”: ”значение”.

Структура объекта kvs

Имя	Тип значения	Описание
key	строка	Наименование ключа.
value	строка или число, массив строк или чисел	Единичное значение или массив из строк или целых чисел.

Структура объекта details

Имя	Тип значения	Описание
key	строка	Наименование ключа.

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Имя	Тип значения	Описание
value	строка или число, массив строк или чисел	Единичное значение, представляющее собой строку или целое число, или массив, состоящий из строк или целых чисел.

Пример ресурса


```
{
    ”id”:”054b939a-3627-41c9-a351-151be3de0980”,
    ”created_at”:”2021-01-18T14:38:24.5589857Z”,
    ”updated_at”:”2021-01-18T14:38:24.5789853Z”,
    ”tenant_id”:”cbad0dcf-fe8f-4989-bc03-27ae36112104”,
    ”external_id”:”054b939a-3627-41c9-a351-151be3de0980@cbad0dcf-fe8f-4989-bc03-27ae36112104”,
    ”type”:”resource.group.computers”,
    ”group_condition”:”comment like 'Simple*'”,
    ”parent_group_ids”:[
       ”301d1574-849e-4714-859f-3a2ec12a218b”
    ],
    ”allowed_member_types”:[
        ”resource.machine”
    ],
    ”name”:”Dynamic group”,
    ”user_defined_name”:”Dynamic group”,
    ”attributes”:[
       {
          ”name”:”group”,
          ”kvs”:[
             {
                ”key”:”parent_id”,
                ”value”:[
                   ”62E5834D-FB8E-500D-956E-02C182D463D2”
                ]
             },
             {
                ”key”:”tz_offset”,
                ”value”:[
                   ”480”
                ]
             }
          ]
```


(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
      }
    ]
}
```

Справочники

Справочники доступны по ссылке.

Действия с ресурсами

Операция	Используемые методы и конечные точки
Получение списка ресурсов	`GET /resources`
Получение подробной информации о ресурсе	`GET /resources/<resource_id>`

Получение списка ресурсов

1. Выполните аутентификацию с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступными следующие переменные:

   ```
   >>> base_url  # the base URL of the API
   'https://installaddress.ru/api/v1'
   >>> auth  # the 'Authorization' header value with the access token
   {'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
   ```

2. Отправьте запрос GET на конечную точку `/resources`:

   ```
   >>> response = requests.get(f'{base_url}/resources', headers=auth)
   ```

3. Проверьте код состояния запроса:

   ```
   >>> response.status_code
   200
   ```

   Код состояния HTTP 200 означает, что запрос был успешно выполнен.

   Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

   Кроме того, тело ответа содержит объект с ключом `items` в формате JSON, содержащий массив объектов ресурсов. После преобразования в объект он будет выглядеть так:

```
>>> pprint.pprint(response.json())
{'items': [{'created_at': '2021-02-03T14:57:07.293995787Z',
            'external_id': '23effcf6-2798-4631-9a52-5785bf3af657@cbad0dcf-fe8f-4989-bc03-
↪27ae36112104',
            'id': '23effcf6-2798-4631-9a52-5785bf3af657',
            'name': 'DESKTOP-JRPTA4A',
            'tenant_id': 'cbad0dcf-fe8f-4989-bc03-27ae36112104',
            'type': 'resource.machine',
            'updated_at': '2021-02-03T18:13:48.312293448Z',
            'user_defined_name': 'DESKTOP-JRPTA4A'},
            ...],
'paging': {'cursors': {"after": ""}}}
```

4. Преобразуйте текст JSON, содержащийся в теле ответа, в объект, а затем извлеките список ресурсов из ответа:

```
>>> resources = response.json()['items']
```

Получение подробной информации о ресурсе

1. Выполните аутентификацию с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Получите список ресурсов, как описано в разделе *Вывод списка ресурсов*. В качестве примера будет взят идентификатор первого ресурса. В результате должна быть доступна следующая переменная:

```
>>> resource_id = resources[0]['id']
>>> resource_id
'054b939a-3627-41c9-a351-151be3de0980'
```

3. Отправьте запрос GET на конечную точку `/resources/{resource_id}`:

```
>>> response = requests.get(f'{base_url}/resources/{resource_id}', headers=auth)
```

4. Проверьте код состояния запроса:

```
>>> response.status_code
200
```

Код состояния HTTP 200 означает, что запрос был успешно выполнен.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

После преобразования в объект ресурс будет выглядеть так:

```
>>> pprint.pprint(response.json())
{
        'id':'054b939a-3627-41c9-a351-151be3de0980',
        'created_at':'2021-01-18T14:38:24.5589857Z',
        'updated_at':'2021-01-18T14:38:24.5789853Z',
        'tenant_id':'00000000-0000-0000-0000-000000000000',
        'external_id':'054b939a-3627-41c9-a351-151be3de0980@00000000-0000-0000-0000-
↪000000000000',
        'type':'resource.group.computers',
        'group_condition':'comment like 'Simple*'',
        'parent_group_ids':[
           '301d1574-849e-4714-859f-3a2ec12a218b'
        ],
        'allowed_member_types':[
            'resource.machine'
        ],
        'name':'Dynamic group',
        'user_defined_name':'Dynamic group',
        'attributes': [{'kvs': [{'key': 'components', 'value': ['HyperVAgent',
↪'x64WindowsAgent']},
                                {'key': 'features', 'value': ['HyperVAgent', 'x64WindowsAgent
↪']},
                                {'key': 'os_family', 'value': 'Windows'},
                                {'key': 'os_name', 'value': 'Microsoft Windows Server 2019¬
↪Standard'},
                                {'key': 'os_arch', 'value': 'x64'},
                                {'key': 'os_family', 'value': 'Windows'},
                                {'key': 'os_caps', 'value': 18},
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                                {'key': 'os_product_type', 'value': 3},
                                {'key': 'os_service_pack', 'value': 0},
                                {'key': 'os_suite_mask', 'value': 272},
                                {'key': 'sku', 'value': 7},
                                {'key': 'os_version_major', 'value': 10},
                                {'key': 'os_version_minor', 'value': 0},
                                {'key': 'mac_addresses', 'value': ['00:50:56:84:DB:B7']},
                                {'key': 'memory_size', 'value': 17178800128},
                                {'key': 'processor_frequency', 'value': 3504},
                                {'key': 'processor_name', 'value': 3504},
                                {'key': 'chassis', 'value': 'other'},
                                {'key': 'vm_host_type', 'value': 'vmwesx'},
                                {'key': 'esx_address', 'value': ''},
                                {'key': 'residental_addresses', 'value': ['10.34.16.186']},
                                {'key': 'inside_virtual', 'value': 1},
                                {'key': 'tz_offset', 'value': 120},
                                {'key': 'version', 'value': '15.0.26355'},
                                {'key': 'enabled', 'value': 1},
                                {'key': 'is_online', 'value': 0}],
                    'name': 'agent'},
                    {'details':   [{'key': 'cyberfit_score_details',
                                    'value': {'bitlocker': {'encrypted_volumes': 0,
                                                            'is_system_volume_encrypted
↪': False,
                                                            'volumes': 2},
                                              'os': {'name': 'Microsoft Windows Server¬
↪2019 Standard',
                                                     'version': '10.0.17763'},
                                              'programs': {'antispyware': [{'name': 'Cyber¬
↪Protect Agent',
                                                                            'vendor':
↪'Cyberprotect',
                                                                            'version': '15.0.
↪26355'}],
                                                           'antivirus': [{'name':
↪'Cyber Protect Agent',
                                                                          'vendor':
↪'Cyberprotect',
                                                                          'version': '15.0.
↪26355'}],
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                                                                'backup': [{'name': 'Cyber¬
↪Protect Agent',
                                                                            'vendor':
↪'Cyberprotect',
                                                                            'version': '15.0.
↪26355'}],
                                                                'disk_encryption': [],
                                                                'firewall': [],
                                                                'vpn': []},
                                                    'windows_defender': {'antispyware': {'enabled
↪': True},
                                                                                    'antivirus': {
↪'enabled': True}},
                                                    'windows_firewall': {'private_network': {
↪'enabled': True},
                                                                                    'public_network': {
↪'enabled': True}},
                                                    'windows_server_backup': {'installed': False}
↪}},
                                            {'key': 'cyberfit_score_value', 'value': 625},
                                            {'key': 'cyberfit_score_server_version', 'value': 38}
↪,
                                            {'key': 'cyberfit_score_client_version', 'value':¬
↪108},
                                            {'key': 'cyberfit_score_assessment_date', 'value':
↪'2021-02-03T18:13:46.9931835Z'},
                                            {'key': 'cyberfit_score_metrics',
                                            'value': {'antimalware': {'current': 275, 'max': 275}
↪,
                                                    'backup': {'current': 175, 'max': 175},
                                                    'disk_encryption': {'current': 0,
                                                                        'max': 125},
                                                    'firewall': {'current': 175, 'max': 175},
                                                    'ntlm': {'current': 0, 'max': 25},
                                                    'vpn': {'current': 0, 'max': 75}}}],
                                'kvs': [{'key': 'cyberfit_score_value', 'value': '625'}],
                                'name': 'cyberfit'},
                            {'kvs': [{'key': 'ip_addresses', 'value': '10.34.16.186'},
                                        {'key': 'ldap_status', 'value': 'ldapoff'},
                                        {'key': 'operating_system_product_type', 'value': 3},
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                                    {'key': 'operating_system_type', 'value': 3},
                                    {'key': 'architecture', 'value': 'x64'},
                                    {'key': 'ip', 'value': '10.34.16.186'},
                                    {'key': 'name', 'value': 'DESKTOP-JRPTA4A'},
                                    {'key': 'operating_system', 'value': 'Microsoft Windows¬
↪Server 2019 Standard'},
                                    {'key': 'os_product_type', 'value': '3'},
                                    {'key': 'residential_address_v4', 'value': '0a2210ba'}],
                          'name': 'default'}]}
```

Группа

Группа — это набор ресурсов, к которому можно применить план защиты. Группы упрощают управление большим количеством рабочих нагрузок.

Группы подразделяются на типы:

- **Статические группы** содержат рабочие нагрузки, добавленные вручную.
- **Динамические группы** содержат рабочие нагрузки, добавленные автоматически в соответствии с поисковыми критериями, определенными при создании группы. Состав динамической группы меняется автоматически. Рабочая нагрузка остается в группе до тех пор, пока отвечает заданным критериям.

После появления рабочей нагрузки в группе, она будет защищена планом. Если рабочая нагрузка удалена из группы, она больше не будет защищена планом.

Все операции с группами находятся в конечной точке `/groups`.

Структура объекта JSON группы

Описание структуры объекта JSON группы доступно по ссылке.

Справочники

Справочники доступны по ссылке.

Действия с группами

Операция	Используемые методы и конечные точки
Создание группы	`POST /groups`
Обновление группы	`PUT /groups/<group_id>`

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Операция	Используемые методы и конечные точки
Удаление группы	`DELETE /groups/<group_id>`
Добавление ресурсов в статическую группу или удаление их из нее	`PATCH /groups/<resource_id>/resources`

Создание группы

1. Выполните аутентификацию с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Запросите ресурсы, как описано в разделе *Получение списка ресурсов*. Задайте переменную `parent_id` и присвойте ей в качестве значения `parent_id` ресурса с параметром `”name”: ”All”`:

```
>>> parent_id = next(
    (item['parent_id'] for item in data['items'] if item['name'] == ”All”),
    None
)
>>> parent_id
301D1574-849E-4714-859F-3A2EC12A218B
```

3. Задайте переменную `group_data`, а затем присвойте этой переменной объект, содержащий параметры создаваемой группы:

```
>>> group_data = {
  'name': 'Static group',
  'membership_type': 'STATIC',
  'parent_id': parent_id,
  'type': 'group.computers',
  'comment': 'Comment static group'
}
```

Примечание

Условия формирования динамической группы передаются в параметре `condition`.

Доступные параметры и правила формирования условий описаны в данном разделе документации.

4. Отправьте запрос POST на конечную точку `/groups`:

```
>>> response = requests.post(
...     f'{base_url}/groups',
...     headers={'Content-Type': 'application/json', **auth},
...     data=group_data,
... )
```

5. Проверьте код состояния запроса:

```
>>> response.status_code
201
```

Код состояния HTTP 201 означает, что запрос был успешно выполнен.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

После преобразования в объект группа будет выглядеть так:

```
>>> pprint.pprint(response.json())
{
  'comment': 'Comment static group',
  'created_at': '2026-03-03T14:45:36.967158383Z',
  'id': '5AFCBCF6-6070-47A9-9569-69C8D3E68CE5',
  'member_types': [
    'machine'
  ],
  'membership_type': 'STATIC',
  'name': 'Static group',
  'parent_id': '301D1574-849E-4714-859F-3A2EC12A218B',
  'tenant_id': '82632930-5a7c-45c1-aec0-f0e8a37ef414',
  'type': 'group.computers',
  'updated_at': '2026-03-03T14:45:36.967158383Z'
}
```

Обновление группы

1. Выполните аутентификацию с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступными следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Запросите ресурсы, как описано в разделе *Получение списка ресурсов*. Задайте переменную `group_id` нужной группы и присвойте ей в качестве значения `id` ресурса. В примере использован ресурс с параметром `”name”: ”Static group”`:

```
>>> group_id = next(
    (item['id'] for item in data['items'] if item['name'] == ”Static group”),
    None
)
>>> group_id
5AFCBCF6-6070-47A9-9569-69C8D3E68CE5
```

3. Задайте переменную `group_data`, а затем присвойте этой переменной объект, содержащий параметры создаваемой группы:

```
>>> group_data = {
  'name': 'Static group 2',
  'comment': 'Новый комментарий для группы'
}
```

4. Отправьте запрос PUT на конечную точку `/groups/{group_id}`:

```
>>> response = requests.put(
...     f'{base_url}/groups/{group_id}',
...     headers={'Content-Type': 'application/json', **auth},
...     data=group_data,
... )
```

5. Проверьте код состояния запроса:

```
>>> response.status_code
200
```

Код состояния HTTP 200 означает, что запрос был успешно выполнен.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

После преобразования в объект группа будет выглядеть так:

```
>>> pprint.pprint(response.json())
{
  'comment': 'Новый комментарий для группы',
  'created_at': '2026-03-03T14:45:36.967158383Z',
  'id': '5AFCBCF6-6070-47A9-9569-69C8D3E68CE5',
  'member_types': [
    'machine'
  ],
  'membership_type': 'STATIC',
  'name': 'Static group 2',
  'parent_id': '301D1574-849E-4714-859F-3A2EC12A218B',
  'tenant_id': '82632930-5a7c-45c1-aec0-f0e8a37ef414',
  'type': 'group.computers',
  'updated_at': '2026-03-03T14:45:36.967158383Z'
}
```

Удаление группы

1. Выполните аутентификацию с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступными следующие переменные:

   ```
   >>> base_url  # the base URL of the API
   'https://installaddress.ru/api/v1'
   >>> auth  # the 'Authorization' header value with the access token
   {'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
   ```

2. Запросите ресурсы, как описано в разделе *Получение списка ресурсов*. Задайте переменную `group_id` нужной группы и присвойте ей в качестве значения `id` ресурса. В примере использован ресурс с параметром `"name": "Static group"`:

```
>>> group_id = next(
    (item['id'] for item in data['items'] if item['name'] == ”Static group”),
    None
)
>>> group_id
5AFCBCF6-6070-47A9-9569-69C8D3E68CE5
```

3. Отправьте запрос DELETE на конечную точку `/groups/{group_id}`:

```
>>> requests.delete(f'{base_url}/groups/{group_id}', headers=auth)
```

4. Проверьте код состояния запроса:

```
>>> response.status_code
204
```

Код состояния HTTP 204 означает, что запрос был успешно выполнен.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Добавление ресурсов в статическую группу или удаление их из нее

1. Выполните аутентификацию с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступными следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Запросите ресурсы, как описано в разделе *Получение списка ресурсов*. Задайте переменную `group_id` нужной группы и присвойте ей в качестве значения `id` ресурса. В примере использован ресурс с параметром `”name”: ”Static group”`:

```
>>> group_id = next(
    (item['id'] for item in data['items'] if item['name'] == ”Static group”),
    None
)
>>> group_id
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
5AFCBCF6-6070-47A9-9569-69C8D3E68CE5
```

3. В примере будем добавлять ресурсы в группу. Задайте переменную `group_data`, а затем присвойте этой переменной объект, содержащий перечень ресурсов, которые нужно добавить в группу:

```
>>> group_data = {
  'add': [
    '301D1574-849E-4714-859F-3A218A2EC12B',
    '301D1574-849E-4714-859F-3A2A21EC128B'
  ]
}
```

4. Отправьте запрос PATCH на конечную точку `/groups/{group_id}/resources`:

```
>>> response = requests.patch(
...     f'{base_url}/groups/{group_id}/resources',
...     headers={'Content-Type': 'application/json', **auth},
...     data=group_data,
... )
```

5. Проверьте код состояния запроса:

```
>>> response.status_code
204
```

   Код состояния HTTP 204 означает, что запрос был успешно выполнен.

   Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Политика

Политика (план резервного копирования) — это набор настроек для обеспечения защиты данных, который можно применить к ресурсам и запустить с помощью агента защиты. Политика используется для настройки резервного копирования и восстановления данных.

Все операции с политиками находятся в конечной точке `/policies`.

Типы политик

Префикс имени	Описание
`policy.backup`	Политика резервного копирования.
`policy.recovery`	Политика восстановления данных (запланировано в следующих версиях документации открытого API).

Структура объекта JSON политики

Описание структуры объекта JSON политики доступно по ссылке.

Имя	Тип значения	Описание
`id`	строка UUID	Уникальный идентификатор (UUID) политики.
`created_at`	строка	Дата и время создания политики согласно стандарту ISO 8601.
`updated_at`	строка	Дата и время последнего обновления политики согласно стандарту ISO 8601.
`deleted_at`	строка	Дата и время мягкого удаления (установки флага ”удалён”) политики согласно стандарту ISO 8601 или `null` в случае, если ресурс не удалён.
`tenant_id`	строка UUID	Уникальный идентификатор (UUID) тенанта владельца политики.
`type`	строка	Тип политики. Наименования типов политик, доступных в текущей версии API, описаны в разделе *Справочник по настройкам политик*.
`enabled`	логическое значение	Флаг, который отключает или включает политику на платформе.
`name`	строка	Название политики.
`settings`	объект	Описание параметров настроек политик, доступных в текущей версии API, представлено в разделе *Справочник по настройкам политик*.
`settings_constraints`	строка, число, логическое значение или объект	Ограничивающие правила для текущих параметров политики. Допустимые значения: диапазоны чисел, маски строковых значений и др.
`validation_dependencies`	массив строк UUID	Ссылки на зависимости. Допускаются только уникальные идентификаторы (UUID) дочерних политик в рамках одного составного элемента (иные идентификаторы игнорируются). Должен заполняться на первом этапе проверки политики.

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Имя	Тип значения	Описание
`ls_features`	строка	Перечень лицензий, рассчитанный на основе полей ”settings”. Не заполняется, когда поле ”enabled” имеет значение false.

Пример политики

Примеры политик доступны в разделе *Параметры политики защиты*.

Справочники

Справочники доступны по ссылке.

Действия с политиками

Операция	Используемые методы и конечные точки
Создание плана резервного копирования	`POST /policies`
Получение перечня планов резервного копирования	`GET /policies`
Получение информации о плане резервного копирования	`GET /policies/{policy_id}`
Редактирование параметров плана резервного копирования	`PATCH /policies/{policy_id}`
Включение/выключение плана резервного копирования	`PATCH /policy_applications/enabled`
Удаление плана резервного копирования	`DELETE /policies/{policy_id}`

Параметры политики защиты

Политика резервного копирования машины

Политика обеспечивает функциональность резервного копирования рабочей нагрузки машины целиком.

Дополнительные сведения о параметрах планов резервного копирования см. в документации по продукту.

Описание параметров запроса для создания политики резервного копирования см. в справочнике API.

Следующий пример можно использовать при создании планов защиты с данной политикой:

```
{
    'enabled': true,
    'ls_features': ' ',
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    'name': 'физическая машина. резервная копия всей машины',
    'settings': {
        'alerts': {
            'enabled': false,
            'max_days_without_backup': 0
        },
        'archive': {
            'compression': 'MAX',
            'encryption': {
                'algorithm': 'AES192',
                'password': 'string'
            },
            'format': '12',
            'name': '[Machine Name]-[Plan ID]-[Unique ID]A[Plan ID][Plan name][Unique¬
↪ID][Virtualization Server Type]F',
            'splitting': {
                'size': 214748364800
            }
        },
        'cbt': 'USE_IF_ENABLED',
        'check_file_timestamp': true,
        'destination': {
            'location': {
                'settings': {
                    'name': 'customer-dg09',
                    'storage_type': 'LOCAL_FOLDER',
                    'uri': 'E:/',
                    'use_policy_credentials': true
                },
                'type': 'VAULT'
            },
            'performance_window': {
                'enabled': false
            },
            'retention': {
                'execution': 'AFTER_BACKUP',
                'rules': [
                    {
                        'max_count': 7
                    }
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
            ]
        }
    },
    'error_handling': {
        'enabled': true,
        'interval': {
            'count': 1,
            'type': 'MINUTES'
        },
        'max_attempts': 1
    },
    'exclude_asz': true,
    'exclude_page_files': true,
    'fast_backup_enabled': true,
    'file_filters': {
        'exclude_hidden': true
    },
    'ignore_bad_sectors': true,
    'lvm_snapshotting_enabled': true,
    'multi_volume_snapshotting_enabled': true,
    'notarization': {
        'enabled': true
    },
    'owner_id': '32e95e91-39f4-6053-9ce4-0da7e86c8b81',
    'preserve_file_security_settings': true,
    'reset_free_blocks_list': true,
    'scheduling': {
        'backup_sets': [
            {
                'schedule': {
                    'alarms': {
                        'time': {
                            'rand_max_delay': {
                                'count': 30,
                                'type': 'MINUTES'
                            },
                            'repeat_at': [
                                {
                                    'hour': 9,
                                    'minute': 0
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                            }
                        ],
                        'weekdays': [
                            'MON',
                            'TUE',
                            'WED'
                        ]
                    }
                },
                'conditions': {},
                'prevent_sleep': true,
                'type': 'DAILY'
            },
            'type': 'AUTO'
        }
    ],
    'enabled': true,
    'rand_max_delay': {
        'count': 30,
        'type': 'MINUTES'
    },
    'scheme': 'WEEKLY_FULL_DAILY_INCREMENTAL',
    'weekly_backup_day': 'MON'
},
'sector_by_sector': true,
'silent_mode_enabled': true,
'tapes': {
    'devices': null,
    'eject_media_if_successfull': false,
    'multiplexing': {
        'enabled': false,
        'streams_per_drive': 0
    },
    'multistreaming': {
        'enabled': false
    },
    'overwrite_data_on_tape': false,
    'preserve_tapes_position': true,
    'tapeset': ''
},
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
        'validate_backup': true,
        'vss': {
            'enabled': true,
            'provider': 'NATIVE'
        }
    },
    'tenant_id': '00000000-0000-0000-0000-000000000000',
    'type': 'policy.backup.machine'
}
```

Политика резервного копирования диска

Политика обеспечивает функциональность резервного копирования рабочей нагрузки диска или тома машины.

Дополнительные сведения о параметрах планов резервного копирования см. в документации по продукту.

Описание параметров запроса для создания политики резервного копирования см. в справочнике API.

Следующий пример можно использовать при создании планов защиты с данной политикой:

```
{
    'enabled': true,
    'ls_features': '',
    'name': 'виртуальная машина. резервная копия диска',
    'settings': {
        'alerts': {
            'enabled': false,
            'max_days_without_backup': 0
        },
        'archive': {
            'compression': 'NORMAL',
            'format': 'AUTO',
            'name': '[Machine Name]-[Plan ID]-[Unique ID]A',
            'splitting': {
                'size': 214748364800
            }
        },
        'cbt': 'ENABLE_AND_USE',
        'destination': {
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
        'location': {
            'settings': {
                'path': 'F:/local_auto_backups/'
            },
            'type': 'LOCAL_FOLDER_TEMPLATE'
        },
        'performance_window': {
            'enabled': false
        },
        'retention': {
            'execution': 'AFTER_BACKUP',
            'rules': [
                {
                    'backup_set': 'DAILY',
                    'max_age': {
                        'count': 7,
                        'type': 'DAYS'
                    }
                },
                {
                    'backup_set': 'WEEKLY',
                    'max_age': {
                        'count': 4,
                        'type': 'WEEKS'
                    }
                },
                {
                    'backup_set': 'MONTHLY',
                    'max_age': {
                        'count': 6,
                        'type': 'MONTHS'
                    }
                }
            ]
        }
    },
    'error_handling': {
        'enabled': true,
        'interval': {
            'count': 30,
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
            'type': 'SECONDS'
        },
        'max_attempts': 300
    },
    'fast_backup_enabled': true,
    'file_filters': {},
    'fixed_drive_behavior_for_removable': true,
    'multi_volume_snapshotting_enabled': true,
    'notarization': {
        'enabled': false
    },
    'owner_id': 'daec7891-3a66-ed85-7aa6-2dbb0d6e0c4e',
    'preserve_file_security_settings': true,
    'quiesce_snapshotting_enabled': true,
    'scheduling': {
        'backup_sets': [
            {
                'schedule': {
                    'alarms': {
                        'time': {
                            'monthdays': [
                                1,
                                15
                            ],
                            'months': [
                                1,
                                2,
                                3,
                                4,
                                5,
                                6,
                                7,
                                8,
                                9,
                                10,
                                11,
                                12
                            ],
                            'repeat_at': [
                                {
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                                    'hour': 12,
                                    'minute': 30
                                }
                            ]
                        }
                    },
                    'conditions': {
                        'users_idle': true
                    },
                    'prevent_sleep': true,
                    'type': 'MONTHLY'
                },
                'type': 'FULL'
            }
        ],
        'enabled': true,
        'rand_max_delay': {
            'count': 30,
            'type': 'MINUTES'
        },
        'scheme': 'ALWAYS_FULL',
        'weekly_backup_day': 'MON'
    },
    'selection': {
        'items': [
            {
                'id': '[[\'LocalID\',\'\\\\local\\\\hd_ev\\\\vol_
↪guid(8FE6E78C44C9D8F9B7B6A0950B209597)\'],[\'ItemType\',\'mms::disk::volume\'],[\'0B781614-5AED-
↪4A10-9B79-0A607CB7EEAE\',\'491168AF-257C-4FD2-8952-F76260B943C0\'],[\'4B2A7A93-A44F-4155-BDE3-
↪A023C57C9431\',\'gct::disks\']]',
                'name': 'C:'
            }
        ]
    },
    'silent_mode_enabled': true,
    'tapes': {
        'devices': null,
        'eject_media_if_successfull': false,
        'multiplexing': {
            'enabled': false,
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                'streams_per_drive': 0
            },
            'multistreaming': {
                'enabled': false
            },
            'overwrite_data_on_tape': false,
            'preserve_tapes_position': true,
            'tapeset': ''
        },
        'vm_snapshot_error_handling': {
            'enabled': true,
            'interval': {
                'count': 5,
                'type': 'MINUTES'
            },
            'max_attempts': 3
        },
        'vss': {
            'enabled': true,
            'provider': 'TARGET_SYSTEM_DEFINED'
        }
    },
    'tenant_id': 'ca1adfdd-f0d4-4644-a22f-f6a1881d17dd',
    'type': 'policy.backup.disks'
}
```

Политика резервного копирования баз данных Microsoft SQL Server

Политика обеспечивает функциональность резервного копирования рабочей нагрузки машины целиком.

Дополнительные сведения о параметрах планов резервного копирования см. в документации по продукту.

Описание параметров запроса для создания политики резервного копирования см. в справочнике API.

Следующий пример можно использовать при создании планов защиты с данной политикой:

```
{
    ”enabled”: true,
    ”ls_features”: ””,
    ”name”: ”MS SQL. резервная копия баз данных”,
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
"settings": {
    "alerts": {
        "enabled": true,
        "max_days_without_backup": 5
    },
    "archive": {
        "comments": "",
        "compression": "HIGH",
        "display_name": "",
        "format": "AUTO",
        "name": "[Machine Name]-[Plan ID]-[Unique ID]A",
        "slice_comments": "",
        "splitting": {
            "auto": true
        }
    },
    "cluster": {
        "strategy": "REPLICA_FIRST"
    },
    "destination": {
        'location': {
            'settings': {
                'name': 'customer-dg09',
                'storage_type': 'LOCAL_FOLDER',
                'uri': 'E:/',
                'use_policy_credentials': true
            },
            'type': 'VAULT'
        },
        "performance_window": {
            "enabled": false
        },
        "retention": {
            "execution": "AFTER_BACKUP",
            "rules": [
                {
                    "backup_set": "DAILY",
                    "max_age": {
                        "count": 7,
                        "type": "DAYS"
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                    }
                },
                {
                    "backup_set": "WEEKLY",
                    "max_age": {
                        "count": 4,
                        "type": "WEEKS"
                    }
                }
            ]
        }
    },
    "error_handling": {
        "enabled": true,
        "interval": {
            "count": 30,
            "type": "SECONDS"
        },
        "max_attempts": 300,
        "silent_mode": true
    },
    "express_backup_usage": {
        "continue_files_tracking": true,
        "dont_create_session": false,
        "fallback_mode": "AUTO",
        "session_timeout": 0
    },
    "scheduling": {
        "backup_sets": [
            {
                "schedule": {
                    "alarms": {
                        "time": {
                            "rand_max_delay": {
                                "count": 30,
                                "type": "MINUTES"
                            },
                            "repeat_at": [
                                {
                                    "hour": 17,
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)


```
                                    "minute": 0
                                }
                            ],
                            "weekdays": [
                                "MON",
                                "TUE",
                                "WED",
                                "THU",
                                "FRI"
                            ]
                        }
                    },
                    "prevent_sleep": true,
                    "type": "DAILY"
                },
                "type": "AUTO"
            }
        ],
        "enabled": true,
        "max_parallel_backups": 1,
        "rand_max_delay": {
            "count": 30,
            "type": "MINUTES"
        },
        "scheme": "WEEKLY_FULL_DAILY_INCREMENTAL",
        "task_failure": {
            "enabled": true,
            "interval": {
                "count": 1,
                "type": "HOURS"
            },
            "max_attempts": 5
        },
        "weekly_backup_day": "MON"
    },
    "tapes": {
        "eject_full_media": false,
        "eject_media_if_successfull": false,
        "multiplexing": {
            "enabled": false,
```


(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                "streams_per_drive": 0
            },
            "multistreaming": {
                "enabled": false
            },
            "overwrite_data_on_tape": false,
            "preserve_tapes_position": true,
            "tapeset": ""
        },
        "truncate_logs": true,
        "vss": {
            "enabled": true,
            "provider": "TARGET_SYSTEM_DEFINED"
        },
        "windows_event_log": {
            "enabled": true,
            "level": "ERRORS_AND_WARNINGS"
        }
    },
    "tenant_id": "f1579c67-f143-480b-9a13-095a345c6b0f",
    "type": "policy.backup.mssql"
}
```

Политика резервного копирования баз данных Microsoft Exchange Server

Политика обеспечивает функциональность резервного копирования рабочей нагрузки машины целиком.

Дополнительные сведения о параметрах планов резервного копирования см. в документации по продукту.

Описание параметров запроса для создания политики резервного копирования см. в справочнике API.

Следующий пример можно использовать при создании планов защиты с данной политикой:

```
{
    'enabled': true,
    'ls_features': '',
    'name': 'MS Exchange. резервная копия баз данных',
    'settings': {
        'alerts': {
            'enabled': false,
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
        'max_days_without_backup': 5
    },
    'archive': {
        'comments': '',
        'compression': 'HIGH',
        'display_name': '',
        'format': 'AUTO',
        'name': '[Database name]_[Unique ID]_[Plan ID]A',
        'slice_comments': '',
        'splitting': {
            'auto': true
        }
    },
    'cluster': {
        'strategy': 'REPLICA_FIRST'
    },
    'dag_list': [],
    'destination': {
        'location': {
            'settings': {
                'name': 'customer-dg09',
                'storage_type': 'LOCAL_FOLDER',
                'uri': 'E:/',
                'use_policy_credentials': true
            },
            'type': 'VAULT'
        },
        'performance_window': {
            'enabled': false,
            'presets': [
                {
                    'cpu_priority': 'LOW',
                    'id': 'HIGH',
                    'speed_limit': {
                        'type': 'PERCENT',
                        'value': 100
                    },
                    'timetable': [
                        {
                            'days_of_week': [
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                            'MON',
                            'TUE',
                            'WED',
                            'THU',
                            'FRI',
                            'SAT',
                            'SUN'
                        ],
                        'time_from': {
                            'hour': 0,
                            'minute': 0
                        },
                        'time_to': {
                            'hour': 23,
                            'minute': 59,
                            'second': 59
                        }
                    }
                ]
            },
            {
                'cpu_priority': 'LOW',
                'id': 'LOW',
                'speed_limit': {
                    'type': 'PERCENT',
                    'value': 25
                },
                'timetable': []
            },
            {
                'cpu_priority': 'NORMAL',
                'id': 'CANCEL',
                'speed_limit': {
                    'type': 'PERCENT',
                    'value': 100
                },
                'timetable': []
            }
        ]
    },
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
        'retention': {
            'execution': 'AFTER_BACKUP',
            'rules': [
                {
                    'backup_set': 'DAILY',
                    'max_age': {
                        'count': 7,
                        'type': 'DAYS'
                    }
                },
                {
                    'backup_set': 'WEEKLY',
                    'max_age': {
                        'count': 4,
                        'type': 'WEEKS'
                    }
                }
            ]
        }
    },
    'error_handling': {
        'enabled': true,
        'interval': {
            'count': 30,
            'type': 'SECONDS'
        },
        'max_attempts': 300,
        'silent_mode': true
    },
    'express_backup_usage': {
        'continue_files_tracking': true,
        'dont_create_session': false,
        'fallback_mode': 'AUTO',
        'session_timeout': 0
    },
    'metadata_to_include': 'MAILBOXES',
    'scheduling': {
        'backup_sets': [
            {
                'schedule': {
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                'alarms': {
                    'time': {
                        'rand_max_delay': {
                            'count': 30,
                            'type': 'MINUTES'
                        },
                        'repeat_at': [
                            {
                                'hour': 20,
                                'minute': 0
                            }
                        ],
                        'weekdays': [
                            'MON',
                            'THU',
                            'SAT'
                        ]
                    }
                },
                'prevent_sleep': true,
                'type': 'DAILY'
            },
            'type': 'AUTO'
        }
    ],
    'enabled': true,
    'max_parallel_backups': 2,
    'rand_max_delay': {
        'count': 30,
        'type': 'MINUTES'
    },
    'scheme': 'WEEKLY_FULL_DAILY_INCREMENTAL',
    'task_failure': {
        'enabled': true,
        'interval': {
            'count': 1,
            'type': 'HOURS'
        },
        'max_attempts': 3
    },
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
            'weekly_backup_day': 'MON'
        },
        'tapes': {
            'eject_full_media': false,
            'eject_media_if_successfull': false,
            'multiplexing': {
                'enabled': false,
                'streams_per_drive': 0
            },
            'multistreaming': {
                'enabled': false
            },
            'overwrite_data_on_tape': false,
            'preserve_tapes_position': true,
            'tapeset': ''
        },
        'vss': {
            'enabled': true,
            'provider': 'TARGET_SYSTEM_DEFINED'
        },
        'windows_event_log': {
            'enabled': false,
            'level': 'ERRORS_AND_WARNINGS'
        }
    },
    'tenant_id': 'e721a936-7742-40b5-bf3a-eac627723ae3',
    'type': 'policy.backup.msexchange.db'
}
```

Политика резервного копирования почтовых ящиков Microsoft Exchange Server

Политика обеспечивает функциональность резервного копирования рабочей нагрузки машины целиком.

Дополнительные сведения о параметрах планов резервного копирования см. в документации по продукту.

Описание параметров запроса для создания политики резервного копирования см. в справочнике API.

Следующий пример можно использовать при создании планов защиты с данной политикой:

```
{
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
'enabled': true,
'ls_features': '',
'name': 'MS Exchange. резервная копия ящиков',
'settings': {
    'alerts': {
        'enabled': true,
        'max_days_without_backup': 5
    },
    'archive': {
        'comments': '',
        'compression': 'MAX',
        'display_name': '',
        'format': '12',
        'name': '[Mailbox ID]_mailbox_[Plan ID]A',
        'slice_comments': '',
        'splitting': {
            'auto': true
        }
    },
    'destination': {
        'location': {
            'settings': {
                'name': 'customer-dg09',
                'storage_type': 'LOCAL_FOLDER',
                'uri': 'E:/',
                'use_policy_credentials': true
            },
            'type': 'VAULT'
        },
        'performance_window': {
            'enabled': true,
            'presets': [
                {
                    'cpu_priority': 'LOW',
                    'id': 'HIGH',
                    'speed_limit': {
                        'type': 'PERCENT',
                        'value': 100
                    },
                    'timetable': [
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                    {
                        'days_of_week': [
                            'MON',
                            'SUN'
                        ],
                        'time_from': {
                            'hour': 0,
                            'minute': 0
                        },
                        'time_to': {
                            'hour': 23,
                            'minute': 59,
                            'second': 59
                        }
                    },
                    {
                        'days_of_week': [
                            'TUE',
                            'SAT'
                        ],
                        'time_from': {
                            'hour': 0,
                            'minute': 0
                        },
                        'time_to': {
                            'hour': 18,
                            'minute': 59,
                            'second': 59
                        }
                    }
                ]
            },
            {
                'cpu_priority': 'LOW',
                'id': 'LOW',
                'speed_limit': {
                    'type': 'PERCENT',
                    'value': 25
                },
                'timetable': [
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                    {
                        'days_of_week': [
                            'TUE',
                            'SAT'
                        ],
                        'time_from': {
                            'hour': 19,
                            'minute': 0
                        },
                        'time_to': {
                            'hour': 23,
                            'minute': 59,
                            'second': 59
                        }
                    }
                ]
            },
            {
                'cpu_priority': 'NORMAL',
                'id': 'CANCEL',
                'speed_limit': {
                    'type': 'PERCENT',
                    'value': 100
                },
                'timetable': [
                    {
                        'days_of_week': [
                            'WED',
                            'THU',
                            'FRI'
                        ],
                        'time_from': {
                            'hour': 0,
                            'minute': 0
                        },
                        'time_to': {
                            'hour': 23,
                            'minute': 59,
                            'second': 59
                        }
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                        }
                    ]
                }
            ]
        },
        'retention': {
            'execution': 'AFTER_BACKUP',
            'rules': [
                {
                    'backup_set': 'DAILY',
                    'max_age': {
                        'count': 7,
                        'type': 'DAYS'
                    }
                },
                {
                    'backup_set': 'WEEKLY',
                    'max_age': {
                        'count': 4,
                        'type': 'WEEKS'
                    }
                },
                {
                    'backup_set': 'MONTHLY',
                    'max_age': {
                        'count': 6,
                        'type': 'MONTHS'
                    }
                }
            ]
        }
    },
    'error_handling': {
        'enabled': true,
        'interval': {
            'count': 30,
            'type': 'SECONDS'
        },
        'max_attempts': 10,
        'silent_mode': true
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    },
    'scheduling': {
        'backup_sets': [
            {
                'schedule': {
                    'alarms': {
                        'time': {
                            'repeat_at': [
                                {
                                    'hour': 10,
                                    'minute': 15
                                }
                            ],
                            'weekdays': [
                                'MON',
                                'TUE',
                                'WED',
                                'THU',
                                'FRI'
                            ]
                        }
                    },
                    'prevent_sleep': true,
                    'type': 'WEEKLY'
                },
                'type': 'AUTO'
            }
        ],
        'enabled': true,
        'max_parallel_backups': 8,
        'rand_max_delay': {
            'count': 30,
            'type': 'MINUTES'
        },
        'scheme': 'ALWAYS_INCREMENTAL',
        'task_failure': {
            'enabled': true,
            'interval': {
                'count': 1,
                'type': 'HOURS'
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                },
                'max_attempts': 1
            },
            'weekly_backup_day': 'MON'
        },
        'windows_event_log': {
            'enabled': true,
            'level': 'ERRORS_AND_WARNINGS'
        }
    },
    'tenant_id': 'e721a936-7742-40b5-bf3a-eac627723ae3',
    'type': 'policy.backup.msexchange.doc'
}
```

Создание плана резервного копирования

1. Выполните аутентификацию на платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

   ```
   >>> base_url  # the base URL of the API
   'https://installaddress.ru/api/v1'
   >>> auth  # the 'Authorization' header value with the access token
   {'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
   ```

2. Задайте переменную `plan_data`, а затем присвойте этой переменной объект с ключом `subject`, объект, содержащий ключ `policy`, и массив, содержащий политику резервного копирования:

   ```
   >>> plan_data = {
       'subject': {
           'policy': {
                           'enabled': true,
                           'ls_features': '',
                           'name': 'machines_plan_single_{{id_plan_name}}',
                           'settings': {
                               'alerts': {
                                   'enabled': false,
                                   'max_days_without_backup': 0
                               },
                               'archive': {
   ```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                    'compression': 'NORMAL',
                    'format': 'AUTO',
                    'name': '[Machine Name]-[Plan ID]-[Unique ID]A',
                    'splitting': {
                        'size': 214748364800
                    }
                },
                'cbt': 'DO_NOT_USE',
                'destination': {
                    'location': {
                        'display_name': 'win10x64simple: E:\\tmp\\',
                        'settings': {
                            'credentials': '',
                            'name': 'win10x64simple: E:\\tmp\\',
                            'storage_type': 'LOCAL_FOLDER',
                            'uri': 'E:/tmp/',
                            'use_policy_credentials': true
                        },
                    'performance_window': {
                        'enabled': false
                    },
                    'retention': {
                        'execution': 'AFTER_BACKUP',
                        'rules': [
                            {
                                'backup_set': 'DAILY',
                                'max_age': {
                                    'count': 7,
                                    'type': 'DAYS'
                                }
                            },
                            {
                                'backup_set': 'WEEKLY',
                                'max_age': {
                                    'count': 4,
                                    'type': 'HOURS'
                                }
                            }
                        ]
                    }
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                    },
                    'error_handling': {
                        'enabled': true,
                        'interval': {
                            'count': 30,
                            'type': 'SECONDS'
                        },
                        'max_attempts': 30
                    },
                    'file_filters': {
                        'exclude_hidden': true
                    },
                    'file_level_backup_snapshot': 'CREATE_WHEN_POSSIBLE',
                    'fixed_drive_behavior_for_removable': true,
                    'multi_volume_snapshotting_enabled': true,
                    'notarization': {
                        'enabled': false
                    },
                    'sector_by_sector':true,
                    'preserve_file_security_settings': true,
                    'scheduling': {
                        'backup_sets': [
                            {
                                'schedule': {
                                    'alarms': {
                                        'time': {
                                            'rand_max_delay': {
                                                'count': 30,
                                                'type': 'MINUTES'
                                            },
                                            'repeat_at': [
                                                {
                                                    'hour': 15,
                                                    'minute': 0,
                                                    'second': null
                                                }
                                            ],
                                            'weekdays': [
                                                'MON',
                                                'TUE',
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                                        'WED',
                                        'THU',
                                        'FRI'
                                    ]
                                }
                            },
                            'prevent_sleep': true,
                            'type': 'DAILY'
                        },
                        'type': 'AUTO'
                    }
                ],
                'enabled': true,
                'rand_max_delay': {
                    'count': 30,
                    'type': 'MINUTES'
                },
                'scheme': 'WEEKLY_FULL_DAILY_INCREMENTAL'
            },
            'silent_mode_enabled': true,
            'tapes': {
                'eject_media_if_successfull': false,
                'multiplexing': {
                    'enabled': false,
                    'streams_per_drive': 0
                },
                'multistreaming': {
                    'enabled': false
                },
                'overwrite_data_on_tape': false,
                'preserve_tapes_position': true,
                'tapeset': ''
            },
            'vss': {
                'enabled': true,
                'provider': 'TARGET_SYSTEM_DEFINED'
            }
        },
        'tenant_id': '00000000-0000-0000-0000-000000000000',
        'type': 'policy.backup.machine'
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                    }
        }
```

3. Преобразуйте объект `plan_data` в текст в формате JSON.

```
>>> plan_data = json.dumps(plan_data, indent=4)
```

4. Отправьте запрос POST с текстом в формате JSON на конечную точку `/policies`:

```
>>> response = requests.post(
...     f'{base_url}/policies',
...     headers={'Content-Type': 'application/json', **auth},
...     data=plan_data,
... )
```

5. Проверьте код состояния запроса:

```
>>> response.status_code
200
```

Код состояния HTTP 200 означает, что план защиты создан.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Кроме того, тело ответа содержит объект с ключом `result` в формате JSON, содержащий идентификатор политики. После преобразования в объект он будет выглядеть так:

```
>>> pprint.pprint(response.json())
{'result': ['256f2ee5-c9fd-4aad-8607-97a977deeaae@policy.backup.machine']}
```

Получение перечня планов резервного копирования

1. Выполните аутентификацию на облачной платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Отправьте запрос GET на конечную точку `/policies`:

```
>>> response = requests.get(f'{base_url}/policies', headers=auth)
```

3. Проверьте код состояния запроса:

```
>>> response.status_code
200
```

Код состояния HTTP 200 означает, что запрос был успешно выполнен.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Кроме того, тело ответа содержит объект с ключом `items` в формате JSON, содержащий массив объектов политик защиты. После преобразования в объект он будет выглядеть так:

```
>>> pprint.pprint(response.json())
{'items': [
        {
            'policy': {
                'created_at': '2025-10-07T21:01:10.1136754Z',
                'days_without_backup_alert': null,
                'deleted_at': null,
                'enabled': true,
                'id': 'f5e556c4-27e0-4767-b596-e48fe8d005af',
                'ls_features':
↪'BPCompression|BPRetentionRules|BPValidation|ForensicMode|Notary|diskSource',
                'name': ' ',
                'plan_hash': '00000000-0000-0000-0000-000000000000',
                'settings': {
                    'alerts': {
                        'enabled': false,
                        'max_days_without_backup': 0
                    },
                    'archive': {
                        'compression': 'MAX',
                        'encryption': {
                            'algorithm': 'AES192',
                            'password': 'string'
                        },
                        'format': '12',
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                    'name': '[Machine Name]-[Plan ID]-[Unique ID]A[Plan ID][Plan-
↪name][Unique ID][Virtualization Server Type]F',
                    'splitting': {
                        'size': 214748364800
                    }
                },
                'cbt': 'USE_IF_ENABLED',
                'check_file_timestamp': true,
                'destination': {
                    'location': {
                        'settings': {
                            'credentials': '',
                            'name': 'win10x64simple: E:\\tmp\\',
                            'storage_type': 'LOCAL_FOLDER',
                            'uri': 'E:/tmp/',
                            'use_policy_credentials': true
                        },
                        'type': 'VAULT'
                    },
                    'performance_window': {
                        'enabled': false
                    },
                    'retention': {
                        'execution': 'AFTER_BACKUP',
                        'rules': [
                            {
                                'max_count': 7
                            }
                        ]
                    }
                },
                'error_handling': {
                    'enabled': true,
                    'interval': {
                        'count': 1,
                        'type': 'MINUTES'
                    },
                    'max_attempts': 1
                },
                'exclude_asz': true,
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                'exclude_page_files': true,
                'fast_backup_enabled': true,
                'file_filters': {
                    'exclude_hidden': true
                },
                'fixed_drive_behavior_for_removable': true,
                'forensic': {
                    'dump_running_processes': false,
                    'dump_type': 'RAW_MEMORY_DUMP',
                    'enabled': true
                },
                'ignore_bad_sectors': true,
                'lvm_snapshotting_enabled': true,
                'multi_volume_snapshotting_enabled': true,
                'notarization': {
                    'enabled': true
                },
                'owner_id': '32e95e91-39f4-6053-9ce4-0da7e86c8b81',
                'preserve_file_security_settings': true,
                'reset_free_blocks_list': true,
                'scheduling': {
                    'backup_sets': [
                        {
                            'schedule': {
                                'alarms': {
                                    'time': {
                                        'rand_max_delay': {
                                            'count': 30,
                                            'type': 'MINUTES'
                                        },
                                        'repeat_at': [
                                            {
                                                'hour': 9,
                                                'minute': 0
                                            }
                                        ],
                                        'weekdays': [
                                            'MON',
                                            'TUE',
                                            'WED'
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                                ]
                            }
                        },
                        'conditions': {},
                        'prevent_sleep': true,
                        'type': 'DAILY'
                    },
                    'type': 'AUTO'
                }
            ],
            'enabled': true,
            'rand_max_delay': {
                'count': 30,
                'type': 'MINUTES'
            },
            'scheme': 'WEEKLY_FULL_DAILY_INCREMENTAL',
            'weekly_backup_day': 'MON'
        },
        'sector_by_sector': true,
        'silent_mode_enabled': true,
        'tapes': {
            'devices': null,
            'eject_media_if_successfull': false,
            'multiplexing': {
                'enabled': false,
                'streams_per_drive': 0
            },
            'multistreaming': {
                'enabled': false
            },
            'overwrite_data_on_tape': false,
            'preserve_tapes_position': true,
            'tapeset': ''
        },
        'validate_backup': true,
        'vss': {
            'enabled': true,
            'provider': 'NATIVE'
        }
    },
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                'source_type': 'machines',
                'tenant_id': '00000000-0000-0000-0000-000000000000',
                'type': 'policy.backup.machine',
                'updated_at': '2025-10-07T21:01:10.1131115Z'
            }
        },
        {
            'policy': {
                'created_at': '2025-10-05T00:58:55.1236553Z',
                'days_without_backup_alert': null,
                'deleted_at': null,
                'enabled': true,
                'id': 'cffe3239-102c-40f0-ac9a-f0f5520b3f9e',
                'ls_features':
↪'BPCompression|BPRetentionRules|BPValidation|ForensicMode|Notary|diskSource',
                'name': '89',
                'plan_hash': '00000000-0000-0000-0000-000000000000',
                'settings': {
                    'alerts': {
                        'enabled': false,
                        'max_days_without_backup': 0
                    },
                    'archive': {
                        'compression': 'MAX',
                        'encryption': {
                            'algorithm': 'AES192',
                            'password': 'string'
                        },
                        'format': '12',
                        'name': '[Machine Name]-[Plan ID]-[Unique ID]A[Plan ID][Plan¬
↪name][Unique ID][Virtualization Server Type]F',
                        'splitting': {
                            'size': 214748364800
                        }
                    },
                    'cbt': 'USE_IF_ENABLED',
                    'check_file_timestamp': true,
                    'destination': {
                        'location': {
                            'settings': {
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                'credentials': '',
                'name': 'win10x64simple: E:\\tmp\\',
                'storage_type': 'LOCAL_FOLDER',
                'uri': 'E:/tmp/',
                'use_policy_credentials': true
            },
            'type': 'VAULT'
        },
        'performance_window': {
            'enabled': false
        },
        'retention': {
            'execution': 'AFTER_BACKUP',
            'rules': [
                {
                    'max_count': 7
                }
            ]
        }
    },
    'error_handling': {
        'enabled': true,
        'interval': {
            'count': 1,
            'type': 'MINUTES'
        },
        'max_attempts': 1
    },
    'exclude_asz': true,
    'exclude_page_files': true,
    'fast_backup_enabled': true,
    'file_filters': {
        'exclude_hidden': true
    },
    'fixed_drive_behavior_for_removable': true,
    'forensic': {
        'dump_running_processes': false,
        'dump_type': 'RAW_MEMORY_DUMP',
        'enabled': true
    },
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                'ignore_bad_sectors': true,
                'lvm_snapshotting_enabled': true,
                'multi_volume_snapshotting_enabled': true,
                'notarization': {
                    'enabled': true
                },
                'owner_id': '32e95e91-39f4-6053-9ce4-0da7e86c8b81',
                'preserve_file_security_settings': true,
                'reset_free_blocks_list': true,
                'scheduling': {
                    'backup_sets': [
                        {
                            'schedule': {
                                'alarms': {
                                    'time': {
                                        'rand_max_delay': {
                                            'count': 30,
                                            'type': 'MINUTES'
                                        },
                                        'repeat_at': [
                                            {
                                                'hour': 9,
                                                'minute': 0
                                            }
                                        ],
                                        'weekdays': [
                                            'MON',
                                            'TUE',
                                            'WED'
                                        ]
                                    }
                                },
                                'conditions': {},
                                'prevent_sleep': true,
                                'type': 'DAILY'
                            },
                            'type': 'AUTO'
                        }
                    ],
                    'enabled': true,
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                    'rand_max_delay': {
                        'count': 30,
                        'type': 'MINUTES'
                    },
                    'scheme': 'WEEKLY_FULL_DAILY_INCREMENTAL',
                    'weekly_backup_day': 'MON'
                },
                'sector_by_sector': true,
                'silent_mode_enabled': true,
                'tapes': {
                    'devices': null,
                    'eject_media_if_successfull': false,
                    'multiplexing': {
                        'enabled': false,
                        'streams_per_drive': 0
                    },
                    'multistreaming': {
                        'enabled': false
                    },
                    'overwrite_data_on_tape': false,
                    'preserve_tapes_position': true,
                    'tapeset': ''
                },
                'validate_backup': true,
                'vss': {
                    'enabled': true,
                    'provider': 'NATIVE'
                }
            },
            'source_type': 'machines',
            'tenant_id': '00000000-0000-0000-0000-000000000000',
            'type': 'policy.backup.machine',
            'updated_at': '2025-10-05T00:58:55.1217739Z'
        }
    },
    {
        'policy': {
            'created_at': '2025-10-04T13:32:54.5440459Z',
            'days_without_backup_alert': null,
            'deleted_at': null,
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                'enabled': true,
                'id': '0183b910-0107-4ea8-b6b0-dedd2e2f5f8d',
                'ls_features': 'BPCompression|BPRetentionRules|BPValidation|Notary|fileSource
↪',
                'name': 'rename_12',
                'plan_hash': 'B84AE99C-05BA-39BF-9E40-36AE4FF5FE51',
                'settings': {
                    'alerts': {
                        'enabled': false,
                        'max_days_without_backup': 0
                    },
                    'archive': {
                        'compression': 'MAX',
                        'encryption': {
                            'algorithm': 'AES192',
                            'password': 'string'
                        },
                        'format': '12',
                        'name': '[Machine Name]-[Plan ID]-[Unique ID]A[Plan ID][Plan¬
↪name][Unique ID][Virtualization Server Type]F',
                        'splitting': {
                            'size': 214748364800
                        }
                    },
                    'destination': {
                        'location': {
                            'settings': {
                                'credentials': '',
                                'name': 'win10x64simple: E:\\tmp\\',
                                'storage_type': 'LOCAL_FOLDER',
                                'uri': 'E:/tmp/',
                                'use_policy_credentials': true
                            },
                            'type': 'VAULT'
                        },
                        'performance_window': {
                            'enabled': false
                        },
                        'retention': {
                            'execution': 'AFTER_BACKUP',
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
            'rules': [
                {
                    'max_count': 7
                }
            ]
        }
    },
    'error_handling': {
        'enabled': true,
        'interval': {
            'count': 1,
            'type': 'MINUTES'
        },
        'max_attempts': 1
    },
    'file_filters': {
        'exclude_hidden': true
    },
    'file_level_backup_snapshot': 'CREATE_WHEN_POSSIBLE',
    'fixed_drive_behavior_for_removable': true,
    'ignore_bad_sectors': true,
    'multi_volume_snapshotting_enabled': true,
    'notarization': {
        'enabled': true
    },
    'owner_id': '32e95e91-39f4-6053-9ce4-0da7e86c8b81',
    'preserve_file_security_settings': true,
    'scheduling': {
        'backup_sets': [
            {
                'schedule': {
                    'alarms': {
                        'time': {
                            'rand_max_delay': {
                                'count': 30,
                                'type': 'MINUTES'
                            },
                            'repeat_at': [
                                {
                                    'hour': 9,
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                                        'minute': 0
                                    }
                                ],
                                'weekdays': [
                                    'MON',
                                    'TUE',
                                    'WED'
                                ]
                            }
                        },
                        'conditions': {},
                        'prevent_sleep': true,
                        'type': 'DAILY'
                    },
                    'type': 'AUTO'
                }
            ],
            'enabled': true,
            'rand_max_delay': {
                'count': 30,
                'type': 'MINUTES'
            },
            'scheme': 'WEEKLY_FULL_DAILY_INCREMENTAL',
            'weekly_backup_day': 'MON'
        },
        'selection': {},
        'silent_mode_enabled': true,
        'tapes': {
            'devices': null,
            'eject_media_if_successfull': false,
            'multiplexing': {
                'enabled': false,
                'streams_per_drive': 0
            },
            'multistreaming': {
                'enabled': false
            },
            'overwrite_data_on_tape': false,
            'preserve_tapes_position': true,
            'tapeset': ''
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                    },
                    'validate_backup': true,
                    'vss': {
                        'enabled': true,
                        'provider': 'NATIVE'
                    }
                },
                'source_type': 'files',
                'tenant_id': '00000000-0000-0000-0000-000000000000',
                'type': 'policy.backup.files',
                'updated_at': '2025-10-04T23:11:21.3135901Z'
            }
        }],
 'paging': {'cursors': {'after': ''}}}
```

4. Преобразуйте текст в формате JSON, содержащийся в теле ответа, в объект, а затем извлеките список политик из ответа:

```
>>> policies = response.json()['items']
```

Получение информации о плане резервного копирования

1. Выполните аутентификацию на облачной платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Присвойте переменной `policy_id` UUID политики, найденной с помощью поиска (см. *Получение перечня планов резервного копирования*):

```
>>> policy_id = created_policy_id
>>> policy_id
'dac6c7c4-6b3f-4b46-9aa8-b78a189a0f10'
```

3. Отправьте запрос GET на конечную точку `/policies/<policy_id>`. Для отображения параметров раздела `settings` URL-адрес конечной точки должен содержать параметр запроса

include_settings=true, например:

```
>>> params = {'include_settings': true}
>>> response = requests.get(f'{base_url}/policies/{policy_id}', headers=auth, params=params)
```

4. Проверьте код состояния запроса:

```
>>> response.status_code
200
```

Код состояния HTTP 200 означает, что запрос был успешно выполнен.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Кроме того, тело ответа содержит объект с ключом items в формате JSON, содержащий массив с одним объектом политики защиты. После преобразования в объект он будет выглядеть так:

```
>>> pprint.pprint(response.json())
{'items': [
        {
            'policy': {
                'created_at': '2025-10-07T21:01:10.1136754Z',
                'days_without_backup_alert': null,
                'deleted_at': null,
                'enabled': true,
                'id': 'f5e556c4-27e0-4767-b596-e48fe8d005af',
                'ls_features':
↪'BPCompression|BPRetentionRules|BPValidation|ForensicMode|Notary|diskSource',
                'name': ' ',
                'plan_hash': '00000000-0000-0000-0000-000000000000',
                'settings': {
                    'alerts': {
                        'enabled': false,
                        'max_days_without_backup': 0
                    },
                    'archive': {
                        'compression': 'MAX',
                        'encryption': {
                            'algorithm': 'AES192',
                            'password': 'string'
                        },
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                        'format': '12',
                        'name': '[Machine Name]-[Plan ID]-[Unique ID]A[Plan ID][Plan¬
↪name][Unique ID][Virtualization Server Type]F',
                        'splitting': {
                            'size': 214748364800
                        }
                    },
                    'cbt': 'USE_IF_ENABLED',
                    'check_file_timestamp': true,
                    'destination': {
                        'location': {
                            'settings': {
                                'credentials': '',
                                'name': 'win10x64simple: E:\\tmp\\',
                                'storage_type': 'LOCAL_FOLDER',
                                'uri': 'E:/tmp/',
                                'use_policy_credentials': true
                            },
                            'type': 'VAULT'
                        },
                        'performance_window': {
                            'enabled': false
                        },
                        'retention': {
                            'execution': 'AFTER_BACKUP',
                            'rules': [
                                {
                                    'max_count': 7
                                }
                            ]
                        }
                    },
                    'error_handling': {
                        'enabled': true,
                        'interval': {
                            'count': 1,
                            'type': 'MINUTES'
                        },
                        'max_attempts': 1
                    },
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                'exclude_asz': true,
                'exclude_page_files': true,
                'fast_backup_enabled': true,
                'file_filters': {
                    'exclude_hidden': true
                },
                'fixed_drive_behavior_for_removable': true,
                'forensic': {
                    'dump_running_processes': false,
                    'dump_type': 'RAW_MEMORY_DUMP',
                    'enabled': true
                },
                'ignore_bad_sectors': true,
                'lvm_snapshotting_enabled': true,
                'multi_volume_snapshotting_enabled': true,
                'notarization': {
                    'enabled': true
                },
                'owner_id': '32e95e91-39f4-6053-9ce4-0da7e86c8b81',
                'preserve_file_security_settings': true,
                'reset_free_blocks_list': true,
                'scheduling': {
                    'backup_sets': [
                        {
                            'schedule': {
                                'alarms': {
                                    'time': {
                                        'rand_max_delay': {
                                            'count': 30,
                                            'type': 'MINUTES'
                                        },
                                        'repeat_at': [
                                            {
                                                'hour': 9,
                                                'minute': 0
                                            }
                                        ],
                                        'weekdays': [
                                            'MON',
                                            'TUE',
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                                    'WED'
                                ]
                            }
                        },
                        'conditions': {},
                        'prevent_sleep': true,
                        'type': 'DAILY'
                    },
                    'type': 'AUTO'
                }
            ],
            'enabled': true,
            'rand_max_delay': {
                'count': 30,
                'type': 'MINUTES'
            },
            'scheme': 'WEEKLY_FULL_DAILY_INCREMENTAL',
            'weekly_backup_day': 'MON'
        },
        'sector_by_sector': true,
        'silent_mode_enabled': true,
        'tapes': {
            'devices': null,
            'eject_media_if_successfull': false,
            'multiplexing': {
                'enabled': false,
                'streams_per_drive': 0
            },
            'multistreaming': {
                'enabled': false
            },
            'overwrite_data_on_tape': false,
            'preserve_tapes_position': true,
            'tapeset': ''
        },
        'validate_backup': true,
        'vss': {
            'enabled': true,
            'provider': 'NATIVE'
        }
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
                },
                'source_type': 'machines',
                'tenant_id': '00000000-0000-0000-0000-000000000000',
                'type': 'policy.backup.machine',
                'updated_at': '2025-10-07T21:01:10.1131115Z'
            }
        }
],
 'paging': {'cursors': {'after': ''}}}
```

5. Преобразуйте текст в формате JSON, содержащийся в теле ответа, в объект, а затем извлеките данные о политике из ответа:

```
>>> policies = response.json()['items']
```

Редактирование параметров плана резервного копирования

1. Выполните аутентификацию на облачной платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Запросите политики, как описано в разделе *Получение перечня планов резервного копирования*. Задайте переменную `policy_id` и присвойте ей в качестве значения идентификатор политики. В данном примере использован идентификатор первой политики:

```
>>> policy_id = policies[0]['id']
>>> policy_id
33965f81-7293-45d4-9f13-dc4281d7bdfd
```

3. Задайте переменную `policy_update`, а затем присвойте этой переменной объект с ключом `subject`, содержащий параметры:

```
>>> policy_update = {
...     ”subject”: {
...         ”policy”: [
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
...             {
...                "settings": {
...                  "alerts": {
...                     "enabled": true,
...                     "max_days_without_backup": 5 }
...                }
...             }
...         ]
...     }
... }
```

> **Примечание**
>
> Список доступных параметров политики приведен в справочнике API.

4. Преобразуйте объект `policy_update` в текст в формате JSON.

```
>>> policy_update = json.dumps(policy_update, indent=4)
```

5. Отправьте запрос PATCH с текстом в формате JSON на конечную точку `/policies/{policy_id}`:

```
>>> response = requests.patch(
...     f'{base_url}/policies/{policy_id}',
...     headers={'Content-Type': 'application/json', **auth},
...     data=policy_update,
... )
```

6. Проверьте код состояния запроса:

```
>>> response.status_code
204
```

Код состояния HTTP 204 означает, что план был успешно обновлен.

Код состояния HTTP 200 означает, что план был обновлен с предупреждениями.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Включение/выключение плана резервного копирования

1. Выполните аутентификацию на облачной платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Запросите политики, как описано в разделе *Получение перечня планов резервного копирования*. Задайте переменную `policy_id` и присвойте ей в качестве значения идентификатор политики. В данном примере использован идентификатор первой политики:

```
>>> policy_id = policies[0]['id']
>>> policy_id
33965f81-7293-45d4-9f13-dc4281d7bdfd
```

3. Задайте переменную `policy_enabled`, а затем присвойте этой переменной объект:

```
>>> policy_enabled = {
...   "policy_ids": [{policy_id}],
...   "value": true
... }
```

> **Примечание**
>
> Для выключения плана требуется передать `"value": false`.
>
> Для включения или выключения плана на определённых устройствах требуется передать `"context_ids": [{context_id}, ...]`.

4. Преобразуйте объект `policy_enabled` в текст в формате JSON.

```
>>> policy_enabled = json.dumps(policy_enabled, indent=4)
```

5. Отправьте запрос PATCH с текстом в формате JSON на конечную точку `/policy_applications/enabled`:

```
>>> response = requests.patch(
...     f'{base_url}/policy_applications/enabled',
...     headers={'Content-Type': 'application/json', **auth},
...     data=policy_enabled,
... )
```

6. Проверьте код состояния запроса:

```
>>> response.status_code
204
```

Код состояния HTTP 204 означает, что план был успешно обновлен.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Удаление плана резервного копирования

1. Выполните аутентификацию на облачной платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Запросите планы защиты, как описано в разделе *Получение перечня планов резервного копирования*. Задайте переменную `policy_id` и присвойте ей в качестве значения идентификатор плана защиты. В данном примере использован идентификатор первого плана защиты:

```
>>> policy_id = policies[0]['id']
>>> policy_id
5b15f6e1-88ec-4dce-b523-0e8394c0bc19
```

3. Отправьте запрос DELETE на конечную точку `/policies/{policy_id}`:

```
>>> response = requests.delete(f'{base_url}/policies/{policy_id}', headers=auth)
```

4. Проверьте код состояния запроса:

```
>>> response.status_code
204
```

Код состояния HTTP 204 означает, что план защиты был успешно удален.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Применение политики

Применение политики (плана резервного копирования) предоставляет информацию о следующих статусах:

- статус развёртывания политики;
- статус выполнения политики;
- статус процесса применения политики.

Все операции с применением политик находятся в конечной точке `/policy_applications`.

Структура объекта JSON применения политики

Описание структуры объекта JSON применения политики доступно по ссылке.

Справочники

Справочники доступны по ссылке.

Действия с применениями политик

Операция	Используемые методы и конечные точки
Получение списка применений плана резервного копирования	`GET /policy_applications`
Запуск плана резервного копирования	`POST /policy_applications:run`

Получение списка применений плана резервного копирования

1. Выполните аутентификацию на облачной платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Запросите ресурсы, как описано в разделе *Получение списка ресурсов*. Задайте переменную `resource_id` и присвойте ей в качестве значения идентификатор ресурса. В данном примере использован идентификатор первого ресурса:

```
>>> resource_id = resources[0]['id']
>>> resource_id
'b19cf0fe-95eb-4657-a743-9d4ce6bc34f6'
```

3. Задайте переменную `filters`, а затем присвойте объекту, содержащему идентификатор ресурса в ключе `context_id`, эту переменную:

```
>>> filters = {
...     'context_id': resource_id
... }
```

> **Примечание**
>
> Чтобы получить доступ к планам, применимым к нескольким ресурсам, укажите идентификаторы ресурсов в следующем формате: `or(5b15f6e1-88ec-4dce-b523-0e8394c0bc19,c70134c4-a244-4b22-99ad-e081301f7530)`.

4. Отправьте запрос GET на конечную точку `/policy_applications`:

```
>>> response = requests.get(f'{base_url}/policy_applications', headers=auth, params=filters)
```

5. Проверьте код состояния запроса:

```
>>> response.status_code
200
```

Код состояния HTTP 200 означает, что запрос был успешно выполнен.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Кроме того, тело ответа содержит объект с ключом `items` в формате JSON, содержащий массив

объектов политик защиты. После преобразования в объект он будет выглядеть так:

```
>>> pprint.pprint(response.json())
{
    'items': [
        [
            {
                'agent_id': '0d47f411-3b67-42c8-b9d5-33a48b2bbe82',
                'context': {
                    'id': 'b19cf0fe-95eb-4657-a743-9d4ce6bc34f6',
                    'type': 'resource.machine'
                },
                'created_at': '2025-10-29T11:13:51.1782663Z',
                'deleted_at': null,
                'deployment': {
                    'state': 'DEPLOYED'
                },
                'enabled': true,
                'id': '7d61b4b9-8f39-42ad-824c-d508a6ccbf59',
                'last_activity': 'policy.backup.machine',
                'next_activity': 'policy.backup.machine',
                'next_run_time': '2025-10-29T20:00:00Z',
                'origin_contexts': [
                    'b19cf0fe-95eb-4657-a743-9d4ce6bc34f6'
                ],
                'policy': {
                    'id': '5ea704d9-a214-4e81-97d9-d228098f8431',
                    'name': 'Новый план защиты',
                    'type': 'policy.backup.machine'
                },
                'running': {
                    'state': 'IDLE'
                },
                'status': 'OK',
                'tenant_id': '00000000-0000-0000-0000-000000000000',
                'updated_at': '2025-10-29T11:13:51.7559748Z'
            }
        ]
    ],
    'paging': {
        'cursors': {
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
            'total': 1
        }
    }
}
```

Применение ранее созданного плана резервного копирования к ресурсу

1. Выполните аутентификацию на платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

   ```
   >>> base_url  # the base URL of the API
   'https://installaddress.ru/api/v1'
   >>> auth  # the 'Authorization' header value with the access token
   {'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
   ```

2. Запросите планы резервного копирования, как описано в разделе *Получение перечня планов резервного копирования*. Задайте переменную `policy_id` и присвойте ей в качестве значения идентификатор плана резервного копирования. В данном примере использован идентификатор первого плана резервного копирования:

   ```
   >>> policy_id = policies[0]['id']
   >>> policy_id
   5b15f6e1-88ec-4dce-b523-0e8394c0bc19
   ```

3. Запросите ресурсы, как описано в разделе *Получение списка ресурсов*. Задайте переменную `resource_id` и присвойте ей в качестве значения идентификатор ресурса. В данном примере использован идентификатор первого ресурса:

   ```
   >>> resource_id = resources[0]['id']
   >>> resource_id
   '5c350066-2ba6-4eeb-aa91-1213dd35f033'
   ```

4. Задайте переменную `application_data`, а затем назначьте объекту, содержащему ключ `policy_id`, ID плана резервного копирования, а объекту `context` — ключ `items`, содержащий список идентификаторов ресурсов, для этой переменной:

   ```
   >>> application_data = {
   ...     'policy_id': policy_id,
   ```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
...     'context': {
...         'items': [
...             resource_id
...         ]
...     }
... }
```

5. Преобразуйте объект `application_data` в текст в формате JSON.

```
>>> application_data = json.dumps(application_data, indent=4)
```

6. Отправьте запрос POST с текстом в формате JSON на конечную точку `/policy_applications`:

```
>>> response = requests.post(
...     f'{base_url}/policy_applications',
...     headers={'Content-Type': 'application/json', **auth},
...     data=application_data,
... )
```

7. Проверьте код состояния запроса:

```
>>> response.status_code
204
```

Код состояния HTTP 204 означает, что план резервного копирования был применен к выбранным ресурсам.

Код состояния HTTP 200 означает, что план резервного копирования был применен к выбранным ресурсам с некритическими предупреждениями.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Тело ответа с кодом состояния HTTP 200 содержит объект с ключом `issues` в формате JSON, содержащий массив с предупреждениями. После преобразования в объект он будет выглядеть так:

```
>>> pprint.pprint(response.json())
{'issues': [...]}
```

Отзыв ранее созданного плана резервного копирования с ресурса

1. Выполните аутентификацию на облачной платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Запросите планы резервного копирования, как описано в разделе *Получение перечня планов резервного копирования*. Задайте переменную `policy_id` и присвойте ей в качестве значения идентификатор плана резервного копирования. В данном примере использован идентификатор первого плана резервного копирования:

```
>>> policy_id = policies[0]['id']
>>> policy_id
5b15f6e1-88ec-4dce-b523-0e8394c0bc19
```

3. Запросите ресурсы, как описано в разделе *Получение списка ресурсов*. Задайте переменную `resource_id` и присвойте ей в качестве значения идентификатор ресурса. В данном примере использован идентификатор первого ресурса:

```
>>> resource_id = resources[0]['id']
>>> resource_id
'5c350066-2ba6-4eeb-aa91-1213dd35f033'
```

4. Задайте переменную `filters`, а затем присвойте этой переменной объект с ключом `policy_id`, содержащим ID политики, и ключом `context`, содержащим ID ресурса:

```
>>> filters = {
...     'policy_id': policy_id,
...     'context_id': resource_id
... }
```

> **Примечание**
>
> Чтобы отозвать планы из нескольких ресурсов, укажите значения в следующем формате: `or(33965f81-7293-45d4-9f13-dc4281d7bdfd,c70134c4-a244-4b22-99ad-e081301f7530)`.

Список доступных параметров запроса приведен в справочнике API.

5. Отправьте запрос DELETE на конечную точку `/policy_applications`:

```
>>> response = requests.delete(
...     f'{base_url}/policy_applications',
...     headers=auth,
...     params=filters,
... )
```

6. Проверьте код состояния запроса:

```
>>> response.status_code
204
```

Код состояния HTTP 204 означает, что план резервного копирования был успешно отозван с выбранных ресурсов.

Код состояния HTTP 200 означает, что план резервного копирования был отозван с выбранных ресурсов с некритическими предупреждениями.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Тело ответа с кодом состояния HTTP 200 содержит объект с ключом `issues` в формате JSON, содержащий массив с предупреждениями. После преобразования в объект он будет выглядеть так:

```
>>> pprint.pprint(response.json())
{'issues': [...]}
```

Запуск плана резервного копирования

1. Выполните аутентификацию на платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Запросите планы резервного копирования, как описано в разделе *Получение перечня планов*

резервного копирования. Задайте переменную `policy_id` и присвойте ей в качестве значения идентификатор плана резервного копирования. В данном примере использован идентификатор первого плана резервного копирования:

```
>>> policy_id = policies[0]['id']
>>> policy_id
33965f81-7293-45d4-9f13-dc4281d7bdfd
```

3. [Необязательно] Запросите ресурсы, как описано в разделе *Получение списка ресурсов*. Задайте переменную `resource_id` и присвойте ей в качестве значения идентификатор ресурса. В данном примере использован идентификатор первого ресурса:

```
>>> resource_id = resources[0]['id']
>>> resource_id
'5c350066-2ba6-4eeb-aa91-1213dd35f033'
```

> **Примечание**
>
> Если в запросе не указаны ресурсы, политика будет выполнена для всех ресурсов, к которым она применяется.

4. Задайте переменную `policy_exec_data`, а затем присвойте этой переменной объект, содержащий следующие параметры JSON:

```
>>> policy_exec_data = {
...     'state': 'RUNNING',
...     'policy_id': policy_id,
...     'context_ids': [resource_id]
... }
```

> **Примечание**
>
> При передаче `'state': 'IDLE'` будет выполнена остановка плана резервного копирования.
>
> Список доступных параметров запроса приведен в справочнике API.

5. Преобразуйте объект `policy_exec_data` в текст в формате JSON:

```
>>> policy_exec_data = json.dumps(policy_exec_data, indent=4)
```

6. Отправьте запрос POST с текстом в формате JSON на конечную точку `/policy_applications:run`:

```
>>> response = requests.post(
...     f'{base_url}/policy_applications:run',
...     headers={'Content-Type': 'application/json', **auth},
...     data=policy_exec_data,
... )
```

7. Проверьте код состояния запроса:

```
>>> response.status_code
202
```

   Код состояния HTTP 202 означает, что запрос был принят на выполнение.

   Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Реквизиты для входа

Реквизиты для входа — это учётные данные (имена учетных записей для входа, пароли, ключи доступа) для обеспечения доступа. Применяются для выполнения операций над объектами, требующими авторизации при доступе.

Типы реквизитов для входа

Типы	Подтипы
account	Учётные записи для авторизации. Используемые подтипы: • computer • esx • vcenter • scale • ovirt • ms_exchange • vm_guest_system • ldap
archive	Учётные данные архивов. Используемые подтипы: • endpoint • ' '
share	Учётные данные сетевых папок. Используемые подтипы: • smb • nfs • sftp
asz	Учётные данные Зоны безопасности. Подтипы отсутствуют.
db	Учётные данные баз данных. Используемые подтипы: • mysql • mssql
folder	Учётные данные Локальных папок. Подтипы отсутствуют.

Все операции с реквизитами для входа находятся в конечной точке `/credentials`.

Структура объекта JSON реквизитов для входа

Описание структуры объекта JSON реквизитов для входа доступно по ссылке.

Справочники

Справочники доступны по ссылке.

Действия с реквизитами для входа

Операция	Используемые методы и конечные точки
Создание реквизитов для входа	`POST /credentials`
Получение информации о реквизитах для входа	`GET /credentials/{credentials_id}`
Удаление реквизитов для входа	`DELETE /credentials/{credentials_id}`

Создание реквизитов для входа

1. Выполните аутентификацию на платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Задайте переменную `cred_data`, а затем присвойте этой переменной объект:

```
>>> cred_data = {
    'scope': {
        'type': 'folder'
        },
    'description': 'новое описание',
    'identity': 'user_name',
    'secret': 'user_pass'
    }
```

3. Преобразуйте объект `cred_data` в текст в формате JSON.

```
>>> cred_data = json.dumps(cred_data, indent=4)
```

4. Отправьте запрос POST с текстом в формате JSON на конечную точку `/credentials`:

```
>>> response = requests.post(
...     f'{base_url}/credentials',
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
...     headers={'Content-Type': 'application/json', **auth},
...     data=cred_data,
... )
```

5. Проверьте код состояния запроса:

```
>>> response.status_code
201
```

Код состояния HTTP 201 означает, что реквизиты для входа созданы.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Кроме того, тело ответа содержит объект с ключом `id` в формате JSON, содержащий идентификатор реквизитов для входа. После преобразования в объект он будет выглядеть так:

```
>>> pprint.pprint(response.json())
{
    'id': 'b12e9914-ecea-4869-8e55-3e62a6dd8bb8'
}
```

Получение информации о реквизитах для входа

1. Выполните аутентификацию на платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Присвойте переменной `credentials_id` UUID реквизитов для входа, созданных с помощью API (см. *Создание реквизитов для входа*):

```
>>> credentials_id = created_credentials_id
>>> credentials_id
'b12e9914-ecea-4869-8e55-3e62a6dd8bb8'
```

3. Отправьте запрос GET на конечную точку `/credentials/<credentials_id>`:

```
>>> response = requests.get(f'{base_url}/credentials/{credentials_id}', headers=auth)
```

4. Проверьте код состояния запроса:

```
>>> response.status_code
200
```

Код состояния HTTP 200 означает, что запрос был успешно выполнен.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Кроме того, тело ответа содержит объект в формате JSON, содержащий параметры реквизитов для входа. После преобразования в объект он будет выглядеть так:

```
>>> pprint.pprint(response.json())
{
    'scope': {
        'type': 'folder',
        'subtype': ''
    },
    'description': 'новое описание',
    'identity': 'user_name',
    'id': 'b12e9914-ecea-4869-8e55-3e62a6dd8bb8',
    'tenant': '8f2ce031-3a88-448a-b6b6-84fc8728eab4',
    'version': 0,
    'secret_autogenerated': true,
    'created_at': '2025-11-05T14:15:50Z',
    'updated_at': '2025-11-05T14:15:50Z'
}
```

Удаление реквизитов для входа

1. Выполните аутентификацию на платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Присвойте переменной `credentials_id` UUID реквизитов для входа, созданных с помощью API (см. *Создание реквизитов для входа*):

```
>>> credentials_id = created_credentials_id
>>> credentials_id
'b12e9914-ecea-4869-8e55-3e62a6dd8bb8'
```

3. Отправьте запрос DELETE на конечную точку `/credentials/{credentials_id}`:

```
>>> response = requests.delete(f'{base_url}/credentials/{credentials_id}', headers=auth)
```

4. Проверьте код состояния запроса:

```
>>> response.status_code
204
```

   Код состояния HTTP 204 означает, что план защиты был успешно удален.

   Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Архив резервных копий

Архив — файл, содержащий набор резервных копий (точек востановления) и метаданные для них. Файл содержит цепочку резервных копий: полную или дифференциальную резервную копию и все зависящие от неё инкрементные копии. Архив формируется при создании хотя бы одной резервной копии. В дальнейшем он используется для хранения данных при резервном копировании и восстановлении. В архиве применяется сжатие и дедупликация данных.

Все операции с архивами находятся в конечной точке `/archives`.

Структура объекта JSON архива

Описание структуры объекта JSON архива доступно по ссылке.

Справочники

Справочники доступны по ссылке.

Действия с архивами

Операция	Используемые методы и конечные точки
Просмотр архивов резервных копий	`GET /archives`
Удаление архива резервных копий	`DELETE /archives`

Просмотр архивов резервных копий

1. Выполните аутентификацию на платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

   ```
   >>> base_url  # the base URL of the API
   'https://installaddress.ru/api/v1'
   >>> auth  # the 'Authorization' header value with the access token
   {'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
   ```

2. Отправьте запрос GET на конечную точку `/archives`:

   ```
   >>> response = requests.get(f'{base_url}/archives', headers=auth)
   ```

   Примечание

   Список доступных параметров строки запроса приведен в справочнике API.

3. Проверьте код состояния запроса:

   ```
   >>> response.status_code
   200
   ```

   Код состояния HTTP 200 означает, что запрос был успешно выполнен.

   Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

   Кроме того, тело ответа содержит массив `items` в формате JSON, содержащий данные архивов. После преобразования в объект он будет выглядеть так:

```
>>> pprint.pprint(response.json())
{
    'items': [
        {
            'id': '5ab0433d-991a-3d0f-1328-4b1874d621a3',
            'name': 'AAURxWO@odio_mock_200_account_workmail_A630C04C-14F1-32BA-9585-
↪2C9472EF3E00_edd0ec02-67df-4017-abd8-ee37611dbeb5A',
            'vault_id': '21882913-cc56-646e-1036-9eac88c3977a',
            'size': 20480,
            'compressed_data_size': 0,
            'data_size': 0,
            'original_data_size': 0,
            'logical_size': 0,
            'type': 'EMPTY',
            'format': '12',
            'created_at': '1970-01-01T00:00:00Z',
            'updated_at': '0001-01-01T00:00:00Z',
            'last_backup_created_at': '1970-01-01T00:00:00Z',
            'protected_by_password': false,
            'encryption_algorithm': 'NONE',
            'deleted': false,
            'file_name': '',
            'tenant_id': 'f9cf6b51-90c8-43bb-8da5-1d6018de265e',
            'agent_id': '00000000-0000-0000-0000-000000000000',
            'agent_name': '',
            'description': '',
            'display_name': 'AAURxWO@odio_mock_200_account_workmail_A630C04C-14F1-32BA-9585-
↪2C9472EF3E00_edd0ec02-67df-4017-abd8-ee37611dbeb5A',
            'owner_id': '',
            'owner_name': '',
            'resource_id': '',
            'consistent': false,
            'marked_for_deletion': false,
            'last_seen_at': '2025-09-29T15:59:09Z',
            'vault': {
                'id': '21882913-cc56-646e-1036-9eac88c3977a',
                'name': 'pk-customer',
                'storage_type': 'ONLINE'
            }
        },
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
        {
            'id': 'b9ac73cc-e091-d511-2205-965b46a24e71',
            'name': 'AbouXIf@odio_mock_200_account_workmail_C9DABFE4-02E7-34B1-8FF0-
↪239E0E69DD84_edd0ec02-67df-4017-abd8-ee37611dbeb5A',
            'vault_id': '21882913-cc56-646e-1036-9eac88c3977a',
            'size': 20480,
            'compressed_data_size': 0,
            'data_size': 0,
            'original_data_size': 0,
            'logical_size': 0,
            'type': 'EMPTY',
            'format': '12',
            'created_at': '1970-01-01T00:00:00Z',
            'updated_at': '0001-01-01T00:00:00Z',
            'last_backup_created_at': '1970-01-01T00:00:00Z',
            'protected_by_password': false,
            'encryption_algorithm': 'NONE',
            'deleted': false,
            'file_name': '',
            'tenant_id': 'f9cf6b51-90c8-43bb-8da5-1d6018de265e',
            'agent_id': '00000000-0000-0000-0000-000000000000',
            'agent_name': '',
            'description': '',
            'display_name': 'AbouXIf@odio_mock_200_account_workmail_C9DABFE4-02E7-34B1-8FF0-
↪239E0E69DD84_edd0ec02-67df-4017-abd8-ee37611dbeb5A',
            'owner_id': '',
            'owner_name': '',
            'resource_id': '',
            'consistent': false,
            'marked_for_deletion': false,
            'last_seen_at': '2025-09-29T15:59:09Z',
            'vault': {
                'id': '21882913-cc56-646e-1036-9eac88c3977a',
                'name': 'pk-customer',
                'storage_type': 'ONLINE'
            }
        },
        {
            'id': 'c60019f2-98ce-b1ec-af1a-0db0fb2644e2',
            'name': 'aajsSxM@odio_mock_200_account_workmail_67028231-BFC9-352E-A3D9-
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
↪0D934164DDC3_edd0ec02-67df-4017-abd8-ee37611dbeb5A',
            'vault_id': '21882913-cc56-646e-1036-9eac88c3977a',
            'size': 20480,
            'compressed_data_size': 0,
            'data_size': 0,
            'original_data_size': 0,
            'logical_size': 0,
            'type': 'EMPTY',
            'format': '12',
            'created_at': '1970-01-01T00:00:00Z',
            'updated_at': '0001-01-01T00:00:00Z',
            'last_backup_created_at': '1970-01-01T00:00:00Z',
            'protected_by_password': false,
            'encryption_algorithm': 'NONE',
            'deleted': false,
            'file_name': '',
            'tenant_id': 'f9cf6b51-90c8-43bb-8da5-1d6018de265e',
            'agent_id': '00000000-0000-0000-0000-000000000000',
            'agent_name': '',
            'description': '',
            'display_name': 'aajsSxM@odio_mock_200_account_workmail_67028231-BFC9-352E-A3D9-
↪0D934164DDC3_edd0ec02-67df-4017-abd8-ee37611dbeb5A',
            'owner_id': '',
            'owner_name': '',
            'resource_id': '',
            'consistent': false,
            'marked_for_deletion': false,
            'last_seen_at': '2025-09-29T15:59:09Z',
            'vault': {
                'id': '21882913-cc56-646e-1036-9eac88c3977a',
                'name': 'pk-customer',
                'storage_type': 'ONLINE'
            }
        }
    ]
}
```

4. Преобразуйте текст в формате JSON, содержащийся в теле ответа, в объект, а затем извлеките данные об архивах из ответа:

```
>>> archives = response.json()['items']
```

Удаление архива резервных копий

1. Выполните аутентификацию на платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

   ```
   >>> base_url  # the base URL of the API
   'https://installaddress.ru/api/v1'
   >>> auth  # the 'Authorization' header value with the access token
   {'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
   ```

2. Запросите архивы резервных копий, как описано в разделе *Просмотр архивов резервных копий*. Задайте переменную `archive_id` и присвойте ей в качестве значения идентификатор архива резервной копии. В данном примере использован идентификатор первого архива резервной копии:

   ```
   >>> archive_id = archives[0]['id']
   >>> archive_id
   21882913-cc56-646e-1036-9eac88c3977a
   ```

3. Задайте переменную `vault_id` и присвойте ей в качестве значения идентификатор хранилища выбранного архива резервной копии:

   ```
   >>> vault_id = archives[0]['vault_id']
   >>> vault_id
   5f51ad20-f489-2bf7-1340-f1dfd1c08e93
   ```

4. Задайте переменную `resource_id` и присвойте ей в качестве значения `id` ресурса, архив резервных копий для которого нужно удалить:

   ```
   >>> resource_id = archives[0]['resource_ids'][0]
   ```

5. Запросите ресурсы, как описано в разделе *Получение списка ресурсов*. Задайте переменную `machine_id` и присвойте ей в качестве значения `id` агента. В примере использован `id` агента ресурса, резервную копию для которого нужно удалить:

   ```
   >>> machine_id = next(
       (item['agent_id'] for item in data['items'] if item['id'] == 'resource_id'),
       None
   ```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
)
>>> machine_id
122da62d-ff99-4fa7-b3cd-2580ee831a31
```

6. Отправьте запрос DELETE на конечную точку `/archives`. URL-адрес конечной точки должен содержать параметры запроса `archive_id`, `vault_id` и `machine_id`.

```
>>> params = {'archive_id': archive_id, 'vault_id': vault_id, 'machine_id': machine_id}
>>> response = requests.delete(f'{base_url}/backups', headers=auth, params=params)
```

7. Проверьте код состояния запроса:

```
>>> response.status_code
200
```

   Код состояния HTTP 200 означает, что запрос был успешно выполнен.

   Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

   Кроме того, тело ответа содержит массив `items` в формате JSON, содержащий список активностей удаления архивов резервных копий. После преобразования в объект он будет выглядеть так:

```
>>> pprint.pprint(response.json())
{
    'items': [
      ”752b761e-fa88-477f-9031-cc3f03816083”
    ]
}
```

8. Преобразуйте текст в формате JSON, содержащийся в теле ответа, в объект, а затем извлеките список задач из ответа:

```
>>> activities = response.json()['items']
```

Резервная копия

Резервная копия (точка востановления) — это сохранённые данные ресурса и метаданные для них. Резервная копия создаётся при операциях резервного копирования. Данные резервной копии используются в операциях восстановления.

Все операции с резервными копиями находятся в конечной точке `/backups`.

Структура объекта JSON резервной копии

Описание структуры объекта JSON резервной копии доступно по ссылке.

Справочники

Справочники доступны по ссылке.

Действия с резервными копиями

Операция	Используемые методы и конечные точки
Просмотр резервных копий	`GET /backups`
Получить количество хранимых резервных копий в облачном хранилище возможно с помощью операции *Просмотр резервных копий*.	`GET /backups`
Удаление резервных копий	`DELETE /backups`

Просмотр резервных копий

1. Выполните аутентификацию на платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

   ```
   >>> base_url  # the base URL of the API
   'https://installaddress.ru/api/v1'
   >>> auth  # the 'Authorization' header value with the access token
   {'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
   ```

2. [Необязательно] Задайте переменную `filters`

   ```
   >>> filters = {
   ...     'count': true
   ... }
   ```

   Примечание

   Для получения количества хранимых резервных копий, размещенных в облачном хранилище, укажите в запросе параметр `count = true`. Количество резервных копий будет возвращено в атрибуте `total_count`.

   Список доступных параметров строки запроса приведен в справочнике API.

3. Отправьте запрос GET на конечную точку `/backups`:

```
>>> response = requests.get(f'{base_url}/backups', headers=auth, params=filters)
```

4. Проверьте код состояния запроса:

```
>>> response.status_code
200
```

Код состояния HTTP 200 означает, что запрос был успешно выполнен.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Кроме того, тело ответа содержит массив `items` в формате JSON, содержащий данные резервных копий. После преобразования в объект он будет выглядеть так:

```
>>> pprint.pprint(response.json())
{
    'items': [
        {
            'backup_id': 'b79a69d6-3baf-44d1-b367-5a9038e258f2',
            'vault_id': '21882913-cc56-646e-1036-9eac88c3977a',
            'archive_id': '5b342641-8f24-a094-29f1-97ff33745e42',
            'legacy_id': 'avfs:/online?account%3dpk-customer%26provider%3dCyberProtect#arl:/
↪00000000-0000-0000-0000-000000000000/00000000-0000-0000-0000-000000000000/5B342641-8F24-
↪A094-29F1-97FF33745E42/B79A69D6-3BAF-44D1-B367-5A9038E258F2?archive_name%3dAALToOq@odio_
↪mock_200_account_workmail_E68393C6-E69E-3765-815F-E056D6A6F50E_edd0ec02-67df-4017-abd8-
↪ee37611dbeb5A',
            'created_at': '2025-09-29T16:02:45Z',
            'size': 24576,
            'deduplicated_size': 0,
            'backed_up_data_size': 0,
            'original_data_size': 38934,
            'tenant_id': 'f9cf6b51-90c8-43bb-8da5-1d6018de265e',
            'resource_ids': [
                'E68393C6-E69E-3765-815F-E056D6A6F50E'
            ],
            'type': 'FULL',
            'marked_for_deletion': false,
            'last_seen_at': '2025-09-29T16:02:45Z',
            'validation_status': null,
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
            'created_in_network_isolation': false,
            'deleted': false
        },
        {
            'backup_id': '2382810e-8197-47ee-85dc-ba3ef28eb98c',
            'vault_id': '21882913-cc56-646e-1036-9eac88c3977a',
            'archive_id': '86150517-97b7-c2e2-0413-f137b16ae87f',
            'legacy_id': 'avfs:/online?account%3dpk-customer%26provider%3dCyberProtect#arl:/
↪00000000-0000-0000-0000-000000000000/00000000-0000-0000-0000-000000000000/86150517-97B7-
↪C2E2-0413-F137B16AE87F/2382810E-8197-47EE-85DC-BA3EF28EB98C?archive_name%3dAADDcSu@odio_
↪mock_200_account_workmail_A5C0987D-F28C-3B2B-A8DD-93D833A3FA05_edd0ec02-67df-4017-abd8-
↪ee37611dbeb5A',
            'created_at': '2025-09-29T16:02:45Z',
            'size': 24576,
            'deduplicated_size': 0,
            'backed_up_data_size': 0,
            'original_data_size': 39405,
            'tenant_id': 'f9cf6b51-90c8-43bb-8da5-1d6018de265e',
            'resource_ids': [
                'A5C0987D-F28C-3B2B-A8DD-93D833A3FA05'
            ],
            'type': 'FULL',
            'marked_for_deletion': false,
            'last_seen_at': '2025-09-29T16:02:45Z',
            'validation_status': null,
            'created_in_network_isolation': false,
            'deleted': false
        },
        {
            'backup_id': 'd9a2f2ba-3f36-4be1-92a4-989479bccedc',
            'vault_id': '21882913-cc56-646e-1036-9eac88c3977a',
            'archive_id': 'afe414ad-34c9-1dd0-6955-27619a7ce7a3',
            'legacy_id': 'avfs:/online?account%3dpk-customer%26provider%3dCyberProtect#arl:/
↪00000000-0000-0000-0000-000000000000/00000000-0000-0000-0000-000000000000/AFE414AD-34C9-
↪1DD0-6955-27619A7CE7A3/D9A2F2BA-3F36-4BE1-92A4-989479BCCEDC?archive_name%3daajsSxM@odio_
↪mock_200_account_workmail_1316D72D-9923-3061-B4DA-26A46398A795_edd0ec02-67df-4017-abd8-
↪ee37611dbeb5A',
            'created_at': '2025-09-29T16:02:45Z',
            'size': 24576,
            'deduplicated_size': 0,
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
            'backed_up_data_size': 0,
            'original_data_size': 38920,
            'tenant_id': 'f9cf6b51-90c8-43bb-8da5-1d6018de265e',
            'resource_ids': [
                '1316D72D-9923-3061-B4DA-26A46398A795'
            ],
            'type': 'FULL',
            'marked_for_deletion': false,
            'last_seen_at': '2025-09-29T16:02:45Z',
            'validation_status': null,
            'created_in_network_isolation': false,
            'deleted': false
        }
        ...
        ],
    ”paging”: {
        ”total_count”: 51,
        ”cursors”: {}
    },
}
```

5. Преобразуйте текст в формате JSON, содержащийся в теле ответа, в объект, а затем извлеките данные о резервных копиях из ответа:

```
>>> backups = response.json()['items']
```

Удаление резервных копий

1. Выполните аутентификацию на платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Запросите резервные копии, как описано в разделе *Просмотр резервных копий*. Задайте переменную `backup_id` и присвойте ей в качестве значения идентификатор резервной копии. В данном примере использован идентификатор первой резервной копии:

```
>>> backup_id = backups[0]['backup_id']
>>> backup_id
b79a69d6-3baf-44d1-b367-5a9038e258f2
```

> **Примечание**
>
> Для удаления нескольких резервных копий в одном запросе укажите их `id` в таком виде: `&backup_id=<backup_id_1>&backup_id=<backup_id_2>`.

3. Задайте переменную `archive_id` и присвойте ей в качестве значения идентификатор архива выбранной резервной копии:

```
>>> archive_id = backups[0]['archive_id']
>>> archive_id
21882913-cc56-646e-1036-9eac88c3977a
```

4. Задайте переменную `vault_id` и присвойте ей в качестве значения идентификатор хранилища выбранной резервной копии:

```
>>> vault_id = backups[0]['vault_id']
>>> vault_id
21882913-cc56-646e-1036-9eac88c3977a
```

5. Задайте переменную `resource_id` и присвойте ей в качестве значения `id` ресурса, резервную копию для которого нужно удалить:

```
>>> resource_id = backups[0]['resource_ids'][0]
```

6. Запросите ресурсы, как описано в разделе *Получение списка ресурсов*. Задайте переменную `machine_id` и присвойте ей в качестве значения `agent_id` агента. В примере использован `agent_id` агента ресурса, резервную копию для которого нужно удалить:

```
>>> machine_id = next(
    (item['agent_id'] for item in data['items'] if item['id'] == 'resource_id'),
    None
)
>>> machine_id
122da62d-ff99-4fa7-b3cd-2580ee831a31
```

7. Отправьте запрос DELETE на конечную точку `/backups`. URL-адрес конечной точки должен

содержать параметры запроса `backup_id` и `vault_id`.

```
>>> params = {'backup_id': backup_id, 'archive_id': archive_id, 'vault_id': vault_id,
↪'machine_id': machine_id}
>>> response = requests.delete(f'{base_url}/backups', headers=auth, params=params)
```

8. Проверьте код состояния запроса:

```
>>> response.status_code
204
```

Код состояния HTTP 204 означает, что удаление резервных копий выполнено.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Задача

Задача — это единица выполнения, достаточно значимая (ценная) для того, чтобы конечный пользователь мог отслеживать её и управлять ею. Задача состоит из набора действий (шагов в выполнении задачи), которые выполняются в определённое время или при наступлении определённого события.

Все операции с задачами находятся в конечной точке `/tasks`.

Структура объекта JSON задачи

Описание структуры объекта JSON задачи доступно по ссылке.

Справочники

Справочники доступны по ссылке.

Действия с задачами

Операция	Используемые методы и конечные точки
Получение списка задач	`GET /tasks`
Получение информации об отдельной задаче	`GET /tasks/{tasks_id}`

Получение списка задач

1. Выполните аутентификацию на платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Отправьте запрос GET на конечную точку `/tasks`:

```
>>> response = requests.get(f'{base_url}/tasks', headers=auth)
```

> **Примечание**
>
> Список доступных параметров строки запроса приведен в справочнике API.

3. Проверьте код состояния запроса:

```
>>> response.status_code
200
```

   Код состояния HTTP 200 означает, что запрос был успешно выполнен.

   Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

   Кроме того, тело ответа содержит массив `items` в формате JSON, содержащий объекты задачи. После преобразования в объект он будет выглядеть так:

```
>>> pprint.pprint(response.json())
{
    'items': [
        {
            'cancel_requested': false,
            'cancellable': true,
            'completed_at': '2025-09-23T12:48:24.590984669Z',
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)


```
            'context': {
                'is_legacy': true,
                'is_process_root': true,
                'persistent': {
                    'id': '8b6e876b-69b3-4b2b-95a9-077722b1fe8d',
                    'name': 'main-win11',
                    'owner_id': '18'
                },
                'specific': 'Business',
                'title': 'Adding agent 'main-win11' to the management server',
                'user_name': 'main-win11\\CMS User'
            },
            'enqueued_at': '2025-09-23T12:48:24.590984669Z',
            'id': '1628761130376499200',
            'issuer': {
                'cluster_id': '',
                'id': ''
            },
            'priority': 'NORMAL',
            'queue': 'legacySync',
            'result': {
                'code': 'OK',
                'payload': 'MachineManagement::AddAgentResponse'
            },
            'started_at': '2025-09-23T12:48:24.590984669Z',
            'started_by_user': 'main-win11\\CMS User',
            'state': 'COMPLETED',
            'tenant': {
                'id': 'f9cf6b51-90c8-43bb-8da5-1d6018de265e',
                'locator': '/1/17/18/',
                'name': 'pk-customer'
            },
            'type': 'A59E8BF2-39C3-42C4-B667-CB672381A214',
            'updated_at': '2025-09-23T12:48:24.593232809Z',
            'uuid': '65d8ffd7-d953-49d9-8d7d-1fdc3714d3ae'
        },
        {
            'cancel_requested': false,
            'cancellable': true,
            'completed_at': '2025-09-23T12:48:22Z',
```


(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
            'context': {
                '_runtime': {
                    'source_stamp': 20
                },
                'command_id': 'E510638A-E988-4D90-9985-80CFAF8981FC',
                'execution_queue': 'default_queue',
                'has_children': false,
                'is_legacy': true,
                'is_process_root': true,
                'machine_name': 'main-win11',
                'message_id': '',
                'parent_uuid': 'f14e531d-faa2-49b9-a250-a3a05a48afb7',
                'process_id': 76332,
                'specific': 'Business',
                'title': 'Upgrading Startup Recovery Manager',
                'user_name': 'WORKGROUP\\MAIN-WIN11$'
            },
            'enqueued_at': '2025-09-23T12:48:21Z',
            'executor': {
                'cluster_id': '',
                'id': '8b6e876b-69b3-4b2b-95a9-077722b1fe8d'
            },
            'id': '1628761138328391680',
            'issuer': {
                'cluster_id': '',
                'id': ''
            },
            'priority': 'NORMAL',
            'progress': {
                'current': 100,
                'total': 100
            },
            'queue': 'legacySync',
            'result': {
                'code': 'OK',
                'payload': null
            },
            'started_at': '2025-09-23T12:48:21Z',
            'state': 'COMPLETED',
            'tenant': {
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
            'id': 'f9cf6b51-90c8-43bb-8da5-1d6018de265e',
            'locator': '/1/17/18/',
            'name': 'pk-customer'
        },
        'type': 'E510638A-E988-4D90-9985-80CFAF8981FC',
        'updated_at': '2025-09-23T12:48:26.58076431Z',
        'uuid': 'f14e531d-faa2-49b9-a250-a3a05a48afb7'
    },
    {
        'cancel_requested': false,
        'cancellable': true,
        'completed_at': '2025-09-23T12:52:37.4201818Z',
        'context': {
            'account_name': 'VK WorkMail'
        },
        'enqueued_at': '2025-09-23T12:52:37.1783328Z',
        'executor': {
            'cluster_id': '',
            'id': '8b6e876b-69b3-4b2b-95a9-077722b1fe8d'
        },
        'id': '1628762140825767936',
        'issuer': {
            'cluster_id': '',
            'id': ''
        },
        'priority': 'NORMAL',
        'progress': {
            'current': 100,
            'total': 100
        },
        'queue': 'queue_gba_discovery',
        'result': {
            'code': 'OK',
            'payload': {
                'accounts': 0,
                'domains': 0,
                'start_time': '2025-09-23T15:52:37.2245779+03:00'
            }
        },
        'started_at': '2025-09-23T12:52:37.1896586Z',
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
            'state': 'COMPLETED',
            'tenant': {
                'id': 'f9cf6b51-90c8-43bb-8da5-1d6018de265e',
                'locator': '/1/17/18/',
                'name': 'pk-customer'
            },
            'type': 'GenericDiscovery',
            'updated_at': '2025-09-23T12:52:47.229676264Z',
            'uuid': '04424590-3e69-4f47-a757-8a1fe097078b'
        }
    ],
    'paging': {
        'cursors': {
            'after': 'eyJMb2QiOiJza...'
        }
    },
    'size': 3,
    'time_stamp': '1763130428692733875ns'
}
```

4. Преобразуйте текст в формате JSON, содержащийся в теле ответа, в объект, а затем извлеките задачи из ответа:

```
>>> tasks = response.json()['items']
```

Получение информации об отдельной задаче

1. Выполните аутентификацию на облачной платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Присвойте переменной `task_id` UUID задачи, найденной с помощью поиска (см. *Получение списка задач*). В данном примере использован идентификатор первой задачи:

```
>>> task_id = tasks[0]['id']
>>> task_id
'1628761130376499200'
```

3. Отправьте запрос GET на конечную точку `/tasks/<task_id>`:

```
>>> response = requests.get(f'{base_url}/tasks/{task_id}', headers=auth)
```

> **Примечание**
>
> Список доступных параметров строки запроса приведен в справочнике API.

4. Проверьте код состояния запроса:

```
>>> response.status_code
200
```

Код состояния HTTP 200 означает, что запрос был успешно выполнен.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Кроме того, тело ответа в формате JSON содержит информацию о задаче. После преобразования в объект он будет выглядеть так:

```
>>> pprint.pprint(response.json())
{
    'cancel_requested': false,
    'cancellable': true,
    'completed_at': '2025-09-23T12:48:24.590984669Z',
    'context': {
        'is_legacy': true,
        'is_process_root': true,
        'persistent': {
            'id': '8b6e876b-69b3-4b2b-95a9-077722b1fe8d',
            'name': 'main-win11',
            'owner_id': '18'
        },
        'specific': 'Business',
        'title': 'Adding agent 'main-win11' to the management server',
        'user_name': 'main-win11\\CMS User'
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)


```
    },
    'enqueued_at': '2025-09-23T12:48:24.590984669Z',
    'id': '1628761130376499200',
    'issuer': {
        'cluster_id': '',
        'id': ''
    },
    'priority': 'NORMAL',
    'queue': 'legacySync',
    'result': {
        'code': 'OK',
        'payload': 'MachineManagement::AddAgentResponse'
    },
    'started_at': '2025-09-23T12:48:24.590984669Z',
    'started_by_user': 'main-win11\\CMS User',
    'state': 'COMPLETED',
    'tenant': {
        'id': 'f9cf6b51-90c8-43bb-8da5-1d6018de265e',
        'name': 'pk-customer'
    },
    'type': 'A59E8BF2-39C3-42C4-B667-CB672381A214',
    'updated_at': '2025-09-23T12:48:24.593232809Z',
    'uuid': '65d8ffd7-d953-49d9-8d7d-1fdc3714d3ae'
}
```


Действие

Действие — это последовательный набор операций, направленных на достижение некоторой конечной и четко определенной цели пользователя. Обычно является шагом выполнения задачи. Действие выполняется для ресурса, его параметры определены используемыми политиками. Действие может быть инициировано пользователем или самой платформой, используется для подробного представления задачи конечному пользователю.

Все операции с действиями находятся в конечной точке `/activities`.

Структура объекта JSON действия

Описание структуры объекта JSON действия доступно по ссылке.

Справочники

Справочники доступны по ссылке.

Операции с действиями

Операция	Используемые методы и конечные точки
Получение списка действий	`GET /activities`
Получение информации о действии	`GET /activities/{activity_id}`

Получение списка действий

1. Выполните аутентификацию на платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

   ```
   >>> base_url  # the base URL of the API
   'https://installaddress.ru/api/v1'
   >>> auth  # the 'Authorization' header value with the access token
   {'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
   ```

2. Отправьте запрос GET на конечную точку `/activities`:

   ```
   >>> response = requests.get(f'{base_url}/activities', headers=auth)
   ```

   Примечание

   Список доступных параметров строки запроса приведен в справочнике API.

3. Проверьте код состояния запроса:

   ```
   >>> response.status_code
   200
   ```

   Код состояния HTTP 200 означает, что запрос был успешно выполнен.

   Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Кроме того, тело ответа содержит массив `items` в формате JSON, содержащий объекты действий. После преобразования в объект он будет выглядеть так:

```
>>> pprint.pprint(response.json())
{
    'items': [
        {
            'completed_at': '2025-11-17T07:38:59.4628885Z',
            'context': {
                'is_legacy': true,
                'is_process_root': true,
                'persistent': {
                    'id': 'd45dc397-541a-49fa-a0b1-fb20b93d09d9',
                    'name': 'Template-WindowsServer-2022',
                    'owner_id': '00000000-0000-0000-0000-000000000000'
                },
                'specific': 'Business',
                'title': 'Adding agent 'Template-WindowsServer-2022' to the management server
↪',
                'user_name': 'Template-WindowsServer-2022\\CMS User'
            },
            'created_at': '2025-11-17T07:38:59.4628885Z',
            'executor': {
                'cluster_id': '',
                'id': ''
            },
            'id': '1647694869855993856',
            'result': {
                'code': 'OK',
                'payload': 'MachineManagement::AddAgentResponse'
            },
            'started_at': '2025-11-17T07:38:59.4628885Z',
            'started_by_user': 'Template-WindowsServer-2022\\CMS User',
            'state': 'COMPLETED',
            'task_id': '1647694869855993856',
            'tenant': {
                'id': '00000000-0000-0000-0000-000000000000',
                'locator': '/00000000-0000-0000-0000-000000000000/',
                'name': 'Organization'
            },
            'type': 'A59E8BF2-39C3-42C4-B667-CB672381A214',
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
            'updated_at': '2025-11-17T07:38:59.4634214Z',
            'uuid': '7b6f1aba-1d6c-4d4e-9d5c-fe97316a641a'
        }
    ],
    'size': 1,
    'time_stamp': '1763367418236566900ns'
}
```

4. Преобразуйте текст в формате JSON, содержащийся в теле ответа, в объект, а затем извлеките действия из ответа:

```
>>> activities = response.json()['items']
```

Получение информации о действии

1. Выполните аутентификацию на облачной платформе с помощью оболочки Python (см. *Аутентификация на платформе Кибер Бэкап*).

   Должны стать доступны следующие переменные:

```
>>> base_url  # the base URL of the API
'https://installaddress.ru/api/v1'
>>> auth  # the 'Authorization' header value with the access token
{'Authorization': 'Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImMwMD...'}
```

2. Присвойте переменной `activity_id` UUID задачи, найденной с помощью поиска (см. *Получение списка действий*). В данном примере использован идентификатор первого действия:

```
>>> activity_id = activities[0]['id']
>>> activity_id
'1647694869855993856'
```

3. Отправьте запрос GET на конечную точку `/activities/<activity_id>`:

```
>>> response = requests.get(f'{base_url}/activities/{activity_id}', headers=auth)
```

> **Примечание**
>
> Список доступных параметров строки запроса приведен в справочнике API.

4. Проверьте код состояния запроса:

```
>>> response.status_code
200
```

Код состояния HTTP 200 означает, что запрос был успешно выполнен.

Другой код состояния означает, что произошла ошибка. Подробные сведения об ошибке см. в разделе *Коды состояния HTTP*.

Кроме того, тело ответа в формате JSON содержит информацию о действии. После преобразования в объект она будет выглядеть так:

```
>>> pprint.pprint(response.json())
{
    'completed_at': '2025-11-17T07:38:59.4628885Z',
    'context': {
        'is_legacy': true,
        'is_process_root': true,
        'persistent': {
            'id': 'd45dc397-541a-49fa-a0b1-fb20b93d09d9',
            'name': 'Template-WindowsServer-2022',
            'owner_id': '00000000-0000-0000-0000-000000000000'
        },
        'specific': 'Business',
        'title': 'Adding agent 'Template-WindowsServer-2022' to the management server',
        'user_name': 'Template-WindowsServer-2022\\CMS User'
    },
    'created_at': '2025-11-17T07:38:59.4628885Z',
    'executor': {
        'cluster_id': '',
        'id': ''
    },
    'id': '1647694869855993856',
    'result': {
        'code': 'OK',
        'payload': 'MachineManagement::AddAgentResponse'
    },
    'started_at': '2025-11-17T07:38:59.4628885Z',
    'started_by_user': 'Template-WindowsServer-2022\\CMS User',
    'state': 'COMPLETED',
    'task_id': '1647694869855993856',
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
    'tenant': {
        'id': '00000000-0000-0000-0000-000000000000',
        'locator': '/00000000-0000-0000-0000-000000000000/',
        'name': 'Organization'
    },
    'type': 'A59E8BF2-39C3-42C4-B667-CB672381A214',
    'updated_at': '2025-11-17T07:38:59.4634214Z',
    'uuid': '7b6f1aba-1d6c-4d4e-9d5c-fe97316a641a'
}
```

Справочники

Справочник API

Справочник находится по ссылке.

Справочник по атрибутам ресурсов

Справочник находится по ссылке.

Тип ресурса	Ссылка на описание параметров
`Физическая машина, виртуальная машина, группы`	Параметры описаны в схеме Resource
`Ресурсы Microsoft SQL Server`	Параметры описаны в схеме MssqlResource
`Ресурсы Microsoft Exchange Server`	Параметры описаны в схеме MsexchangeResource

Справочник по настройкам политик

Наименование политики	Ссылка на описание параметров
`policy.backup.machine`	Резервное копирование на уровне диска (вся машина) данных физических и виртуальных машин (в том числе VMWare ESXi, Hyper-V, OpenStack, KVM, Citrix, Nutanix). Параметры описаны в схеме MachineBackupSettings
`policy.backup.disks`	Резервное копирование на уровне диска (отдельные диски и тома) данных физических и виртуальных машин. Параметры описаны в схеме DiskBackupSettings
`policy.backup.mssql`	Резервное копирование баз данных Microsoft SQL Server. Параметры описаны в схеме MssqlBackupSettings

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

Наименование политики	Ссылка на описание параметров
`policy.backup.msexchange.db`	Резервное копирование баз данных Microsoft Exchange Server. Параметры описаны в схеме MsexchangeDbBackupSettings
`policy.backup.msexchange.doc`	Резервное копирование почтовых ящиков Microsoft Exchange Server. Параметры описаны в схеме MsexchangeDocBackupSettings

Список изменений

Список изменений находится по ссылке.

Интеграция с агентами

Параметры командной строки для установщика агентов

Параметры программы установки для агентов Windows

```
--help
```

Показать страницу со справочной информацией.

```
--quiet
```

Запускает программу установки без отображения интерфейса пользователя (автоматическая установка).

```
--log-dir=<путь>
```

Путь к папке для сохранения журналов установки.

```
--language=<code>
```

Язык, который будет использовать в продукте и его компонентах. Поддерживаемые значения:

- en — English
- ru — Русский

```
--register-only
```

Применимо только к продукту Кибер Бэкап Облачный. Пропускает установку и показывает только страницу регистрации (регистрация OAuth 2.0). Не использовать с параметром `--quiet`.

```
--anti-tamper-password=<пароль>
```

Пароль с защитой от несанкционированного изменения. Может потребоваться изменить некоторые установленные компоненты.

Файлы установки

```
--installation-files=<путь/url-адрес>
```

URL-адрес или путь файловой системы к файлам установки.

```
--skip-https-cert-verification
```

Пропустить проверку сертификата сервера при скачивании установочных файлов.

```
--skip-file-trust-verification
```

Пропустить проверку доверия скачанных файлов установки.

```
--attempts-number=<число>
```

Количество попыток скачать файл установки. По умолчанию установлено значение «1».

```
--cache-directory=<tmp>
```

Путь к локальной папке, в которой перед установкой будет создан кэш файлов установки.

Компоненты

```
--add-components=<компонент1,компонент2,…,компонентN>
```

Компоненты и группы компонентов для добавления в устанавливаемый продукт.

Если этот параметр не указан, продукт устанавливается с набором компонентов по умолчанию в зависимости от настройки программы установки. Если продукт уже установлен, он будет восстановлен или обновлен в соответствии с версией программы установки. Если этот параметр указан, продукт будет установлен с компонентами и группами компонентов, указанными в значении (можно объединить компоненты и группы в значение). Указанные компоненты, которые уже установлены, будут восстановлены или обновлены в соответствии с версией программы установки.

Поддерживаемые значения компонента, общие для Кибер Бэкап и Кибер Бэкап Облачный:

- `agentForAd` — Агент для Active Directory;
- `agentForExchange` — Агент для Exchange;
- `agentForHyperV` — Агент для Hyper-V;
- `agentForEsx` — Агент для VMware (Windows);

- `agentForOracle` — Агент для Oracle;
- `agentForSql` — Агент для SQL;
- `agentForPostgreSql` — Агент для PostgreSQL;
- `agentForWindows` — Агент для Windows;
- `commandLine` — Программа командной строки;
- `mediaBuilder` — Мастер создания загрузочных носителей;
- `trayMonitor` — Мониторинг Защиты Данных.

Поддерживаемые значения компонента только для Кибер Бэкап:

- `managementServer` — Сервер управления;
- `remotelnstallation` — Компоненты для удаленной установки;
- `virtualAppliance` — Агент для VMware (виртуальное устройство);
- `agentForOffice365` — Агент для Office 365;
- `pxeServer` — PXE-сервер;
- `agentForCommunigate` — Агент для CommuniGate Pro;
- `agentForWorkmail` — Агент для VK WorkMail;
- `storageNode` — Узел хранения;
- `catalogService` — Служба каталога.

Поддерживаемые значения группы:

- `all` — В этой группе объединены все компоненты;
- `allAgents` — В этой группе объединены все агенты.

```
--remove-components=<компонент 1,компонент 2,…,компонент>
```

Компоненты и группы компонентов для удаления.

Используя этот параметр, можно удалить только компоненты продукта. Чтобы полностью удалить продукт, откройте ”Панель управления” и выберите ”Программы и компоненты” > ”Cyber Backup” > ”Удалить”. Поддерживаемые значения см. в описании параметра `-add`.

Путь установки

```
--installdir=<путь>
```

Путь к папке для установки продукта и его компонентов. Если папка не существует, она будет создана.

Регистрация

```
--reg-address=<url>
```

URL-адрес сервера управления.

Для Кибер Бэкап возможно также использование `<имя/ip-адрес>`.

```
--reg-login=<имя входа>
```

Имя входа для сервера управления. Не используйте с `--reg-token`. Если этот параметр указан, параметр `--registration` необязателен.

```
--reg-password=<пароль>
```

Пароль для сервера управления. Не используйте с `--reg-token`. Если этот параметр указан, параметр `--registration` необязателен.

```
--reg-token=<маркер>
```

Маркер регистрации. Не используйте с параметрами `--reg-login` или `--reg-password`. Если этот параметр указан, параметр–`registration` необязателен.

```
--registration=[skip | by-credentials | by-token | device-flow | current-user]
```

Тип регистрации агента после установки. Чтобы пропустить регистрацию, укажите параметр `skip`. Чтобы зарегистрировать агенты по учетным данным или маркеру, укажите `by-credentials` или `by-token` и введите соответствующие параметры.

Применимо только к Кибер Бэкап Облачный. Чтобы зарегистрировать агентов, используя протокол OAuth 2.0, укажите параметр `device-flow`. В этом случае программа установки покажет страницу регистрации по окончании установки. Не используйте `--registration=device-flow` с параметром `--quiet`.

Применимо только к Кибер Бэкап. Чтобы зарегистрировать агентов с использованием учетной записи текущего пользователя Windows, укажите `current-user`.

```
--reg-tenant=<клиент>
```

Применимо только к Кибер Бэкап. Отдел для регистрации агента.

```
--reg-transport=[https | https-ca-system | https-ca-bundle | https-pinned-public-key]
```

Применим только для Кибер Бэкап. Тип передачи данных, который будет использоваться для регистрации агента. Чтобы выполнить регистрацию через HTTPS без проверки сертификата, укажите `https`. Чтобы выполнить регистрацию через HTTPS с проверкой сертификата в системном центре сертификации, укажите `https-ca-system`. Чтобы выполнить регистрацию через HTTPS с проверкой сертификата с использованием пакета центра сертификации, который прилагается к продукту, укажите `https-ca-bundle`. Чтобы выполнить регистрацию по HTTPS с проверкой сертификата с использованием закрепленного открытого ключа, укажите `https-pinned-public-key`. В этом случае необходимо указать параметр `reg-transport-pinned-public-key`.

```
--reg-transport-pinned-public-key=<закрепленный открытый ключ>
```

Применим только для Кибер Бэкап. Закрепленный открытый ключ. Если этот параметр определен, параметр `reg-transport` должен иметь значение `https-pinned-public-key` или быть опущен.

```
--reg-script=<путь к файлу скрипта регистрации>
--reg-script-params="-c <путь к файлу registration.json> -u <имя пользователя администратора
↪Active Directory> -p <пароль администратора Active Directory> -l <путь к log файлу>"
```

Учетная запись для входа службы агента

```
--agent-account-login=<имя входа>
```

Имя входа для учетной записи службы.

```
--agent-account-password=<пароль>
```

Пароль для учетной записи службы.

```
--agent-account=[system | new | custom]
```

Учетная запись, с которой будет запускаться служба агента Кибер Бэкап Облачный. Чтобы использовать учетные записи пользователя услуги, укажите `system`. Чтобы создать новую учетную запись, укажите `new`. Чтобы использовать существующую учетную запись, укажите `custom` или оставьте это поле пустым.

Учетная запись входа для службы сервера управления

Параметры раздела применимы только для Кибер Бэкап.

```
--management-server-account-login=<имя входа>
```

Имя входа для учетной записи службы.

```
--management-server-account-password=<пароль>
```

Пароль для учетной записи службы.

```
--management-server-account=[system | new | custom]
```

Учетная запись, с которой будет запускаться служба сервера управления. Чтобы использовать учетные записи пользователя услуги, укажите `system`. Чтобы создать новую учетную запись, укажите `new`. Чтобы использовать существующую учетную запись, укажите `custom` или оставьте это поле пустым.

Учетная запись входа для службы узла хранения

Параметры раздела применимы только для Кибер Бэкап.

```
--storage-mode-account-login=<имя входа>
```

Имя входа для учетной записи службы.

```
--storage-mode-account-password=<пароль>
```

Пароль для учетной записи службы.

```
--storage-node-account=[system | new | custom]
```

Учетная запись, с которой будет запускаться служба узла хранения. Чтобы использовать учетные записи пользователя услуги, укажите `system`. Чтобы создать новую учетную запись, укажите `new`. Чтобы использовать существующую учетную запись, укажите `custom` или оставьте это поле пустым.

Порт HTTP

Параметры раздела применимы только для Кибер Бэкап.

```
--web-server-port=<порт>
```

Этот порт используется для безопасного обмена данными с веб-интерфейсом.

TCP-порт для компонентов

Параметры раздела применимы только для Кибер Бэкап.

```
--zmq-port=<порт>
```

Этот порт используется для обмена данными между компонентами продукта.

Прокси-сервер HTTP

```
--http-proxy-address=<хост>:<порт>
```

Имя/IP-адрес и порт настраиваемого прокси-сервера HTTP.

```
--http-proxy-login=<имя входа>
```

Имя входя для настраиваемого прокси-сервера HTTP.

```
--http-proxy-password=<пароль>
```

Пароль для настраиваемого прокси-сервера HTTP.

```
--http-proxy=[none | system | custom]
```

Дает инструкцию, использовать ли прокси-сервер HTTP для резервного копирования данных в облачное хранилище данных и восстановление данных из облачного хранилища по сети Интернет. Если не нужно использовать прокси-сервер, укажите `none`. Чтобы использовать прокси-сервер в масштабе системы, укажите `system`. Чтобы использовать какой-либо другой прокси-сервер, укажите `custom`, а затем укажите его адрес и учетные данные в соответствующих ключах.

vCenter/Esxi

```
--esxi-address=<хост>
```

Имя хоста или IP-адрес vCenter Server или ESXi.

```
--esxi-login=<имя входа>
```

Имя входа для vCenter Server или ESXi.

```
--esxi-password=<пароль>
```

Пароль для vCenter Server или ESXi.

База данных для сервера управления

Параметры раздела применимы только для Кибер Бэкап.

```
--management-server-db-auth=[service-account | <user>:<password>]
```

Укажите способ проверки подлинности, который будет использоваться для данного экземпляра.

```
--management-server-db-driver=<драйвер>
```

Драйвер для Microsoft SQL Server.

```
--management-server-db-instance=<экземпляр>
```

Экземпляр Microsoft SQL Server.

```
--management-server-db=[built-in | mssql]
```

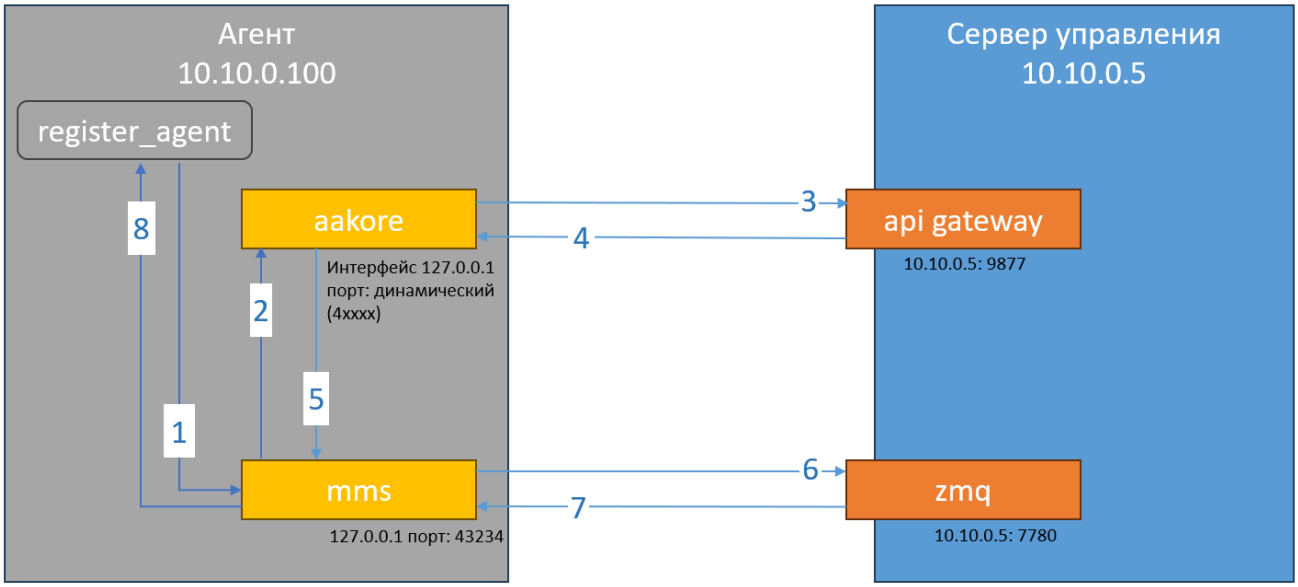
База данных, которая будет использоваться сервером управления.

Параметры установки для агентов Linux

Для Кибер Бэкап Облачный см. описание в разделе ”Параметры автоматической установки или автоматического удаления” инструкции пользователя.

Параметры командной строки для регистрационной утилиты

Регистрация агента — это процедура подключения агента к серверу управления, который будет монопольно управлять данным агентом.



Регистрация состоит из нескольких этапов:

1. Утилита **register-agent** обращается к службе агента **mms** (127.0.0.1:43234) с просьбой зарегистрировать агента на сервере управления 10.10.0.5 (пример).
2. Служба **mms** передает запрос на службу **aakore** (127.0.0.1:<порт динамический в диапазоне 40000-50000, меняется при перезапуске>).
3. Служба **aakore** через внешний интерфейс обращается к службе **api-gateway** (10.10.0.5:9877) на сервере управления для обмена ключами и первичной регистрации агента.
4. Служба **api-gateway** регистрирует агента и выдает ключи службе **aakore** для дальнейшей работы по установленному ранее соединению.
5. Служба **aakore** передает полученные ключи службе **mms** (127.0.0.1:43234).
6. Служба **mms** устанавливает соединение со службой **zmq** (10.10.0.5:7780) на сервере управления и завершает регистрацию. На этом этапе в разделе ”Настройки - агенты” заполняется информация об агенте (ОС, IP-адреса, версия агента и т. д.), и он становится активен.
7. Начинается постоянная синхронизация агента и сервера управления.
8. Служба **mms** возвращает результат регистрации утилите **register-agent**.

Ручная регистрация

В случае если Вам необходимо зарегистрировать уже установленного агента на новом сервере управления или обновить его регистрацию, то можно воспользоваться одним из следующих способов.

Регистрация агента для Windows

Обычная регистрация агента, если разрешена анонимная регистрация

На компьютере с агентом откройте командную строку: **Пуск** → **Поиск** (или **Выполнить**), введите `cmd.exe` и нажмите `Enter`.

Перейдите в папку `C:\\Program Files\\Acronis\\RegisterAgentTool`:

```
cd ”C:\Program Files\Acronis\RegisterAgentTool”
```

Выполните команду:

```
register_agent.exe -o register -a <адрес сервера управления>
```

Для регистрации Узла Хранения на сервере управления необходимо на машине с Узлом Хранения выполнить следующую команду.

Для Windows:

```
cd ”C:\Program Files\Acronis\RegisterAgentTool\register_agent.exe” -o register -s asn -a
↪<management server address>
```

Для Linux:

```
sudo
”/usr/lib/Acronis/RegisterAgentTool/RegisterAgent” -o register -s asn -a <management server¬
↪address>
```

Регистрация с указанием пользователя

```
register_agent.exe -o register -a <адрес сервера управления> -u <имя пользователя> -p <пароль>
```

В приведенной выше команде:

`<адрес сервера управления>` — это IP-адрес или имя хоста сервера управления, на котором должен быть зарегистрирован агент.

`<username>` — это имя администратора для сервера управления.

<пароль> — пароль администратора для сервера управления.

Пример:

```
register_agent.exe -o register -a 192.168.1.0 -u admin -p admin1234
```

Регистрация агента в определенном отделе

```
register_agent.exe -o register -a <адрес сервера управления> --tenant <id_отдела> -u <имя¬
↪пользователя> -p <пароль>
```

Чтобы найти идентификатор отдела, в консоли управления сервера управления перейдите в **Настройки** → **Учетные записи** и далее в нужный Отдел. После этого нажмите кнопку **Сведения** и скопируйте значение свойства ”id” (значение вида ”556c6428-34ef-11ed-83ee-00155d8b5d06”).

Регистрация агента с помощью токена

```
register_agent.exe -o register -a <адрес сервера управления:порт> --token <токен>
```

<токен> — маркер регистрации который можно создать в разделе **Устройства** → **Добавить** → **Маркер регистрации** → **создать**.

Удаление регистрации агента

Для удаления регистрации агента используйте команду регистрации с параметром `-o unregister`:

```
register_agent.exe -o unregister
```

Регистрация агента для Linux

Обычная регистрация

На машине агента откройте **Терминал**.

Выполните команду:

```
/usr/lib/Acronis/RegisterAgentTool/RegisterAgent -o register -a <адрес сервера управления>
```

Регистрация с указанием пользователя

```
/usr/lib/Acronis/RegisterAgentTool/RegisterAgent -o register -a <адрес сервера управления> -u
↪<имя пользователя> -p <пароль>
```

В приведенной выше команде:

`<адрес сервера управления>` — это IP-адрес или имя хоста сервера управления, на котором должен быть зарегистрирован агент.

`<username>` — это имя пользователя администратора для сервера управления.

`<пароль>` — пароль администратора для сервера управления.

Регистрация агента с помощью токена регистрации

```
/usr/lib/Acronis/RegisterAgentTool/RegisterAgent -o register -a <адрес сервера управления> --
↪token <токен>
```

В приведенной выше команде:

`<токен>` — маркер регистрации, который можно создать в разделе **Устройства** → **Добавить** → **Маркер регистрации** → **создать**.

Добавить виртуальное устройство (Virtual appliance)

Откройте консоль устройства в консоли гипервизора, затем нажмите `Ctrl+Shift+F2`, чтобы открыть терминал на устройстве.

Обычная регистрация

Выполните команду:

```
/bin/register_agent -o register -a <адрес сервера управления>
```

Регистрация с указанием пользователя

```
/bin/register_agent -o register -a <адрес сервера управления> -u <имя пользователя> -p <пароль>
```

В приведенной выше команде:

- `<адрес сервера управления>` — это IP-адрес или имя хоста сервера управления, на котором должен быть зарегистрирован агент;
- `<username>` — это имя пользователя администратора для сервера управления;

- <пароль> — пароль администратора для сервера управления.

Регистрация агента с помощью токена регистрации

```
/bin/register_agent -o register -a <адрес сервера управления:порт> --token <токен>
```

В приведенной выше команде:

<токен> — маркер регистрации, который можно создать в разделе **Устройства** → **Добавить** → **Маркер регистрации** → **создать**.