

Общество с ограниченной ответственностью «ОНЛАНТА»

**Руководство по установке
программного обеспечения
«OnCloud. КОСМОС – КОРПОРАТИВНАЯ СИСТЕМА
МУЛЬТИОБЛАЧНЫХ СЕРВИСОВ»**

Москва 2022 г.

Аннотация

Настоящий документ содержит описание порядка установки и начальной настройки программного обеспечения «OnCloud. КОСМОС – КОРПОРАТИВНАЯ СИСТЕМА МУЛЬТИОБЛАЧНЫХ СЕРВИСОВ» (далее – OnCloud.КОСМОС), разработанного ООО «ОНЛАНТА».

Содержание

1 Общие сведения	4
1.1 Наименование	4
1.2 Разработчик	4
1.3 Термины и сокращения	4
2 Краткое описание программного обеспечения	6
2.1 Назначение	6
2.2 Роли	6
2.3 Требования к рабочему месту пользователя	7
3 Порядок установки программного обеспечения	7
3.1 Развёртывание OnCloud.КОСМОС	7
3.2 Обновление	14
4 Получение технической поддержки	18

1 Общие сведения

1.1 Наименование

Полное наименование: Программное обеспечение «OnCloud.КОСМОС – КОРПОРАТИВНАЯ СИСТЕМА МУЛЬТИОБЛАЧНЫХ СЕРВИСОВ».

Условное обозначение: OnCloud.КОСМОС.

1.2 Разработчик

Общество с ограниченной ответственностью «ОНЛАНТА» (ООО «ОНЛАНТА»).

Фактический адрес: 129075, г. Москва, Мурманский проезд, д. 14, стр. 5.

sales@onlanta.ru; www.onlanta.ru

Тел./факс: +7 (495) 721-12-18

1.3 Термины и сокращения

Термин/сокращение	Определение
БД	База данных
ОС	Операционная система
ПО	Программное обеспечение
API	Совокупность инструментов и функций в виде интерфейса для создания новых приложений, благодаря которому одна программа будет взаимодействовать с другой
Гипервизор	Инсталляция системы виртуализации (vSphere, OpenStack, публичные облака VK Cloud Solutions, Яндекс и т.д.), предназначенная для управления через КОСМОС)
Дамп	Файл, включающий в себя содержимое памяти компьютера или базы данных. В нашем случае это файл с расширением .SQL. Он содержит особые данные, благодаря которым можно легко воссоздать копию БД
Дистрибутив	Файл или файлы, предназначенные для установки какой-либо программы
История изменений	Все изменения, которые происходят в системе, с её элементами, сущностями и т.д. должны логироваться. Каждая строчка лога должна содержать информацию: таймштамп, затронутый элемент, инициатор изменения, описание действия. Логируются в т.ч. авторизации
KPI услуги	Ключевой показатель эффективности услуги. Используется для оценки качества оказанной услуги, а также в формулах расчёта стоимости услуг для проектов

Термин/сокращение	Определение
	за отчётный период. Данные для KPI услуги система получает от инфраструктуры в автоматическом режиме через API и рассчитывается в соответствии с заданной логикой
Отчёты	Информация о ресурсах, сформированная в определённых шаблонах представления по заданной логике за промежутки времени. Например, отчёты по потреблению мощностей, отчёты по качеству Услуги (KPI), отчёты по финансам и т.д.
Пользователи	Сотрудники компании, заказчиков, партнёров, обладающие определёнными привилегиями в рамках системы. Каждому пользователю назначается определённая Роль из списка возможных
Привилегии	Полномочия на конкретные действия, либо отображения блоков информации, списка проектов и т.д. в рамках системы
Проект	Уникальный набор контрагентов, услуг, ресурсов, пользователей и других элементов, объединённых в один логический блок
Ресурс	Отдельные виды ценностей (мощностей, лицензий, объёмов и т.д.), предоставляемые в рамках услуги. В составе услуги могут присутствовать обязательные и опциональные компоненты
Роль	Набор привилегий, ограничивающих возможности управления и отображения данных в рамках интерфейсов системы
Техническая поддержка	Возможность подачи заявок на службы технической поддержки компании через интерфейс системы. Обмен данными с ITSM-системой должен быть реализован на уровне API
Тип проекта	<p>Логическое разделение проектов по принципу оказания услуг. Возможны следующие типы проектов:</p> <ul style="list-style-type: none"> – конечный заказчик – проект, основанный на предоставлении услуг на коммерческой основе для конечного клиента; – партнёрский – проект, основанный на предоставлении услуг на коммерческой основе для партнёров по модели WhiteLabel. В рамках партнёрского проекта авторизованное лицо Партнёра может создавать свой набор проектов;

Термин/сокращение	Определение
	– инфраструктурный (внутренний) – проект, нацеленный на организацию и предоставление продуктов и решений компанией. Служат для учёта внутренних расходов компании на инфраструктуру. По функционалу не отличается от конечного заказчика
Услуга	Ценности, предоставляемые и потребляемые пользователями в рамках Проекта. Каждая Услуга состоит из набора Ресурсов

2 Краткое описание программного обеспечения

2.1 Назначение

OnCloud.KOCMOC представляет собой готовое решение для внутреннего учёта потребления ресурсов в разрезе Проектов для корректного управленческого учёта внутри компании.

OnCloud.KOCMOC предназначен для:

- получения информации об объёме использованных ресурсов виртуализации, сетевых ресурсов;
- учёта ресурсов за фактическое потребление;
- предоставления отчётности по использованным ресурсам.

2.2 Роли

В OnCloud.KOCMOC используется пятнадцать ролей пользователей, основные: клиент, менеджер проекта, сейл, пресейл.

Привилегии ролей пользователей перечислены в таблице 1.

Таблица 1 – Привилегии ролей пользователей

Роль пользователя	Привилегии роли пользователя
Клиент	<ul style="list-style-type: none"> – управление Ресурсами в Проекте – управление Услугами в Проекте – управление Отчетами в Проекте – управление настройками в Проекте – просмотр истории изменений в Проекте
Менеджер проекта	<ul style="list-style-type: none"> – просмотр доступных Проектов – управление Ресурсами в Проекте – управление Услугами в Проекте – просмотр истории изменений в Проекте – управление Отчетами в Проекте – управление доступами в Проекте – управление настройками в Проекте – доступ к Проектам

	<ul style="list-style-type: none"> – просмотр типов Ресурсов – просмотр себестоимостей – просмотр учетных записей пользователей
Сейл	<ul style="list-style-type: none"> – просмотр доступных Проектов – управление Ресурсами в Проекте – управление Услугами в Проекте – просмотр истории изменений в Проекте – управление Отчетами в Проекте – доступ к Проектам – просмотр типов Ресурсов – просмотр себестоимостей
Пресейл	<ul style="list-style-type: none"> – просмотр доступных Проектов – управление Ресурсами в Проекте – управление Услугами в Проекте – просмотр истории изменений в Проекте – управление Отчетами в Проекте – доступ к Проектам – просмотр типов Ресурсов

2.3 Требования к рабочему месту пользователя

Для полноценной работы с OnCloud.КОСМОС рабочее место пользователя должно удовлетворять следующим параметрам:

- 1) Аппаратные требования:
 - ПК на базе процессора Intel Core i3 (или более производительный);
 - оперативная память: не менее 4 GB (рекомендуется 8 GB и более);
 - HDD 320 GB (или более);
 - подключение к сети через сетевой интерфейс со скоростью не менее 100 Мбит/с;
 - память видеоадаптера: не менее 256 MB (рекомендуется 512 MB и более);
 - монитор с поддержкой разрешения не менее 1280x1024 (рекомендуется разрешение 1920x1080 и выше).
- 2) Программные требования:
 - ОС Windows 7/8/10;
 - браузер Google Chrome версии не ниже 80 или (или другой с аналогичными характеристиками).

3 Порядок установки программного обеспечения

3.1 Развёртывание OnCloud.КОСМОС

Для установки OnCloud.КОСМОС должен использоваться сервер с предварительно установленным ПО node.js v12, memcached, pm2, nginx, PostgreSQL 10.12.

Установка OnCloud.КОСМОС выполняется в следующем порядке:

- 1) Настройка Nginx

Для настройки Nginx нужно установить конфигурацию, по адресу `/etc/nginx/sites-enabled`, предварительно выпустив самоподписанный сертификат и перезагрузить Nginx:

```
server {
    listen 443 ssl;

    server_name bizeapub.onlanta.ru;

    include snippets/self-signed.conf;

    client_max_body_size 10M;

    location / {
        proxy_pass http://localhost:1234;
        proxy_connect_timeout    300s;
        proxy_send_timeout        300s;
        proxy_read_timeout        300s;
        send_timeout               300s;
    }
}
```

2) Настройка PostgreSQL 10.12

Для настройки нужно выполнить следующие действия:

- создать учетную запись пользователя;
- назначить права, созданной учетной записи;
- создать БД OnCloud.КОСМОС от имени этого пользователя;
- взять подготовленный пустой дамп OnCloud.КОСМОС или текущий и загрузить в БД.

3) Разворачивание restnode

Перед установкой restnode необходимо, установить Oracle Basic Client for Linux.

Для этого требуется скачать RPM-пакет, для соответствующей архитектуры:
<https://www.oracle.com/database/technologies/instant-client/downloads.html>

Например, https://download.oracle.com/otn_software/linux/instantclient/193000/oracleinstantclient19.3-basic-19.3.0.0.0-1.x86_64.rpm

При использовании дистрибутива `debian-based`, необходимо преобразовать пакет в `.deb` с помощью команды `alien`:

```
sudo apt install alien
```

```
alien oracle-instantclient19.3-basic-19.3.0.0.0-1.x86_64.rpm
```

Необходимо создать новую учетную запись пользователя с именем restnode (sudo adduser restnode, пароль указать любой), далее необходимо авторизоваться в пользователя (sudo su restnode), в папке пользователя создать папку app и открыть её.

Для gitlab понадобится сгенерировать ключи <https://gitlab.onlanta.ru/help/ssh/README#locating-an-existing-ssh-key-pair> (пароль при создании ключа не указывать!)

Записать:

```
git clone git@gitlab.onlanta.ru:earub_ext/restnode.git. (точка в конце не создает папку,
клонировать в текущую)
```

```
git checkout kosmos-prod
```

```
cp config.js.dist cp config.js
```

Записать в конфигурацию данные БД \ почты, поменять поля secret и groupSecret, последний запомнить, он пригодится в дальнейшем.

Записать:

```
// скачать зависимости, скомпилировать код
```

```
npm set registry https://npm.onlanta.ru/
```

```
npm ci
```

```
npm run build
```

```
// настроить перезагрузку процесса при перезагрузке сервера
```

```
pm2 start main.js
```

```
pm2 save
```

pm2 startup – выдает команду которую надо будет выполнить от имени администратора, имеющего права root

4) Настройка crontab

Записать crontab -е, выбрать редактор (нано норм) и вставить:

```
15 4 1 * * cd /home/restnode/app; ./main.js reports:generate
```

```
20 4 * * * cd /home/restnode/app; ./main.js projects:expire
```

```
30 */2 * * * cd /home/restnode/app; ./main.js backup-information:update
```

```
58 * * * * cd /home/restnode/app; ./main.js datastores:update
```

```
*/5 * * * * cd /home/restnode/app; ./main.js hypervisors:check
```

```
40 * * * * cd /home/restnode/app; ./main.js availability-information:update
```

```
41 * * * * cd /home/restnode/app; ./main.js networks:update
27 5 * * * * cd /home/restnode/app; ./main.js reports:send
30 23 * * * * cd /home/restnode/app; ./main.js bandwidths:update
```

5) Разворачивание masternode

Создать новую учетную запись пользователя с именем masternode, авторизоваться используя созданную учетную запись, создать папку «app» и открыть её.

Для gitlab понадобится сгенерировать ключи

Записать:

```
git clone git@gitlab.onlanta.ru:earub_ext/masternode.git .
cp config.js.dist cp config.js
```

Готовая конфигурация выглядит так:

```
module.exports = {
  listen: 39247,
  logdir: 'logs',
  microservices: {
    oraclenode: {
      url: 'http://localhost:52013',
    },
    mcsnode: {
      url: 'http://localhost:52099',
    },
    restnode: {
      url: 'http://localhost:1234', // проверить чтобы порты были как в конфигурациях
      у других приложений
    },
  },
  groupSecret: "", // здесь указать groupSecret, который надо было запомнить
}
// скачать зависимости, скомпилировать код
npm set registry https://npm.onlanta.ru/
```

```
npm ci
npm run build
```

// настроить перезагрузку процесса при перезагрузке сервера

```
pm2 start main.js
```

```
pm2 save
```

pm2 startup – выдает команду которую надо будет выполнить от имени администратора, имеющего права root

6) Разворачивание oraclenode

Создать новую учетную запись пользователя с именем oraclenode, выполнить авторизацию, используя созданную учетную запись, создать папку «app» и открыть её.

Для gitlab понадобится сгенерировать ключи

Записать:

```
git clone git@gitlab.onlanta.ru:earpub_ext/oraclenode.git .
```

```
cp config.js.dist cp config.js
```

Изменить groupSecret, логины и пароли APEX:

// скачать зависимости, скомпилировать код

```
npm set registry https://npm.onlanta.ru/
```

```
npm ci
```

```
npm run build
```

// настроить перезагрузку процесса при перезагрузке сервера

```
pm2 start main.js
```

```
pm2 save
```

pm2 startup – выдает команду которую надо будет выполнить от имени администратора, имеющего права root

7) Разворачивание mcsnode

Установить sudo apt install python3-openstackclient

Проверить версию, должна быть 5.5.0 (openstack --version) на ubuntu 20.04

Создать новую учетную запись пользователя с именем cloud-mail-ru, авторизоваться с использованием созданной учетной записи, создать папку «app» и открыть её.

Для gitlab понадобится сгенерировать ключи:

Записать:

```
git clone git@gitlab.onlanta.ru:earub/cloud-mail-ru.git .
```

Настроить Postgress создав для этого БД cloud-mail-ru от имени пользователя kosmos:

```
cp config.js.dist cp config.js
```

Изменить groupSecret, secret, vmKeyPairSecret

Итоговая конфигурация выглядит так:

```
/* eslint-disable @typescript-eslint/ban-ts-ignore */
/* eslint-disable @typescript-eslint/no-var-requires */
// @ts-check
/** @type {import("./src/ConfigController").Config} */
const config = {
  systemConfig: {
    secret: '8c3b10edbc146789997642456789hjd',
    isDev: false,
    port: 52099,
    preflightCheck: {
      fatal: false,
      enabled: false,
    },
  },
  db: {
    type: 'postgres',
    host: '127.0.0.1',
    port: 5432,
    name: 'cloud-mail-ru',
    user: 'kosmos',
    password: '23423423',
  },
}
```

```

    groupSecret: 'awe2o32ef@g2e3pj',
    masterMicro: 'http://localhost:39247', // url masternode
  },
  mcsConfig: {
    writeDebug: true,
    updatesDir: './mcs_check',
    vmKeyPairDir: './vm_key_pairs',
    vmKeyPairSecret: '1wa2gW#(@!f1ojih', // длина 16 символов
    frontUrl: 'https://mcs.mail.ru/app/',
  },
  mailConfig: {
    host: 'smtp.mailtrap.io',
    port: 2525,
    secure: false,
    user: '3d5acde2c61cdd',
    password: 'bbda2cddd28',
    sender: 'eapub@onlanta.ru',
    ignoreTLS: false,
    siteurl: 'localhost:3000',
    rateLimiter: {
      callLimit: 1,
      periodInMs: 1,
    },
  },
}

module.exports = config

// скачать зависимости, скомпилировать код
npm set registry https://npm.onlanta.ru/
npm ci
npm run build
npm run migrate // перенос таблиц в БД из конфигурационного файла

```

// настроить перезагрузку процесса при перезагрузке сервера

```
pm2 start main.js
```

```
pm2 save
```

pm2 startup – выдает команду которую надо будет выполнить от имени администратора, имеющего права root

Настроить crontab:

Записать:

crontab -e, выбрать редактор (нано норм) и вставить:

```
*/10 * * * * cd /home/cloud-mail-ru/app; ./main.js mcs:check
```

```
0 */4 * * * * cd /home/cloud-mail-ru/app; ./main.js mcs:check:slow
```

```
*/3 * * * * cd /home/cloud-mail-ru/app; ./main.js mcs:check:tasks
```

3.2 Обновление

Для всех сервисов выше действительно следующее:

- 1) Для авторизации с использованием учетной записи пользователя нужно перейти в его папку, а затем перейти в папку «app»;
- 2) Набрать npm run deploy, или вручную:

```
git pull // скачивает новый код заменить
```

```
npm ci // скачивает зависимости
```

```
npm run build // компилирует код
```

```
npm run migrate // актуализирует изменения БД
```

Также иногда меняются конфигурации, и иногда требуется выполнять разные команды, которые меняют БД, где не хватает миграций.

Иногда требуется сбрасывать memcached, желательно всегда его сбрасывать (telnet localhost 11211 => flush_all => quit) (закончатся сеансы всех пользователей)

Разворачивание препрода выглядит таким же образом, как и все выше, однако нужно взять БД из продуктивной среды, сделав это с помощью самописного скрипта. После обновления БД нужно удалить гипервизор в БД номер 3 (к нему нет доступа) и добавить в БД тестовую учетную запись пользователя.

- 3) Разворачивание фронта

Необходимо зайти на сервер и установить:

node.js v12, memcached, pm2, nginx

Настроить Nginx для webnode конфигурация такая же:

```
server {  
    listen 443 ssl;  
    include snippets/oncloud-wildcard.conf;  
    client_max_body_size 10M;  
    root /home/webnode/app/vue/dist;  
    server_name kosmos.oncloud.ru;  
    server_name www.kosmos.oncloud.ru;  
    index index.html index.htm index.nginx-debian.html;  
    location / {  
        # First attempt to serve request as file, then  
        # as directory, then fall back to displaying a 404.  
        # proxy_pass http://localhost:5000;  
        # proxy_set_header Host $host;  
        try_files $uri /index.html;  
        error_page 405 =200 $uri;  
    }  
    location /api {  
        proxy_pass http://localhost:4433;  
    }  
    location /assets {  
        proxy_pass http://localhost:4433;  
    }  
    location /static {  
        proxy_pass http://localhost:4433;  
    }  
}  
  
server {  
    listen 80;
```

```

server_name kosmos.oncloud.ru;
server_name www.kosmos.oncloud.ru;
return 301 https://kosmos.oncloud.ru/;
}

```

для clientnode конфигурация такая же:

```

server {
    listen 443 ssl;
    include snippets/self-signed.conf;
    root /home/clientnode/app/vue/dist;
    server_name client.kosmos.ru;
    index index.html index.htm index.nginx-debian.html;
    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        # proxy_pass http://localhost:5000;
        # proxy_set_header Host $host;
        try_files $uri /index.html;
        error_page 405 =200 $uri;
    }
    location /api {
        proxy_pass http://localhost:2345;
    }
    location /assets {
        proxy_pass http://localhost:2345;
    }
    location /static {
        proxy_pass http://localhost:2345;
    }
}

```

Не следует забывать, что необходимо подставить необходимые сертификаты.

Дальше все также как при первоначальной установке: создать учетную запись пользователя, клонировать git, изменить конфигурации.

Конфигурация webnode и clientnode выглядит так:

```
module.exports = {
  port: 4434,
  apiServer: {
    url: 'http://localhost:1235',
    appId: 1, // этот id должен храниться в БД OnCloud.КОСМОС, таблица application,
    для webnode и clientnode свой
    secret: 'sdrghserggers', // этот секрет должен храниться в БД OnCloud.КОСМОС,
    таблица application
  },
  secret: 'sergseg', // меняем на свой секрет для кук
  env: 'production', // 'development'|'production'
}
```

Также:

```
npm set registry https://npm.onlanta.ru/
```

```
npm ci
```

```
npm run build
```

```
// настроить перезагрузку процесса при перезагрузке сервера
```

```
pm2 start main.js
```

```
pm2 save
```

```
pm2 startup — выдает команду которую надо будет выполнить от имени администратора, имеющего права root
```

В папке «vue» набрать:

```
npm ci
```

```
npm run build
```

Обновление webnode и clientnode также сводится к логину и переходу в папку и набору команды `npm run deploy`, конфигурации меняются очень редко.

4 Получение технической поддержки

Для получения консультаций по вопросам, связанным с установкой OnCloud.КОСМОС, необходимо обратиться в службу технической поддержки следующими способами:

- связаться по телефону +7 (495) 721-12-18, 8-800-200-69-09 (рабочие дни с 09:00 до 18:00 по московскому времени);
- связаться по электронной почте support@onlanta.ru.

Все поступившие в службу технической поддержки обращения пользователей регистрируется в ITSM-системе производителя OnCloud.КОСМОС.

В тексте обращения, направленного в службу технической поддержки, необходимо указать наименование пользователя OnCloud.КОСМОС, номер лицензии, контактный телефон, а также подробно описать возникшую ситуацию (в т. ч. прикрепить скриншоты экрана и т. п.).