

# B4N CONTROLLER

Версия: 2.4115.3  
Редакция: 01.11.2018

© 2014 – 2018 Программируемые сети ООО (ТМ Brain4Net), Москва, Россия.

Этот документ входит в комплект поставки программного обеспечения, и на него распространяются все условия лицензионного соглашения. Ни одна из частей этого документа не может быть воспроизведена, опубликована, сохранена в электронной базе данных или передана в любой форме или любыми средствами, такими как электронные, механические, записывающие или иначе, для любой цели без предварительного письменного разрешения компании Brain4Net. Компания оставляет за собой возможность изменять, дополнять и/или производить другие действия с данным документом без уведомления.

B4N Controller является зарегистрированной торговой маркой программного обеспечения, разрабатываемого компанией Brain4Net. Все торговые марки и названия программ являются собственностью их владельцев.

117342, г. Москва, ул. Бутлерова, д17 (БЦ NEO GEO), корпус А, офис 285

Телефон: +7.499.649.4646

E-mail: [need@brain4netru](mailto:need@brain4netru)

Web: <http://www.brain4netru>

# СОДЕРЖАНИЕ

## Часть I ВВЕДЕНИЕ

1.1. Требования к аппаратному оборудованию.....	5
1.2. Требования к программному обеспечению.....	5
1.2.1. Настройка NTP.....	6
1.2.2. Установка Hostname.....	6
1.2.3. Установка Docker под CentOS 7.....	6
1.2.4. Установка Docker под Ubuntu 16.04/14.04.....	7

## Часть II УСТАНОВКА И НАСТРОЙКА

2.1. Установка MongoDB.....	10
2.1.1. Настройка MongoDB.....	10
2.2. Установка B4N Controller.....	10

## Часть III УПРАВЛЯЮЩИЕ КОМАНДЫ



# 1. ВВЕДЕНИЕ

B4N Controller является высоко эффективным, основанным на принципах OpenFlow, SDN контроллером, который помогает решить проблему медленного внедрения SDN в сетях операторов связи и корпоративных сетях. Продукт предназначен для обеспечения комплексного решения для создания надёжной, отказоустойчивой единой сетевой структуры, состоящей из физических и виртуальных сетевых элементов для сетей Metro Ethernet и ЦОД. Продукт позволяет оператору связи обеспечить путь доставки трафика клиента от сети доступа к платформе NFV расположенной в ЦОД и тесно интегрировать сервисы сети агрегации Metro с услугами NFV или другими Cloud услугами, работающими в центре обработки данных. B4N Controller ключевая часть портфеля B4N Controller, которая обеспечивает управление топологией, полный набор Carrier Ethernet EVCs (E-Line, E-LAN, E-Tree EVCs), Traffic Engineering и QoS.

B4N Controller может выступать в качестве много доменного SDN контроллера для WAN сети и центра обработки данных. Как SDN контроллер центра обработки данных, B4N Controller, обеспечивает единое управление для физических bare-metal OF коммутаторов и виртуальных коммутаторов в гипервизоре, образуя единую сеть ЦОД без каких-либо наложенных технологий. Такой подход позволяет снизить нагрузку на сеть, улучшить управление сетью и упростить поиск неисправностей.

B4N Controller может быть развёрнут как на физическом сервере, так и виртуальной машине. Установка на физической машине предполагает наличие одного, трёх или пяти физических серверов. Независимости от того, какое решение будет задействовано для разворачивания B4N Controller, должны соблюдаться определённые условия в отношении [аппаратного](#) и [программного](#) обеспечения (см. руководство DesignGuide).

При использовании стандартной процедуры установки серверам потребуется доступ в интернет. Возможна offline установка B4N Controller с предварительной локальной загрузкой Docker-образов. Приведённые ниже инструкции описывают установку решения на серверных редакциях Ubuntu или CentOS, с минимальными требованиями без административных изменений.

## 1.1. Требования к аппаратному оборудованию

Для максимально эффективной работы программного обеспечения, аппаратное оборудование должно соответствовать перечисленным ниже требованиям:

- Ядро – 6 cores;
- RAM – 12 Гб;
- Дисковое пространство – 64 Гб;
- ОС – 64 bit

## 1.2. Требования к программному обеспечению

На серверах должны быть настроены и установлены следующие элементы:

1. Операционная система (x64):
  - Debian не ниже 8.1;
  - Ubuntu не ниже 14.04 LTS;
  - CentOS не ниже 7.1;
  - RedHat не ниже 7.1;
  - SUSE не ниже 12.4;
2. Система [Docker](#) версии 1.5;
3. [NTP](#) (Network Time Protocol – протокол сетевого времени);
4. [Hostname](#).

Docker – программное обеспечение для автоматизации развёртывания и управления приложениями в среде виртуализации на уровне операционной системы. Позволяет "упаковать" приложение со всем его окружением и зависимостями в контейнер, который может быть перенесён на любую Linux-систему. Docker предоставляет возможность управления контейнерами.

Использование Docker при установке решения B4N Controller позволяет решить две задачи:

- Удобную передачу серверного решения клиенту – не нужно тратить уйму времени на решения проблем клиента в части установки скриптов и дополнительных настроек;
- Обеспечение тиражируемости серверов – возможность установки решения B4N Controller на однотипных серверах.

Использование NTP в решении B4N Controller позволяет производить синхронизацию лог файлов, и конфигурационных файлов внутри кластера серверов.

### 1.2.1. Настройка NTP

Для корректной настройки NTP, в зависимости от установленной ОС, в первую очередь проверьте правильность настройки часового пояса, и при необходимости установите корректный при помощи следующей команды:

CENTOS 7

---

```
# date
# yum -y install tzdata
# ln -s /usr/share/zoneinfo/ZONE /etc/localtime
```

UBUNTU 16.04/14.04

---

```
timedatectl
sudo timedatectl set-timezone ZONE
```

Для активации синхронизации времени на сервере по протоколу NTP выполните следующие операции:

CENTOS 7

---

```
# yum -y install ntp
# systemctl enable ntpd
# systemctl start ntpd
```

UBUNTU 16.04/14.04

---

```
sudo apt-get install ntp
```

Для подробных инструкций по настройке NTP в дистрибутиве, пожалуйста, обратитесь к соответствующей документации.

### 1.2.2. Установка Hostname

Hostname должен быть установлен в отличный от localhost и должен разрешаться (resolve) в IP адрес узла. Правила назначения IP адресов и Hostname должны быть определены на этапе дизайна проекта.

### 1.2.3. Установка Docker под CentOS 7

Docker может быть установлен из репозитория дистрибутива, либо с помощью официального зеркала проекта, согласно инструкциям с [docs.docker.com](https://docs.docker.com). Для установки Docker вручную из официального зеркала проекта, поместите в файл /etc/yum.repos.d/docker.repo следующее содержимое:

```
[dockerrepo]
name=Docker Repository
baseurl=https://yum.dockerproject.org/repo/main/centos/$releasever/
enabled=1
gpgcheck=1
gpgkey=https://yum.dockerproject.org/gpg
```

Для установки и запуска Docker выполните следующие операции:

```
# yum makecache fast
# yum -y install docker-engine
# systemctl enable docker
# systemctl start docker
```

Для проверки корректности установки Docker выполните следующую команду:

```
$ sudo docker run hello-world
```

Официальная процедура установки Docker под CentOS 7 приведена по этой ссылке <https://docs.docker.com/engine/installation/linux/centos>.

#### 1.2.4. Установка Docker под Ubuntu 16.04/14.04

Установка Docker на Ubuntu требует предварительной установки дополнительных пакетов. Для установки пакетов выполните следующие операции:

```
$ sudo apt-get update
```

```
$ sudo apt-get install apt-transport-https ca-certificates curl software-properties-common
```

Для добавления gpg-ключа, репозитория и установки Docker выполните следующие операции:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"  
$ sudo apt-get update  
$ sudo apt-get install docker-ce
```

Для проверки корректности установки Docker выполните следующую команду:

```
$ sudo docker run hello-world
```

Официальная процедура установки Docker под Ubuntu приведена по этой ссылке <https://docs.docker.com/engine/installation/linux/ubuntu>.



---

## УСТАНОВКА И НАСТРОЙКА

## 2. УСТАНОВКА И НАСТРОЙКА

### 2.1. Установка MongoDB

Использование MongoDB является хорошим решением в случае хранения объектов со сложной структурой или не однотипных объектов в базе данных. Возможности mapReduce полезны для генерации разнообразной статистики и использовании агрегирующих функций.

Для установки MongoDB через Docker выполните следующую команду:

```
docker run -d -m 2048M --memory-swap=2048M --net=host --restart=always --log-opt max-size=100m --name=ctl-dbs --cpuset-cpus=1 --entrypoint mongod mongo:latest --port 27016 --replSet b4nrs
```

В данном случае контейнеру выделяется 2 Гб RAM, ограниченные использованием swap, с использованием сети хоста без NAT и привязкой контейнера к первому ядру ОС.

#### 2.1.1. Настройка MongoDB

MongoDB в связке с B4N Controller требует небольшой конфигурации, которая заключается в создании replica set, состоящей из одной ноды. Для создания replica set выполните следующую операцию:

```
docker exec ctl-dbs mongo --port 27016 --eval 'rs.initiate()'
```

После создания replica set, необходимо дождаться появления записей в логах. Записи можно посмотреть при помощи команды `docker logs ctl-dbs`, выполнив следующую операцию:

```
transition to primary complete; database writes are now permitted
```

Следующий шаг подразумевает замену `HOST_IP_ADDRESS` на реальный IP узла при помощи следующей команды:

```
docker exec ctl-dbs mongo --port 27016 --eval 'cfg=rs.conf();cfg.members[0].host="HOST_IP_ADDRESS:27016";rs.reconfig(cfg);'
```

Для окончательной настройки MongoDB необходимо выполнить все вышеприведённые операции на всех нодах контроллера.

### 2.2. Установка B4N Controller

Перед началом установки B4N Controller необходимо авторизоваться в репозитории brain4net на docker hub, при помощи следующей команды:

```
docker login -u `USERNAME` -p `PASSWORD`
```

Далее потребуется ввести, полученные при приобретении решения, логин ('USERNAME') и пароль ('PASSWORD'). После авторизации, установка производится так же средствами Docker:

```
docker run -d --net=host -m 8192M --memory-swap=8192M --restart=always --log-opt max-size=100m --volumes-from ctl-dbs -e JAVA_OPTS="-Dmongo.host=127.0.0.1 -Dmongo.port=27016" --cpuset-cpus=2-5 --name ctl-brain4net/ctlsp-v2:release
```

Если в сети присутствуют коммутаторы, не поддерживающие `ethertype 0x88a8`, к `JAVA_OPTS`, необходимо добавить `-Deth.type=0x8100`.

Эти операции необходимо выполнить на всех серверах, предназначенных для установки контроллеров. На этом установка будет завершена. Для сборки кластера перейдите к графическому интерфейсу B4N Orchestrator (См. "Operation manual").

---

## УПРАВЛЯЮЩИЕ КОМАНДЫ

## 3. УПРАВЛЯЮЩИЕ КОМАНДЫ

Раздел содержит список команд, позволяющих управлять операциями B4N Controller:

- Запуск B4N Controller;
- Остановка B4N Controller;
- Перезагрузка B4N Controller;
- Upgrade и Downgrade B4N Controller;
- Логирование B4N Controller;
- Устранение неполадок B4N Controller.

### ЗАПУСК B4N CONTROLLER

---

Для запуска B4N Controller выполните следующую операцию:

```
docker run -d restart=always -p 830:830 -p 6653:6653 -p 7800:7800 --cpuset-cpus B4NCTL_CP --memory 8GB --memory-swap -1 --add-host dockerhost:$B4NCTL_IP --link ctl-db-data --name ctl --hostname $B4NCTL_HN $B4NCTL_RP
```

### ОСТАНОВКА B4N Controller

---

Для остановки B4N Controller выполните следующую команду:

```
docker stop ctl
```

### ПЕРЕЗАГРУЗКА B4N Controller

---

Для перезагрузки B4N Controller выполните следующую команду:

```
docker restart ctl
```

### UPGRADE/DOWNGRADE B4N Controller

---

#### АКТУАЛЬНАЯ ВЕРСИЯ

---

Обновление B4N Controller до актуальной версии происходит по следующей команде:

```
docker rm -f -v ctl
docker pull $B4NCTL_RP
docker run -d --restart=always -p 830:830 -p 6653:6653 -p 7800:7800 --cpuset-cpus B4NCTL_CP --memory 8GB --memory-swap -1 --add-host dockerhost:$B4NCTL_IP --link ctl-db-data --name ctl --hostname $B4NCTL_HN $B4NCTL_RP
```

#### ОПРЕДЕЛЁННАЯ ВЕРСИЯ

---

Обновление B4N Controller до определённой версии происходит по следующей команде:

```
docker rm -f -v ctl
docker pull $B4NCTL_RP:<version>
docker run -d --restart=always -p 830:830 -p 6653:6653 -p 7800:7800 --cpuset-cpus B4NCTL_CP --memory 8GB --memory-swap -1 --add-host dockerhost:$B4NCTL_IP --link ctl-db-data --name ctl --hostname $B4NCTL_HN $B4NCTL_RP:<version>
```

Например, для перехода на версию 1.0.272 выполните следующее:

```
docker pull $B4NCTL_RP:1.0.272
docker run -d --restart=always -p 830:830 -p 6653:6653 -p 7800:7800 --cpuset-cpus B4NCTL_CP --memory 8GB --memory-swap -1 --add-host dockerhost:$B4NCTL_IP --link ctl-db-data --name ctl --hostname $B4NCTL_HN $B4NCTL_RP:1.0.272
```

### ЛОГИРОВАНИЕ B4N Controller

---

#### STDOUT

---

```
docker logs ctl
```

STDOUT FOLLOWING

---

```
docker logs -f ctl
```

ФАЙЛОВЫЕ ОПЕРАЦИИ (TAIL F, CAT, GREP, ETC...)

---

```
docker exec ctl <operation> /opt/b4n/ctl/logs/ctl.log
```

Например, для tail-f, операция будет выглядеть следующим образом:

```
docker exec ctl tail -f /opt/b4n/ctl/logs/ctl.log
```

УСТРАНЕНИЕ НЕПОЛАДОК

---

Для устранения неполадок в работе зайдите в docker и выполните любые задачи администрирования при помощи следующей команды:

```
docker exec -it ctl bash
```

117342, г. Москва, ул. Бутлерова, д.17 (БЦ NEO GEO), корпус А, офис 285  
Тел.: +7.499.649.4646  
E-mail: [need@brain4netru](mailto:need@brain4netru)  
Web: <http://www.brain4netru>



**Brain4Net**  
Программируемые Сети