



СТІ Мониторинг
Руководство по установке

Версия: 1.0
Редакция: 18.10.2024

Оглавление

1. Контакты	3
2. Установка программного продукта СТИ Мониторинг	4
2.1 Важные замечания	4
2.2 Технологический стек	4
2.3 Обобщенная схема решения	5
2.4 Требования к VM с СТИ Мониторинг	5
2.5 Установка	6
2.6 Доступы в СТИ Мониторинг UI	6
2.6.1 Шифрование и редактирование файлов inventories	6
3. Подготовка виртуальных машин к установке ПО	8
3.1 Дополнительные манипуляции с виртуальными машинами	9
3.1.1 Версия Python для работы с Ansible	9
3.1.2 Проблема с запуском asdf	9
4. Установка и удаление Hashicorp Nomad	10
4.1 Установка Hashicorp Nomad	10
4.2 Удаление Hashicorp Nomad	10
4.3 Остановка и запуск Hashicorp Nomad	10
4.4 Проверка статуса Hashicorp Nomad	10
4.4.1 Через консоль и Ansible	10
4.4.2 Через Web-интерфейс	10
5. Установка и удаление СТИ Мониторинг	12
5.1 Установка СТИ Мониторинг	12
5.2 Очистка ClickHouse DB	12
5.3 Очистка PostgreSQL DB	13
5.4 Удаление СТИ Мониторинг	13
6. Установка и удаление агента opentelemetry-collector-contrib на VM с TrueConf на ОС Linux	14
6.1 Установка opentelemetry-collector-contrib	14
6.2 Удаление opentelemetry-collector-contrib	14
7. Установка и удаление агента opentelemetry-collector-contrib на VM с TrueConf на ОС Windows	14
7.1 Установка opentelemetry-collector-contrib	14
7.2 Удаление opentelemetry-collector-contrib	14
8. Установка и удаление агента Vector на VMs с TrueConf	15
8.1 Установка Vector	15
8.2 Удаление Vector	15

1. Контакты

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 СТИ. Правообладатель оставляет за собой возможность изменять, дополнять и/или производить другие действия с данным документом без уведомления пользователя. СТИ Мониторинг является зарегистрированной торговой маркой программного обеспечения, разрабатываемого компанией СТИ. Все торговые марки и названия программ являются собственностью их владельцев.

Телефон: +7.495.784.73.13 (система распознавания русской речи)

Техническая поддержка: +7.495.784.73.11

Многоканальный телефон: +7.800.550.43.57

E-mail: support@cti.ru

Web: www.cti.ru

Содержание

- [Обзор системы](#)
- [Подготовка виртуальных машин](#)
- [Установка оркестратора](#)
- [Установка СТИ Мониторинг](#)
- [Установка OpenTelemetry Collector](#)
- [Установка Vector](#)

2. Установка программного продукта **СТИ Мониторинг**

2.1 Важные замечания

Поставляемый установочный пакет позволяет произвести установку СТИ Мониторинг без использования подключения к сети Интернет.

Все виртуальные машины (далее - VM) должны работать под управлением ОС семейства Linux, в том числе и VM, с которой производится установка.

Необходимо, чтобы на VM с СТИ Мониторинг был установлен Docker.

На VM, с которой осуществляется установка, должна присутствовать утилита GNU Make: `make -v`

Проект содержит автономные копии следующего ПО:

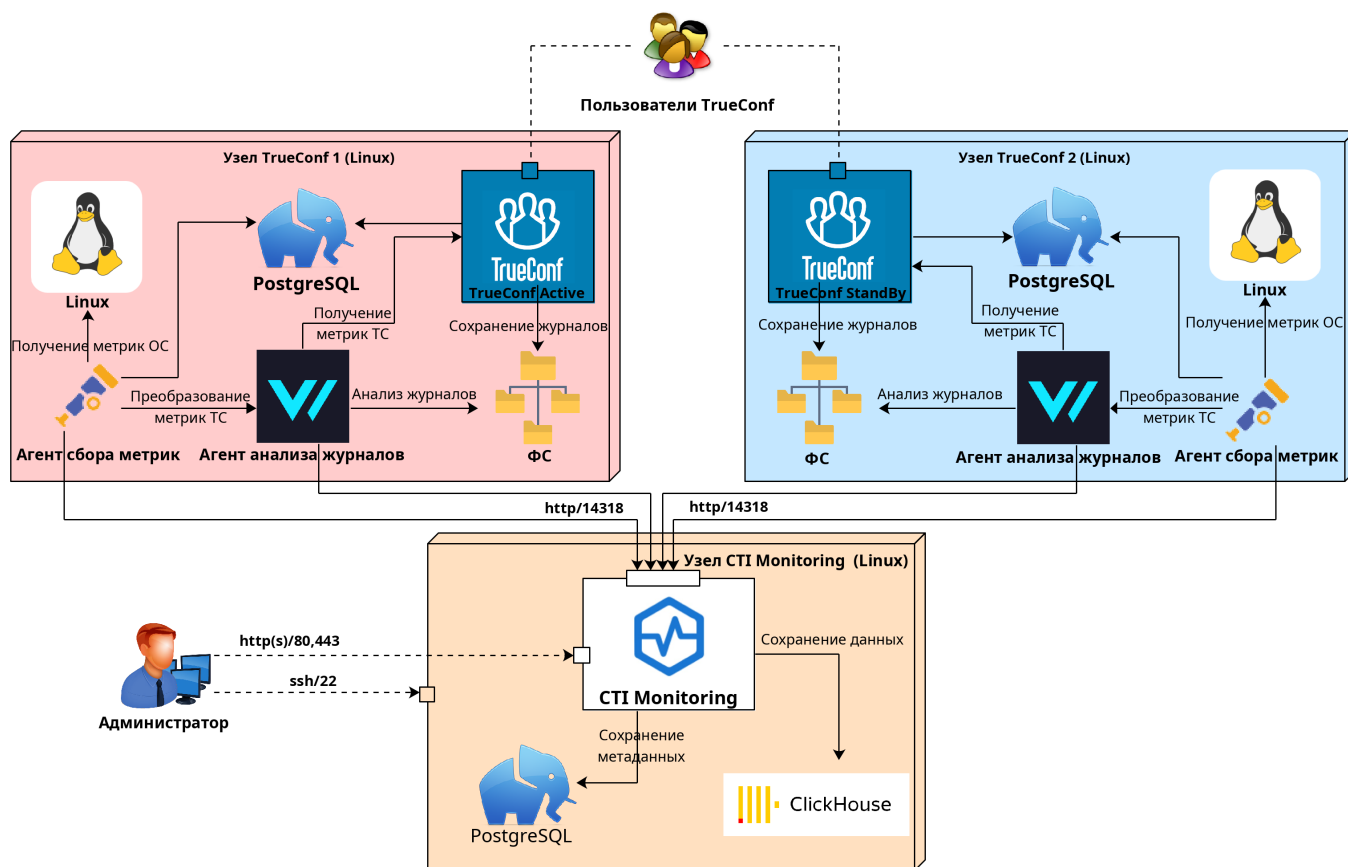
- **Hashicorp Nomad** -> `playbooks/roles/ansible-role-nomad/files/nomad`
- **Python + Ansible** -> `./asdf-ansible.tar.gz`, который при первом запуске команд распаковывается в `./asdf-ansible`
- **Vector** -> `playbooks/packages/vector-X.X.X-x86_64-unknown-linux-gnu.tar.gz`
- **Opentelemetry-collector-contrib** -> `playbooks/packages/otelcol-contrib_X.X.X_linux_amd64.tar.gz`
- **5 автономных docker-образов для самой системы мониторинга** -> `./playbooks/images/...` -> `cti-monitoring`, `clickhouse`, `postgresql`, `vector`, `otelcontrib-collector`

Всё ПО не зависит от дистрибутива Linux и представлено в максимально автономном виде, подходящем для любой версии Linux.

2.2 Технологический стек

- **Установка:** Ansible (Python)
- **Оркестратор:** Hashicorp Nomad (Golang)
- **Единый UI для метрик, логов и телеметрии:** СТИ Мониторинг (Golang/VueJS)
- **Агент сбора данных:** opentelemetry-collector-contrib - (Golang)
- **Агент сбора данных:** Vector - (Rust)

2.3 Обобщенная схема решения



2.4 Требования к VM с CTI Мониторинг

В систему мониторинга будут приходить большое количество данных от узлов TrueConf, поэтому данная VM должна удовлетворять следующим требованиям:

Минимальные требования к оборудованию для установки решения следующие:

- CPU: 8 ядер
- RAM: 8 Гб
- HDD: 100 Гб (размер зависит от количества данных и периода хранения)
- Сеть: 1 Гбит/с
- ОС: Linux (Astra, RED OS)

Рекомендуемые требования к оборудованию для установки решения следующие:

- Процессор: 16+ ядер
- Память: 16+ Гб
- Диск: 512+ Гб (объем данных зависит от интенсивности использования TrueConf и периода хранения данных, который по умолчанию равен 30 дням)
- Сеть: 1 Гбит/с

- ОС: Linux (Astra, RED OS)

2.5 Установка

Сначала распаковываем архив с дистрибутивом и переходим в целевой каталог:

```
tar -xf trueconf-monitoring-xxx.tar.gz
cd ./trueconf-monitoring
```

Внимание: перед установкой в новое окружение, необходимо сделать копию файла `inventories/tc-vms.ini` под другим именем и в нем тщательно проставить IP адреса всех VMs. Он по умолчанию зашифрован. Как его расшифровать написано ниже в данном файле.

После замены адресов, необходимо скорректировать строку:

```
inventory = inventories/tc-vms.ini
```

В `./ansible.cfg`.

Список всех доступных команд можно получить через запуск: `make`.

После этого можно переходить к следующему этапу и проводить установку ПО.

- Подготовка виртуальных машин к установке ПО - при наличии доступа к интернет
- Установка Hashicorp Nomad
- Установка и удаление CTI Мониторинг через Nomad на отдельную VM
- Установка и удаление Opentelemetry-collector-contrib на VM с Trueconf
- Установка и удаление Vector на VM с Trueconf

2.6 Доступы в CTI Мониторинг UI

По умолчанию создается пользователь со следующими логином и паролем:

Доступы:

- **логин:** `tc-mon@cti.ru`
- **пароль:** `trueconf`

Дистрибутивы будут находиться в каталоге `./dist`

2.6.1 Шифрование и редактирование файлов inventories

В целях безопасности, все файлы `inventories` зашифрованы через `ansible-vault`. Пароль поставляется вместе с установочным пакетом.

Для шифрования файлов в `inventories` используются команды:

```
./local-asdf exec ansible-vault encrypt ./inventories/tc-vms.ini
```

Для редактирования файлов в `inventories` используются команды:

```
export EDITOR=nano
./local-asdf exec ansible-vault edit ./inventories/tc-vms.ini
```

Для дешифрования файлов в `inventories` используются команды:

```
./local-asdf exec ansible-vault decrypt ./inventories/tc-vms.ini
```

После редактирования файлов необходимо всегда их зашифровать обратно для безопасности

3. Подготовка виртуальных машин к установке ПО

Данный раздел является не обязательным при установке ПО в продуктивной среде

Обязательно необходимо только расширить лимиты по файловым дескрипторам, во избежание ошибок: `too many open files`

Для выполнения команд необходим доступ к интернет!

Перед установкой целевого ПО необходимо привести все VMs к общему знаменателю:

- обновить пакеты и перезапустить виртуальные машины
- поставить следующие пакеты: `mc htop iotop atop nmap nethogs ncdu nano tree`
- поставить везде `timesyncd` для синхронизации времени
- применить оптимизированные настройки ядра Linux - `sysctl.conf`
- применить новые лимиты по открытым файлам: `limits.conf, sysctl.conf`

Лучше всего, чтобы кластер виртуальных машин был однородным в плане операционных систем.

За эту работу отвечает playbook: [vms_preparation.yml](#) Перечень целевых виртуальных машин находится тут: [inventories/tc-vms.ini](#)

Для того, чтобы обновить текущие пакеты и перезапустить все виртуальные машины используется playbook: [ansible-os-update](#)

Последовательность действий: Все команды выполняются из корня проекта !

```
# добавляем все виртуальные машины в соответствующий inventories/tc-vms.ini
nano -w inventories/tc-vms.ini
# если ещё не положили ssh-ключи, то это обязательно надо сделать на текущем шаге:
ssh-copy-id rnd@172.18.1.61
ssh-copy-id rnd@172.18.1.62
ssh-copy-id rnd@172.18.1.63
# проверяем, что Ansible успешно видит все виртуальные машины, выполнив любую команду на них, например:
./local-asdf exec ansible all -a "uname -a"
# НЕ ОБЯЗАТЕЛЬНО: затем обновляем текущие пакеты и перезапускаем VM:
./local-asdf exec ansible-playbook -b -e hosts=monitoring -e reboot_host=true playbooks/roles/ansible-os-update/main.yml
# НЕ ОБЯЗАТЕЛЬНО: после рестарта VM, устанавливаем дополнительное ПО, в том числе и сервер синхронизации времени:
./local-asdf exec ansible-playbook -b -e hosts=monitoring playbooks/vms_preparation.yml
# если установка дополнительного ПО не нужна, то только увеличиваем максимальное количество файловых дескрипторов:
./local-asdf exec ansible-playbook -e hosts=monitoring --start-at-task="Copy sysctl.conf" playbooks/vms_preparation.yml
```

После выполнения всех этих команд все виртуальные машины должны быть готовы к дальнейшему развертыванию

3.1 Дополнительные манипуляции с виртуальными машинами

3.1.1 Версия Python для работы с Ansible

Перед установкой решения необходимо указать версию Python для работы с Ansible

Есть два решения: * `./local-asdf global python 3.11.8` - сделать для python глобальную версию через asdf 3.11.8 * в файле `tool-versions` в домашнем каталоге пользователя добавить строчку `python 3.11.8`

3.1.2 Проблема с запуском asdf

Если после распаковки архива и попытки запустить Ansible через asdf выдается ошибка `libffi.so.6 cannot open shared object file: no such file or directory`, то стоит сделать следующие действия:

- `find /usr/lib64/ -name "libffi*"` или `find /usr/lib/ -name "libffi*"` - найти libffi на виртуальной машине вывод должен быть примерно такой:

```
/usr/lib64/libffi.so.6
/usr/lib64/libffi.so.8
/usr/lib64/pkgconfig/libffi.pc
/usr/lib64/libffi.so.8.1.0
/usr/lib64/libffi.so
```

- `sudo ln -s /usr/lib64/libffi.so.8 /usr/lib64/libffi.so.6` - сделать символическую ссылку с нужной на рабочую версию libffi

4. Установка и удаление

Если ещё не положили ssh-ключи - то это обязательно надо сделать сейчас.

Пример:

```
ssh-copy-id rnd@172.18.1.61
ssh-copy-id rnd@172.18.1.62
ssh-copy-id rnd@172.18.1.63
```

4.1 Установка

Для установки nomad должен быть установлен `docker`, в частности пакеты: - `docker-ce` - `docker-ce-cli` - `docker-ce-rootless-extras` - `containerd.io` (не требуется для RedOs)

```
# Затем устанавливаем HashiCorp Nomad на VMs в группе nomad
make nomad_install
```

4.2 Удаление

```
make nomad_uninstall
```

4.3 Остановка и запуск

```
make nomad_stop
make nomad_start
```

4.4 Проверка статуса

4.4.1 Через консоль и Ansible

```
make nomad_status
```

4.4.2 Через Web-интерфейс

Web интерфейс Nomad доступен на порту 4646 на целевой VM.

Пример: <http://tc-monitoring.cti.ru:4646/>

5. Установка и удаление

Если ещё не положили ssh-ключи, то это обязательно надо сделать на текущем шаге.

Пример:

```
ssh-copy-id rnd@172.18.1.61
ssh-copy-id rnd@172.18.1.62
ssh-copy-id rnd@172.18.1.63
```

5.1 Установка

```
# Устанавливаем ядро CTI Мониторинг
make deploy_monitoring
```

После установки заходим по http на порт 4646 и убеждаемся, что все запустилось корректно:

Jobs

 × Type ▾ Node Pool ▾ Status ▾ Datacenter ▾ Prefix ▾ Run Job

Name	Status	Type	Node Pool	Priority	Summary
<u>cti-monitoring</u>	RUNNING	service	default	50	<div style="width: 10%;"></div>
<u>postgres</u>	RUNNING	service	default	50	<div style="width: 20%;"></div>
<u>clickhouse</u>	RUNNING	service	default	50	<div style="width: 30%;"></div>
<u>nginx</u>	RUNNING	service	default	50	<div style="width: 40%;"></div>
<u>vector</u>	RUNNING	system	default	50	<div style="width: 50%;"></div>
<u>opentelemetry_collector</u>	RUNNING	system	default	50	<div style="width: 60%;"></div>

Per page ▾ 1 - 6 of 6

5.2 Очистка ClickHouse DB

Использовать в крайних случаях. После этого лучше перезапустить компонент cti-monitoring внутри Nomad

```
make ch_reset
```

5.3 Очистка PostgreSQL DB

Использовать в крайних случаях. После этого обязательно перезапустить sti-monitoring job внутри Nomad

```
make pg_reset
```

5.4 Удаление

```
make undeploy_monitoring
```

6. Установка и удаление агента opentelemetry-collector-contrib на VM с TrueConf на ОС Linux

6.1 Установка opentelemetry-collector-contrib

```
make otel_install
```

6.2 Удаление opentelemetry-collector-contrib

```
make otel_uninstall
```

7. Установка и удаление агента opentelemetry-collector-contrib на VM с TrueConf на ОС Windows

7.1 Установка opentelemetry-collector-contrib

Необходимо распаковать и разместить в папке C:\Program Files\OpenTelemetry Collector файлы из архива, содержащего выполняемый файл otelcol-contrib.exe и файл конфигурации config.yaml. В файле конфигурации необходимо указать IP-адрес сервера (VM), на котором развернут STI Мониторинг.

Для создания службы агента необходимо в консоли от имени администратора выполнить команду:

```
New-Service -Name OpenTelemetryCollectorService -BinaryPathName '"C:\Program Files\OpenTelemetry Collector\otelcol-contrib.exe"' -DisplayName "OpenTelemetry Collector service" -Description "OpenTelemetry Collector service" -StartupType "Automatic"
```

7.2 Удаление opentelemetry-collector-contrib

Остановите службу агента OpenTelemetry Collector и удалите папку C:\Program Files\OpenTelemetry Collector.

8. Установка и удаление агента на VMs с TrueConf

Необходимо создать файлы `~/.pgpass` на каждом сервере с TrueConf, чтобы скрипты обогащения данных не передавали пароль в открытом виде:

```
nano -w ~/.pgpass  
chmod 600 ~/.pgpass
```

Пример файла `~/.pgpass` :

```
172.18.1.61:5444:tcs:postgres:postgres
```

8.1 Установка

```
make vector_install
```

8.2 Удаление

```
make vector_uninstall
```