Программное обеспечение

«Servicepipe Cybert»

Инструкция по установке программного обеспечения

Москва — 2026 г.

Оглавление

1 Общие сведения

1.1 Назначение документа

Настоящий документ содержит инструкцию по установке программного обеспечения «Servicepipe Cybert» (далее - ПО «Servicepipe Cybert», Cybert, программное обеспечение) в он-прем варианте поставки в инфраструктуре заказчика или private cloud.

Документ описывает подготовку установочного комплекта, выполнение первичной настройки установщика, запуск установки, проверку результата, порядок обновления и удаления компонентов программного обеспечения.

Инструкция подготовлена для администраторов, инженеров эксплуатации, DevOps/SRE-специалистов и иных уполномоченных сотрудников, выполняющих развертывание и сопровождение Servicepipe Cybert.

1.2 Сведения о программном обеспечении

ПО «Servicepipe Cybert» предназначено для защиты веб-ресурсов и приложений от вредоносного автоматизированного трафика и прикладных DDoS-атак. Программное обеспечение анализирует сетевые, транспортные, TLS- и HTTP-признаки запросов, поведение клиента в пределах сессии и результаты дополнительных клиентских проверок, после чего формирует решение о пропуске, блокировании, ограничении интенсивности или дополнительной проверке запроса.

В он-прем варианте компоненты Cybert размещаются в инфраструктуре заказчика. На filter nodes фронтенд-сервер NGINX/Angie с модулем Cybert принимает HTTP(S)-трафик и передает контекст запросов в Cresp. Cresp выполняет классификацию, применяет профили и пользовательские правила и возвращает вердикт, который применяется на фронтенд-сервере. Разрешенные запросы передаются защищаемому приложению.

Управление функциями Cybert выполняется через административный контур и API. В типовую on-prem поставку входят Cresp и связанные runtime-компоненты, модуль Cybert для NGINX/Angie, Cresp API и сервис доставки конфигурации, компоненты панели управления, реляционные и аналитические хранилища, системы метрик и журналирования, а также установочные сценарии для развертывания и сопровождения продукта.

1.3 Область применения инструкции

Настоящая инструкция описывает online-установку Servicepipe Cybert из готового online installer archive. Такой архив содержит установщик, inventory-шаблоны, playbook-и, переменные и служебные файлы, но не обязательно содержит полный набор RPM-пакетов, архивов сервисов и дополнительных бинарных файлов компонентов.

В процессе установки целевые хосты получают RPM-пакеты из настроенного DNF repository, а недостающие архивы, файлы поставки и container images скачиваются из raw repository или иного хранилища артефактов, указанного в параметрах установщика.

1.4 Термины и сокращения

Термин	Описание
Controller-хост	Хост, с которого оператор запускает Ansible-установщик и выполняет команды установки, обновления и удаления. Не обязательно входит в состав целевого кластера Cybert.
Целевой хост	Сервер инфраструктуры заказчика, на который устанавливаются компоненты Servicepipe Cybert.
Mgmt nodes	Группа управляющих узлов, на которых размещаются Web Console/backend, Cresp API, Cresp config deliver, базы данных, компоненты журналирования, метрик и иные служебные сервисы.
Filter nodes / runtime nodes	Узлы обработки HTTP(S)-трафика. В типовой on-prem схеме на них размещаются NGINX/Angie с модулем Cybert, Cresp, Cresp-BH и агенты журналирования/метрик.
Online installer archive	Архив установщика, предназначенный для online-развертывания. В текущей схеме поставки может использоваться имя вида cybert-onprem-ansible-online-<version>.tar.gz; фактическое имя определяется комплектом поставки и release manifest.
DNF repository	Репозиторий RPM-пакетов, из которого целевые хосты получают пакеты во время установки и обновления.
Raw repository	Хранилище архивов, файлов поставки и container images, которые не входят в online installer archive и скачиваются во время установки.
Vault password	Пароль Ansible Vault, используемый для шифрования и чтения сгенерированных секретов установщика.
Playbook	Ansible-сценарий, выполняющий установку, обновление или удаление конкретного компонента либо всего комплекта.

2 Модель установки и состав поставки

2.1 Общая модель online-установки

Online-установка Servicepipe Cybert выполняется с controller-хоста с использованием Ansible. Оператор распаковывает online installer archive, запускает bootstrap для формирования рабочей конфигурации установщика и описания топологии, после чего выполняет общий либо компонентные Ansible playbook-и.

Процесс установки включает следующие основные этапы:

- подготовка controller-хоста и проверка доступа к целевым хостам;
- проверка доступности DNF repository и raw repository;
- распаковка online installer archive;
- первичная настройка установщика с помощью bootstrap.sh;
- формирование inventory, group vars, vault vars и команд запуска;
- запуск установки всех выбранных компонентов или отдельных компонентов;
- проверка состояния системных сервисов, контейнеров и служебных HTTP-endpoint-ов;
- сохранение установочных материалов и сведений о выполненной установке.

2.2 Состав установочного комплекта

Online installer archive содержит файлы, необходимые для управления процессом установки, но не является полным offline-репозиторием артефактов. Конкретный состав архива зависит от версии поставки, но в общем случае включает:

- скрипт bootstrap.sh для первичной настройки установщика;
- Ansible playbook-и для установки, обновления и удаления компонентов;
- inventory-шаблоны и переменные групп хостов;
- шаблоны конфигурационных файлов;
- служебный файл .installer.env с параметрами репозиториев и именами артефактов;
- manifest версии и сведения о составе комплекта поставки;
- документацию и вспомогательные файлы установщика.

В текущей схеме поставки в примерах используется каталог установщика cybert-onprem-ansible. Если в конкретной поставке каталог или архив имеют другое имя, необходимо использовать фактическое имя из release manifest и комплекта поставки.

2.3 Основные компоненты установки

Состав устанавливаемых компонентов определяется release manifest, топологией стенда и выбранными ролями узлов. Ниже приведен типовой состав компонентов on-prem поставки Servicepipe Cybert. Часть runtime-компонентов может устанавливаться общим профилем filter nodes, а инфраструктурные и management-компоненты - отдельными компонентными playbook-ами.

Компонент	Назначение
NGINX / Angie + модуль Cybert	Фронтенд-сервер и собственный модуль интеграции Cybert. Принимают HTTP(S)-трафик, формируют контекст запроса, взаимодействуют с Cresp и применяют полученный вердикт.
Cresp	Собственный центр принятия решений Servicepipe: классификация запросов, применение профилей, правил, rate limit и логики защиты.
Cresp-BH	Компонент получения L3/L4-телеметрии и дополнительного контекста сетевой нагрузки для поддерживаемых сценариев защиты.
Cresp API / Cresp config deliver	Контур управления профильной конфигурацией: прием изменений, формирование итоговой runtime-конфигурации и доставка ее на filter nodes.
MariaDB	Хранилище конфигурационных данных Cresp API, на основании которых формируется runtime-конфигурация Cybert.
ClickHouse cluster	Хранение и обработка детальных журналов запросов, решений защиты и аналитических данных.
Kafka stack	Брокер событий и журналов, используемый для потоковой передачи данных между компонентами.
VictoriaMetrics stack	Единое хранилище временных рядов Cybert: прикладных, системных и сервисных метрик.
vmagent	Агент доставки временных рядов в VictoriaMetrics, в том числе с filter/runtime nodes.
Vector	Агент обработки и доставки журналов. Может размещаться на mgmt nodes и filter nodes в соответствии с топологией.
PostgreSQL HA cluster	Отказоустойчивый кластер базы данных административного контура и backend панели управления.
LK docker stack	Контейнерный контур панели управления Servicepipe Cybert, включая frontend/backend и связанные API-компоненты.

Local-repo client configuration	Настройка доступа целевых хостов к используемым репозиториям пакетов и артефактов.

Примечание. Фактический перечень компонентов может быть расширен или сокращен в зависимости от версии Servicepipe Cybert, выбранной топологии и состава поставки. Приоритетным источником состава версии, имен пакетов, сервисов и playbook-ов является release manifest, входящий в комплект поставки.

2.4 Роли узлов и пример топологии

Перед установкой необходимо определить роли узлов и их сетевые адреса. В минимальной инструкции используются группы mgmt nodes и filter nodes. На mgmt nodes размещаются административный контур, Cresp API и Cresp config deliver, PostgreSQL/MariaDB, хранилища журналов и метрик и иные служебные сервисы. На filter nodes размещается runtime-контур обработки трафика: NGINX/Angie с модулем Cybert, Cresp, Cresp-BH и необходимые агенты наблюдаемости.

В примерах используются условные адреса:

```
mgmt1   10.250.10.11
mgmt2   10.250.10.12
mgmt3   10.250.10.13
filter1 10.250.20.11
filter2 10.250.20.12
filter3 10.250.20.13
```

Примечание. Перед использованием инструкции замените все условные адреса, имена пользователей, версии архивов, URL репозиториев и имена компонентов на фактические значения, утвержденные для конкретного стенда.

3 Предварительные требования

3.1 Требования к controller-хосту

Перед запуском установки на controller-хосте должны быть выполнены следующие условия:

- получен готовый архив cybert-onprem-ansible-online-<version>.tar.gz или архив с фактическим именем, указанным в комплекте поставки;
- установлены ansible, python3 и ssh;
- имеется SSH-доступ до всех целевых хостов;
- SSH-пользователь, используемый для установки, может выполнять sudo на целевых хостах;
- controller-хост имеет доступ к DNF repository и raw repository, если скачивание артефактов выполняется через controller-хост;
- известен vault password, который будет использоваться для сгенерированных секретов;
- доступно достаточное дисковое пространство для распаковки архива, хранения inventory, vault-файлов, временных файлов и журналов выполнения установки.

3.2 Требования к целевым хостам

На целевых хостах должны быть выполнены следующие условия:

- операционная система GNU/Linux, архитектура x86-64 и системные пакеты соответствуют требованиям конкретной версии Servicepipe Cybert;
- настроен сетевой доступ между узлами в соответствии с выбранной топологией;
- SSH-сервис доступен с controller-хоста;
- пользователь установки имеет права sudo либо оператор готов вводить пароль sudo при запуске playbook-ов;
- целевые хосты имеют доступ к DNF repository и raw repository, если артефакты скачиваются непосредственно с целевых хостов;
- системное время синхронизировано, так как корректная синхронизация времени важна для TLS, журналов, метрик и корреляции событий;
- на узлах подготовлены сетевые интерфейсы, маршруты, DNS-резолвинг и firewall-правила, необходимые для работы компонентов.
- между mgmt nodes и filter nodes разрешено межкомпонентное взаимодействие, необходимое для Cresp API, доставки runtime-конфигурации, RPC-взаимодействия модуля Cybert с Cresp, передачи журналов и метрик;
- для filter nodes подготовлена поддерживаемая версия NGINX/Angie и параметры интеграции с защищаемыми приложениями в соответствии с проектом внедрения.

3.3 Сетевой доступ к репозиториям

В режиме online целевые хосты получают RPM-пакеты и дополнительные артефакты во время выполнения установки. До запуска bootstrap и playbook-ов необходимо проверить доступность используемых репозиториев.

Проверка доступности DNF repository может выполняться следующими командами на целевых хостах:

```
dnf clean all
dnf makecache
dnf repoquery <package-name>
```

Проверка доступности raw repository может выполняться следующей командой:

```
curl -I <raw-repository-url>/<file-name>
```

3.4 Данные, необходимые перед началом установки

До начала установки рекомендуется подготовить и согласовать следующие данные:

- версию устанавливаемого комплекта Servicepipe Cybert;
- имя и путь к online installer archive;
- состав компонентов, которые должны быть установлены;
- список mgmt nodes и filter/runtime nodes;
- SSH-пользователя для установки;
- порядок использования sudo: без пароля или с вводом пароля;
- URL DNF repository и raw repository;
- vault password для шифрования сгенерированных секретов;
- сетевые параметры стенда, включая адреса, DNS-имена, VIP-адреса, маршруты и правила доступа;
- версии NGINX/Angie и runtime-компонентов Cybert, предусмотренные release manifest;

- параметры размещения Cresp, Cresp-BH, Cresp API, Cresp config deliver и MariaDB по группам узлов;
- контакт ответственного администратора и порядок фиксации результата установки.

4 Подготовка установочного комплекта

4.1 Получение и распаковка online-архива

Скопируйте online installer archive на controller-хост и распакуйте его в рабочий каталог, из которого будет выполняться установка.

```
tar -xzf cybert-onprem-ansible-online-<version>.tar.gz
cd cresp-onprem-ansible
```

Если архив или рабочий каталог имеют другое имя, используйте фактические значения из комплекта поставки. Все последующие команды должны выполняться из рабочей директории установщика, если не указано иное.

4.2 Проверка рабочей директории установщика

После распаковки проверьте наличие основных файлов и каталогов установщика. Типовая структура рабочей директории включает bootstrap.sh, каталог playbooks, каталог inventory, шаблоны переменных и служебный файл .installer.env.

```
ls -la
ls -la playbooks
ls -la inventory
```

Перед внесением изменений в конфигурацию рекомендуется сохранить исходное состояние распакованного каталога либо зафиксировать контрольную сумму полученного архива.

5 Первичная настройка установщика

5.1 Назначение bootstrap

Bootstrap подготавливает рабочие файлы установщика: inventory, group vars, vault vars, playbook-и для выбранных компонентов, секреты и команды запуска. Bootstrap необходимо запускать перед первой установкой.

Повторный запуск bootstrap требуется только при изменении топологии, состава компонентов, базовых настроек установки или при обновлении служебных параметров установщика.

5.2 Запуск bootstrap

Для запуска первичной настройки выполните команду из рабочей директории установщика:

```
./bootstrap.sh
```

В начале bootstrap оператор вводит vault password. Этот пароль сохраняется локально в файле .ansible/vault-password и используется для шифрования сгенерированных vault-файлов.

5.3 Vault password и сгенерированные секреты

Vault password является чувствительным значением. Доступ к файлу .ansible/vault-password должен быть ограничен только уполномоченными администраторами. Не рекомендуется передавать этот файл по незащищенным каналам связи или хранить его в общедоступных каталогах.

Если vault password утрачен, прочитать ранее сгенерированные vault-файлы стандартными средствами Ansible Vault будет невозможно. Поэтому пароль и порядок его хранения должны быть определены до начала установки.

5.4 Настройка топологии и состава компонентов

Во время bootstrap установщик задает вопросы о режиме установки, необходимости настройки online DNF repository, топологии и составе компонентов. В типовой поставке выбираются инфраструктурные компоненты ClickHouse, Kafka, VictoriaMetrics, vmagent, Vector, PostgreSQL HA и LK docker stack. Установка Cresp, Cresp-BH, модуля Cybert для NGINX/Angie, Cresp API, Cresp config deliver и MariaDB выполняется ролями и playbook-ами, предусмотренными release manifest конкретной версии.

Если для нескольких компонентов используются общие mgmt nodes, оператор может определить их один раз и затем использовать как источник узлов для компонентных блоков. Аналогично определяются filter nodes, на которых размещается runtime-контур Cybert и агенты наблюдаемости. Фактический набор вопросов bootstrap может отличаться между релизами.

5.5 Пример выполнения bootstrap

Ниже приведен пример начала bootstrap и одного инфраструктурного компонентного блока. Он иллюстрирует формирование inventory и vault-файлов; конкретный набор вопросов и блоков определяется версией installer-a и release manifest.

```
$ ./bootstrap.sh
Vault password for generated vault files:
====================================
on-prem Ansible bootstrap
====================================

Installation mode: online
Do you need to configure online DNF repository? [yes]:
Do you need ClickHouse cluster? [yes]:
Do you need Kafka stack? [yes]:
Do you need VictoriaMetrics stack? [yes]:
Do you need vmagent on filter nodes? [yes]:
Do you need Vector on mgmt nodes? [yes]:
Do you need PostgreSQL HA cluster? [yes]:
Do you need LK docker stack? [yes]:

Do you want to define common mgmt nodes for components? [yes]:
Mgmt node 1 IP/hostname: 10.250.10.11
Mgmt node 2 IP/hostname: 10.250.10.12
Mgmt node 3 IP/hostname: 10.250.10.13

Do you want to define common filter nodes for vmagent? [yes]:
Filter node 1 IP/hostname: 10.250.20.11
Filter node 2 IP/hostname: 10.250.20.12
Filter node 3 IP/hostname: 10.250.20.13

>>> Running ClickHouse cluster bootstrap...
====================================
 ClickHouse cluster bootstrap
====================================
```

```
ClickHouse Keeper mode: embedded/standalone [embedded]:

ClickHouse nodes source: mgmt/custom [mgmt]:

Writing inventory file:
  /home/operator/cresp-onprem-ansible/inventory/clickhouse.ini
Writing group vars:
  /home/operator/cresp-onprem-ansible/inventory/group_vars/clickhouse.yml
Writing vault vars:
  /home/operator/cresp-onprem-ansible/inventory/group_vars/clickhouse_vault.yml
Writing playbook:
  /home/operator/cresp-onprem-ansible/playbooks/deploy-clickhouse.yml
Encrypt ClickHouse vault file:
  ansible-vault encrypt
/home/operator/cresp-onprem-ansible/inventory/group_vars/clickhouse_vault.yml

ClickHouse bootstrap completed.

Bootstrap completed.

Vault password file:
  /home/operator/cresp-onprem-ansible/.ansible/vault-password

ansible-playbook playbooks/deploy.yml -u {user} --vault-password-file /home/operator/cresp-
onprem-ansible/.ansible/vault-password
```

5.6 Проверка inventory после bootstrap

После завершения bootstrap проверьте список целевых хостов и групп inventory:

```
ansible-inventory --graph
```

Если в выводе отсутствует ожидаемый хост или хост попал не в ту группу, необходимо скорректировать настройки установщика либо повторить bootstrap с правильной топологией.

6 Установка программного обеспечения

6.1 Установка всех выбранных компонентов

Для установки всех компонентов, выбранных во время bootstrap, выполните общий playbook deploy.yml. Команда выполняется из рабочей директории установщика.

```
ansible-playbook playbooks/deploy.yml \
  -u <ssh-user> \
  --vault-password-file .ansible/vault-password
```

Параметр -u задает SSH-пользователя, от имени которого Ansible подключается к целевым хостам. Если для sudo требуется пароль, используйте дополнительный параметр --ask-become-pass.

```
ansible-playbook playbooks/deploy.yml \
  -u <ssh-user> \
  --ask-become-pass \
  --vault-password-file .ansible/vault-password
```

6.2 Установка отдельных компонентов

Если необходимо установить только один компонент или выполнить установку компонентов поэтапно, используйте соответствующий playbook. Ниже приведены типовые команды.

```
ansible-playbook playbooks/deploy-clickhouse.yml -u <ssh-user> --vault-password-
file .ansible/vault-password
ansible-playbook playbooks/deploy-kafka.yml -u <ssh-user> --vault-password-file
.ansible/vault-password
ansible-playbook playbooks/deploy-victoria-metrics.yml -u <ssh-user>
ansible-playbook playbooks/deploy-vmagent.yml -u <ssh-user>
```

```
ansible-playbook playbooks/deploy-vector.yml -u <ssh-user> --vault-password-file
.ansible/vault-password
ansible-playbook playbooks/deploy-postgres-ha.yml -u <ssh-user> --vault-password-
file .ansible/vault-password
ansible-playbook playbooks/deploy-lk.yml -u <ssh-user> --vault-password-file .ansible/vault-
password
```

Компоненты, использующие vault-переменные, запускаются с параметром --vault-password-file. Для компонентов, не использующих зашифрованные переменные, этот параметр может не требоваться.

Runtime-компоненты Cybert (Cresp, Cresp-BH, NGINX/Angie с модулем Cybert), Cresp API, Cresp config deliver и MariaDB устанавливаются playbook-ами или ролями, входящими в конкретную версию installer-a. Точные имена component playbook-ов необходимо брать из release manifest и каталога playbooks поставки; общий playbook deploy.yml устанавливает весь выбранный состав.

6.3 Повторный запуск установки и обработка ошибок

Ansible playbook-и допускают повторный запуск после устранения ошибки. Если команда завершилась ошибкой на одном из хостов, следующие этапы на этом хосте в рамках того же запуска могут быть пропущены. Это является стандартным поведением Ansible.

В таком случае необходимо определить причину ошибки по выводу playbook-a, исправить ее и повторно запустить ту же команду. Перед повторным запуском рекомендуется проверить доступность хоста, права sudo, доступ к репозиториям и корректность vault password.

7 Проверка результата установки

7.1 Проверка системных сервисов

После завершения установки необходимо проверить состояние системных сервисов на целевых хостах. Набор команд зависит от выбранного состава компонентов.

```
systemctl status clickhouse-server
systemctl status kafka
systemctl status victoria-metrics
systemctl status vmagent
systemctl status vector-cresp-custom@1
systemctl status vector-cresp-log@1
systemctl status patroni
systemctl status mariadb    # если MariaDB установлена как unit mariadb
systemctl list-units --type=service | grep -E 'cresp|cybert|nginx|angie'
nginx -t    # если используется NGINX
angie -t    # если используется Angie
docker ps
```

Сервисы должны находиться в активном состоянии, а контейнеры личного кабинета и API управления должны быть запущены. Если отдельный сервис не запустился, необходимо изучить журнал systemd, журналы компонента и вывод Ansible playbook-a.

7.2 Проверка PostgreSQL HA и Patroni

Для проверки состояния Patroni используйте HTTP-endpoint одного из mgmt-узлов:

```
curl http://10.250.10.11:8008/cluster
```

В ответе должен отображаться состав кластера и состояние его участников. Для production-стенда необходимо дополнительно убедиться, что определен лидер, реплики доступны, а состояние кластера соответствует выбранной топологии.

7.3 Проверка VictoriaMetrics

Для single-node режима VictoriaMetrics выполните проверку health endpoint:

```
curl http://10.250.10.11:8428/health
```

Для cluster-режима VictoriaMetrics проверьте health endpoint-ы основных компонентов:

```
curl http://10.250.10.11:8480/health
curl http://10.250.10.11:8481/health
curl http://10.250.10.11:8482/health
```

7.4 Проверка runtime-контура и контейнерного контура личного кабинета

На filter nodes необходимо убедиться, что фронтенд-сервер NGINX/Angie запускается с модулем Cybert, Cresp и Cresp-BH находятся в рабочем состоянии, а сервисные журналы не содержат ошибок загрузки конфигурации или RPC-взаимодействия. Фактические systemd unit-имена определяются release manifest. Для контейнерного контура панели управления выполните команду docker ps на узлах, где установлен LK docker stack.

```
docker ps
```

Дополнительно проверьте доступность web-интерфейса и API управления, состояние Cresp API и Cresp config deliver, а также успешную доставку актуальной runtime-конфигурации на filter nodes. Для контроля результата рекомендуется выполнить тестовый HTTP(S)-запрос к защищаемому ресурсу и убедиться, что запрос обрабатывается Cybert, а сведения о нем появляются в журнале и метриках.

Дополнительно рекомендуется проверить доступность web-интерфейса и API управления с рабочей станции администратора или из разрешенного административного сегмента сети.

7.5 Критерии успешного завершения установки

Установка считается успешно завершенной при выполнении следующих условий:

- общий playbook или playbook-и выбранных компонентов завершились без неустраненных ошибок;
- все установленные системные сервисы находятся в активном состоянии;
- контейнеры LK docker stack запущены;
- PostgreSQL HA/Patroni отображает корректное состояние кластера;
- VictoriaMetrics health endpoint-ы отвечают успешно;
- агенты vmagent и Vector запущены на предусмотренных узлах;
- NGINX/Angie на filter nodes запускается с модулем Cybert, а Cresp и Cresp-BH находятся в рабочем состоянии;
- Cresp API имеет доступ к MariaDB, а сервис Cresp config deliver доставляет сформированную runtime-конфигурацию на filter nodes;

- тестовый HTTP(S)-запрос к защищаемому ресурсу проходит через runtime-контур Cybert и получает ожидаемый вердикт;
- административный интерфейс и API управления доступны из разрешенной сети;
- оператор сохранил используемую версию архива, inventory, .installer.env, журнал установки и сведения о выполненных проверках.

8 Обновление программного обеспечения

8.1 Общий порядок обновления

Обновление Servicepipe Cybert выполняется с использованием нового online installer archive. В новом комплекте поставляется файл .installer.env и release manifest, в которых зафиксированы URL репозиториев, новые имена архивов/container images и совместимый состав компонентов. При обновлении необходимо сохранять согласованность версий Cresp, модуля Cybert для NGINX/Angie, Cresp API, формата runtime-конфигурации и сервисов доставки конфигурации.

Общий порядок обновления:

- распаковать новый online archive во временную директорию;
- заменить .installer.env в рабочей директории установленного инсталлятора;
- обновить рабочие переменные установщика без переопределения топологии;
- запустить update-команды для всех компонентов или только для выбранного компонента;
- проверить состояние сервисов после обновления.

8.2 Подготовка нового online-архива

Для подготовки .installer.env из новой поставки выполните следующие команды:

```
mkdir -p /tmp/cybert-onprem-update
tar -xzf cybert-onprem-ansible-online-<new-version>.tar.gz -C /tmp/cybert-onprem-update

cp /tmp/cybert-onprem-update/cybert-onprem-ansible/.installer.env .installer.env
```

После замены .installer.env обновите рабочие переменные установщика:

```
./bootstrap.sh --refresh-installer
```

8.3 Обновление всех компонентов

Для обновления инфраструктурных компонентов выполните команды по порядку. Если какой-либо компонент не используется на стенде, соответствующую строку необходимо пропустить. Runtime-компоненты Cybert и MariaDB обновляются playbook-ами/ролями, указанными в release manifest конкретной версии.

```
ansible-playbook playbooks/update-local-repo.yml \
  -u <ssh-user>

ansible-playbook playbooks/update-clickhouse.yml \
  -u <ssh-user> \
  --vault-password-file .ansible/vault-password

ansible-playbook playbooks/update-kafka.yml \
  -u <ssh-user> \
  --vault-password-file .ansible/vault-password

ansible-playbook playbooks/update-victoria-metrics.yml \
  -u <ssh-user>
```

```
ansible-playbook playbooks/update-vmagent.yml \
  -u <ssh-user>

ansible-playbook playbooks/update-vector.yml \
  -u <ssh-user> \
  --vault-password-file .ansible/vault-password

ansible-playbook playbooks/update-postgres-ha.yml \
  -u <ssh-user> \
  --vault-password-file .ansible/vault-password

ansible-playbook playbooks/update-lk.yml \
  -u <ssh-user> \
  --vault-password-file .ansible/vault-password
```

8.4 Обновление одного компонента

Для обновления одного компонента выполните только соответствующую update-команду из перечня выше. Перед обновлением отдельного компонента необходимо убедиться, что новая версия совместима с остальными компонентами установленного стенда.

8.5 Обновление только container images личного кабинета

Если необходимо обновить только container images LK без изменения остальных настроек личного кабинета, выполните команду:

```
ansible-playbook playbooks/update-lk-images.yml \
  -u <ssh-user> \
  --vault-password-file .ansible/vault-password
```

После обновления container images необходимо проверить список контейнеров, доступность web-интерфейса и API управления.

9 Удаление программного обеспечения

9.1 Общие правила удаления

Удаление компонентов выполняется отдельными playbook-ами. Перед удалением runtime-компонентов Cybert необходимо вывести защищаемые ресурсы из эксплуатации или перевести трафик на согласованный обходной маршрут, чтобы не нарушить доступность приложений. По умолчанию component remove playbook удаляет service/unit/config/runtime-файлы компонента, но не удаляет рабочие данные и RPM-пакеты, если иное не указано в поставке.

> **Примечание.** Перед удалением компонентов с данными необходимо убедиться, что выполнено резервное копирование требуемых конфигураций, журналов, баз данных и иных материалов, которые должны быть сохранены.

9.2 Удаление компонентов

Для удаления компонентов используйте соответствующие playbook-и:

Для Cresp, Cresp-BH, NGINX/Angie с модулем Cybert, Cresp API, Cresp config deliver и MariaDB используйте component remove playbook-и, включенные в конкретный installer и перечисленные в release manifest. Ниже приведены типовые команды удаления общих инфраструктурных компонентов.

```
ansible-playbook playbooks/remove-lk.yml -u <ssh-user>
ansible-playbook playbooks/remove-vector.yml -u <ssh-user>
```

```
ansible-playbook playbooks/remove-vmagent.yml -u <ssh-user>
ansible-playbook playbooks/remove-postgres-ha.yml -u <ssh-user>
ansible-playbook playbooks/remove-kafka.yml -u <ssh-user>
ansible-playbook playbooks/remove-clickhouse.yml -u <ssh-user>
ansible-playbook playbooks/remove-victoria-metrics.yml -u <ssh-user>
ansible-playbook playbooks/remove-local-repo.yml -u <ssh-user>
```

9.3 Удаление с данными и пакетами

Для удаления компонента вместе с рабочими данными используйте переменную component_purge_data=true:

```
ansible-playbook playbooks/remove-kafka.yml \
  -u <ssh-user> \
  -e component_purge_data=true
```

Для удаления компонента вместе с RPM-пакетами используйте переменную component_remove_packages=true:

```
ansible-playbook playbooks/remove-kafka.yml \
  -u <ssh-user> \
  -e component_remove_packages=true
```

Для полного удаления компонента вместе с рабочими данными и пакетами используйте обе переменные:

```
ansible-playbook playbooks/remove-kafka.yml \
  -u <ssh-user> \
  -e component_purge_data=true \
  -e component_remove_packages=true
```

9.4 Рекомендуемый порядок удаления

При полном удалении стенда рекомендуется удалять компоненты в следующем порядке:

- LK / frontend и backend панели управления;
- Cresp API и Cresp config deliver;
- runtime-компоненты на filter nodes: NGINX/Angie с модулем Cybert, Cresp и Cresp-BH - после прекращения направления рабочего трафика;
- Vector и vmagent;
- PostgreSQL HA и MariaDB - после резервного копирования требуемых данных;
- Kafka;
- ClickHouse;
- VictoriaMetrics;
- local-repo client configuration.

Такой порядок позволяет сначала остановить пользовательский и управляющий контур, затем безопасно вывести из эксплуатации runtime-слой, после этого удалить агенты доставки данных, базы данных, брокеры событий, аналитические хранилища и настройки репозиториев. Конкретный порядок должен учитывать зависимости версии, указанные в release manifest.

10 Типовые проблемы и способы устранения

10.1 Для sudo требуется пароль

Если при выполнении playbook-а появляется сообщение sudo: a password is required, значит SSH-пользователь не может выполнить sudo без пароля. Запустите команду с параметром --ask-become-pass или настройте sudo на целевых хостах.

```
ansible-playbook playbooks/deploy.yml \
  -u <ssh-user> \
  --ask-become-pass \
  --vault-password-file .ansible/vault-password
```

10.2 Хост завершился с ошибкой, следующие этапы пропущены

Если хост получил failed=1, следующие этапы на этом хосте в рамках того же запуска могут быть пропущены. Это стандартное поведение Ansible. Необходимо исправить причину ошибки и перезапустить команду.

10.3 DNF не находит пакет

Если DNF не находит требуемый пакет, проверьте доступность online RPM repository с целевой ноды и наличие пакета в репозитории.

```
dnf clean all
dnf makecache
dnf repoquery <package-name>
```

10.4 Не скачивается архив из raw repository

Если не скачивается archive или иной файл из raw repository, проверьте URL raw repository, указанный при подготовке установщика, и доступность требуемого файла.

```
curl -I <raw-repository-url>/<file-name>
```

10.5 Неверный vault password

Если Ansible сообщает об ошибке расшифровки vault-файлов, убедитесь, что используется тот же vault password, который вводился во время bootstrap. Проверку можно выполнить командой:

```
ansible-vault view inventory/group_vars/lk_nodes_vault.yml \
  --vault-password-file .ansible/vault-password
```

Если файл .ansible/vault-password был заменен или поврежден, необходимо восстановить корректный файл пароля из защищенного хранилища либо повторно сформировать vault-файлы в установленном порядке.

Если после установки или обновления профильная конфигурация не применяется на filter nodes, проверьте состояние Cresp API и MariaDB, журналы Cresp config deliver, сетевую доступность между mgmt и filter nodes и журналы Cresp. Если NGINX/Angie не может получить вердикт от Cresp, проверьте загрузку модуля Cybert, соответствие версий runtime-компонентов и доступность используемого RPC-взаимодействия.

11 Завершение работ и хранение установочных материалов

После завершения установки, обновления или удаления необходимо сохранить сведения, позволяющие подтвердить состав выполненных работ и воспроизвести конфигурацию стенда.

Рекомендуется сохранить:

- версию online installer archive и его контрольную сумму;
- использованный release manifest и файл .installer.env;
- inventory и group vars, сформированные bootstrap;
- сведения о выбранной топологии и ролях узлов;
- перечень запущенных playbook-ов и их результат;
- журнал выполнения установки или обновления;
- результаты проверок systemctl, docker ps, health endpoint-ов и доступности административного интерфейса;
- результаты проверки runtime-контура: состояние NGINX/Angie с модулем Cybert, Cresp/Cresp-BH, Cresp API, MariaDB и доставки runtime-конфигурации;
- сведения об ответственных администраторах и дате выполнения работ.

Файл .ansible/vault-password и иные секреты должны храниться отдельно от общедоступной документации и журналов работ. Доступ к ним должен предоставляться только уполномоченным сотрудникам, ответственным за сопровождение установленного экземпляра Servicepipe Cybert.