

???????? SMP

Установка Single Management Platform / XDR

- [Архитектура и термины](#)
- [Установка SMP](#)
- [Подготовка серверов](#)
- [Настройка СУБД](#)
- [Заполнение файла параметров и инвентаря](#)
- [Полезные команды](#)

???????????? ? ??????????

Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

Официальная документация по данному разделу приведена в Онлайн-справке на продукт: <https://support.kaspersky.com/help/XDR/1.4/ru-RU/298639.htm>

???? ??????

KDT/устройство оператора - устройство для развертывания и управления компонентами OSMP.

Primary/master/controller/первичный рабочий узел - узел контроллера, осуществляющий управление кластером k0s.

Worker/рабочий узел - узел кластера k0s с полезной нагрузкой.

DB/СУБД - сервер с СУБД для кластера OSMP.

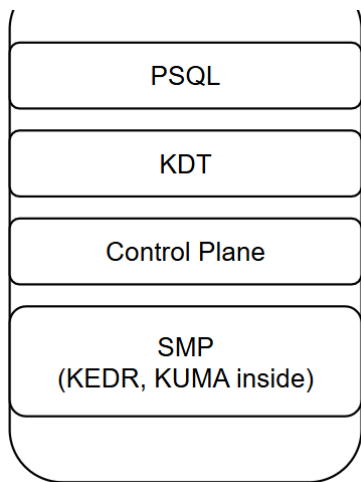
KUMA services/устройство с сервисами KUMA - устройства с установленными сервисами KUMA: коллектор, коррелятор, хранилище.

Целевые устройства - устройства, на которых устанавливается OSMP (все вышеперечисленные узлы)

????????????????????

KEDR

Данная часть не относится к XDR, но относится к SMP, а именно к варианту развертывания KEDR 8.0 в однонодовой/single node конфигурации



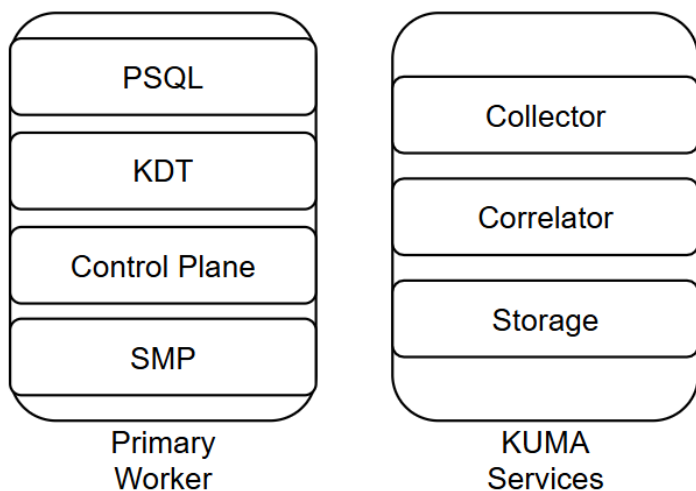
Здесь отличие от схемы ниже заключается в том, что сервисы

KUMA ставятся не на отдельную машину, а помещаются внутрь кластера kubernetes вместе с остальными компонентами SMP, в т.ч. KEDR.

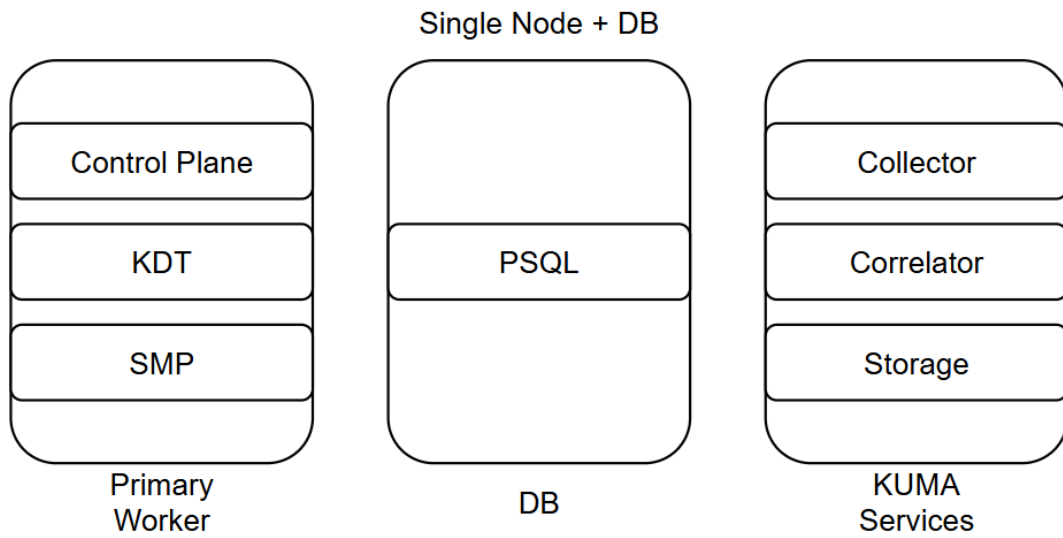
Подробнее про установку KEDR - [статья](#)

Single node/однонодовая конфигурация - вариант развертывания, при котором 1 узел совмещает в себе роли KDT, Primary, Worker и DB, а 2 узел (или несколько узлов) отведен под компоненты KUMA.

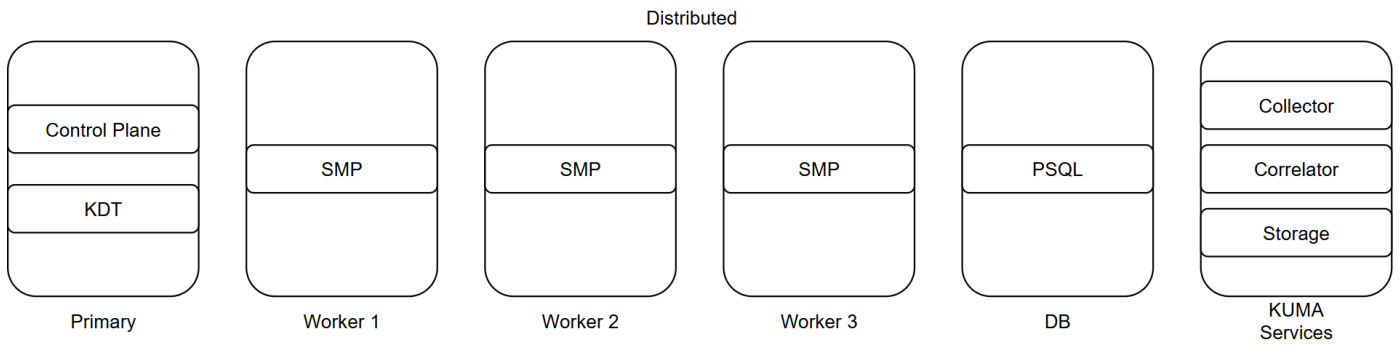
Single Node



Single node + DB - вариант развертывания, при котором 1 узле совмещает в себе роли KDT, Primary, Worker, 2 узел отводится под СУБД и 3 (один или более) узел отводится под компоненты KUMA.



Multi node/Distributed/многонодовая конфигурация - вариант развертывания, при котором каждому узлу отводится своя роль. При этом число узлов с ролью Worker должно быть не менее 3, а роль KDT и Primary можно совместить на одном узле.



???????? SMP

Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

Официальная документация по данному разделу приведена в Онлайн-справке на продукт: <https://support.kaspersky.com/help/XDR/2.0/ru-RU/273808.htm>

В данном разделе приведена инструкция по разворачиванию SMP. Перед прочтением, рекомендуем ознакомиться с основными терминами и архитектурой разворачивания по [ссылке](#).

1. Подготовьте целевые устройства согласно [инструкции](#)
2. Загрузите на хост, с которого будет производиться установка, следующие файлы:
 - `bin.tar.gz` - архив с утилитой kdt, используется для разворачивания, изменения параметров инсталляции
 - `xdr-<version>.tar.gz` - транспортный архив с компонентами SMP (обратите внимание, что данный архив распаковывать не нужно)
 - `license.key` - ключ лицензии для KUMA

Обратите внимание! Для установки нужно указывать именно файл лицензии для **KUMA**. Файл лицензии для функционала **XDR** подгружается после установки в веб-интерфейсе!

3. Распакуйте архив `bin.tar.gz` в директорию на хосте (рекомендуется использовать домашний каталог пользователя, под которым будет выполняться установка). Для удобства перейдите в эту директорию.

4. Дайте файлу kdt права на выполнение. Пример:

```
chmod +x kdt
```

5. Переместите в ту же директорию транспортный архив `xdr-<version>.tar.gz`

6. В зависимости от планируемой конфигурации KUMA скопируйте файл `single.inventory.yml.template` (all in one) или `distributed.inventory.yml.template` (distributed).

Пример:

```
cp single.inventory.yml.template inventory.yml
```

7. Заполните получившийся файл параметров (`inventory.yml`) в соответствии с [инструкцией](#).

8. В зависимости от планируемой конфигурации SMP скопируйте файл `singlenode.smp_param.yml.template` (для однонодовой установки) или `multinode.smp_param.yml.template` (для многонодовой установки). Пример:

```
cp singlenode.smp_param.yml.template param.yml
```

9. Заполните получившийся файл параметров (`param.yml`) в соответствии с [инструкцией](#).

10. Запустите установку командой:

```
./kdt apply -k <полный путь к транспортному архиву> -i <полный путь к файлу конфигурации param.yml> --accept-eula
```

Установка в среднем занимает от 30 до 60 минут. Удачи (:

???????????? ???? ?????

Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

Официальная документация по данному разделу приведена в Онлайн-справке на продукт: <https://support.kaspersky.com/help/XDR/2.0/ru-RU/249228.htm>

???????????? ? ????????? XDR

Primary, worker, KDT, DB

1. Наборы инструкций BMI, AVX, SSE 4.2
2. SWAP должен быть отключен на всех серверах
3. На всех серверах должен быть доступ в Интернет или к внутренним зеркалам репозиторий ОС
4. На всех серверах должна быть настроена синхронизация времени по NTP
5. Поддерживаемые ОС:
 - Ubuntu Server 22.04 LTS
 - Ubuntu Server 24.04 LTS
 - Debian GNU/Linux 12.x (Bookworm)
 - Astra Linux Special Edition РУСБ.10015-01 (очередное обновление 1.8.1)
6. Версия ядра: 5.15.0.107 или выше
7. Поддерживаемые СУБД (для сервера DB):
PostgreSQL 15.x - 17.x 64-разрядные и PostgreSQL Pro 15.x - 17.x 64-разрядные
8. Для Single-node установки на целевом устройстве должно быть доступно **360 Гб** свободного в директории `/var`.
9. Для Multi-node установки на всех целевых устройствах (роли Primary, Worker) должно быть доступно **700 Гб** свободного места в директории `/var`.
Для worker-ов необходимо /var не менее 150 Гб, /opt не менее 1 ТБ

?????????? ? ?????????? ????????

В случае single-node инсталляции выполните все указанные ниже действия на целевом устройстве в той же последовательности, в которой они приведены

KDT

1. Установите python 3.8 или выше:

```
apt install python3
```

2. Установите docker версии 23 или выше из официального репозитория docker. Инструкцию для выбранной ОС можно найти на сайте: <https://docs.docker.com/engine/install/>

DB

1. Установите PostgreSQL поддерживаемой версии из репозитория ОС или репозитория Postgres:

```
apt install postgresql
```

2. Выполните настройку СУБД согласно [статье](#)

Primary

1. Установите пакеты sudo, curl, ufw, python3-apt из репозитория ОС:

```
apt install sudo curl ufw python3-apt
```

2. В конфигурационном файле UFW (`/etc/default/ufw`) установите для параметра `DEFAULT_FORWARD_POLICY` значение `ACCEPT`

3. Откройте требуемые порты согласно [инструкции](#), либо отключите UFW (рекомендуется только для демонстрационной установки!)

Отключение UFW

```
systemctl stop ufw  
systemctl disable ufw
```

Worker

1. Установите пакеты `sudo`, `ufw`, `nfs-common`, `tar`, `iscsi-package`, `wireguard`, `wireguard-tools`, `python3-apt`, `libnfs` из репозитория ОС:

```
apt install sudo ufw nfs-common tar iscsi-package wireguard wireguard-tools python3-apt curl  
libnfs cron
```

2. Откройте требуемые порты согласно [инструкции](#), либо отключите UFW (рекомендуется только для демонстрационной установки!)

Отключение UFW

```
systemctl stop ufw  
systemctl disable ufw
```

KUMA

[СМ. СТАТЬЮ](#)

???????????? ???? ?????????????

1. Зарезервируйте IP-адрес из той же подсети, что и у серверов Primary/Worker. Назначать данный адрес никуда не нужно, он должен быть свободен и будет назначен в процессе установки (указывается в файле `param.yaml` в поле `ingress_ip`).

2. Добавьте в DNS-зону вашей организации (предпочтительно создать поддомен) следующие записи:

```
console.<smp_domain> <ingress_ip>  
admsrv.<smp_domain> <ingress_ip>  
api.<smp_domain> <ingress_ip>  
monitoring.<smp_domain> <ingress_ip>  
updater.<smp_domain> <ingress_ip>  
agentserver.<smp_domain> <ingress_ip>  
kuma.<smp_domain> <ingress_ip>  
*.kuma.<smp_domain> <ingress_ip>
```

где, `<smp_domain>` - домен или поддомен вашей организации (данный домен также должен быть указан в файле параметров в одноименно поле `smp_domain`), а `<ingress_ip>` -

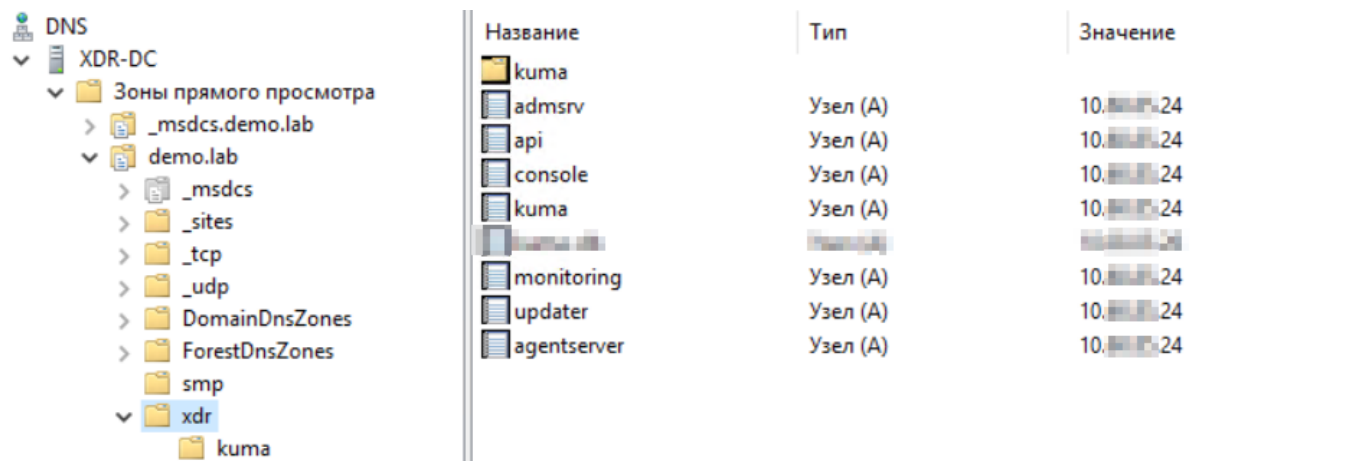
зарезервированный IP из предыдущего пункта.

Также обратите внимание, что для имени **kuma** также необходимо создать wildcard (*) на DNS-сервере

Имена `console`, `admsrv`, `kuma`, `api`, `monitoring`, `updater`, `agentserver` являются именами по умолчанию. В случае, если вы хотите использовать другие имена - следуйте инструкции.

DNS записи на примере Windows DNS

В примере ниже в качестве `ingress_ip` выступает адрес `10.0.0.24`, а в качестве `smp_domain` - `xdr.demo.lab`

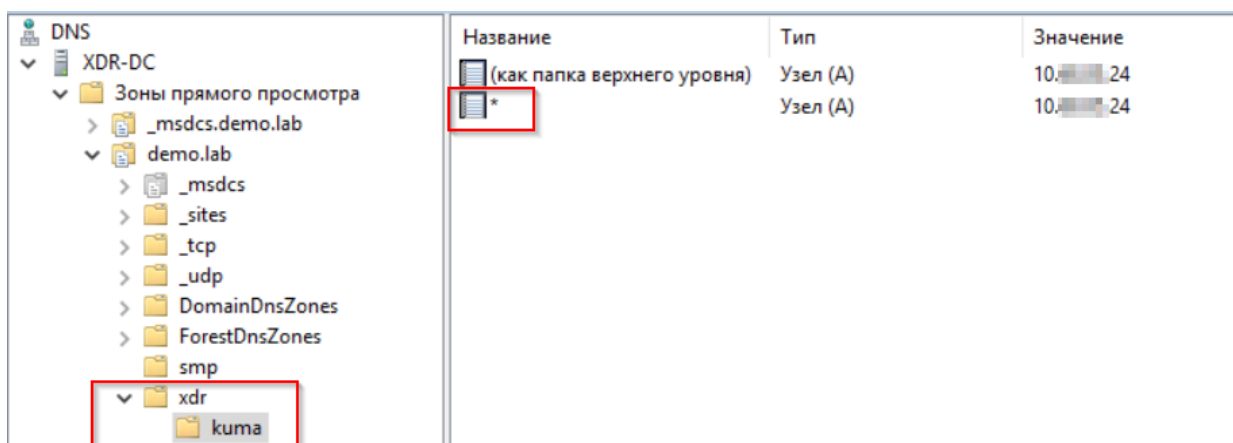


The screenshot shows the Windows DNS console with the following structure:

- DNS > XDR-DC > Зоны прямого просмотра > _msdcs.demo.lab > demo.lab > xdr > kuma

Название	Тип	Значение
kuma		
admsrv	Узел (A)	10.0.0.24
api	Узел (A)	10.0.0.24
console	Узел (A)	10.0.0.24
kuma	Узел (A)	10.0.0.24
monitoring	Узел (A)	10.0.0.24
updater	Узел (A)	10.0.0.24
agentserver	Узел (A)	10.0.0.24

Пример wildcard

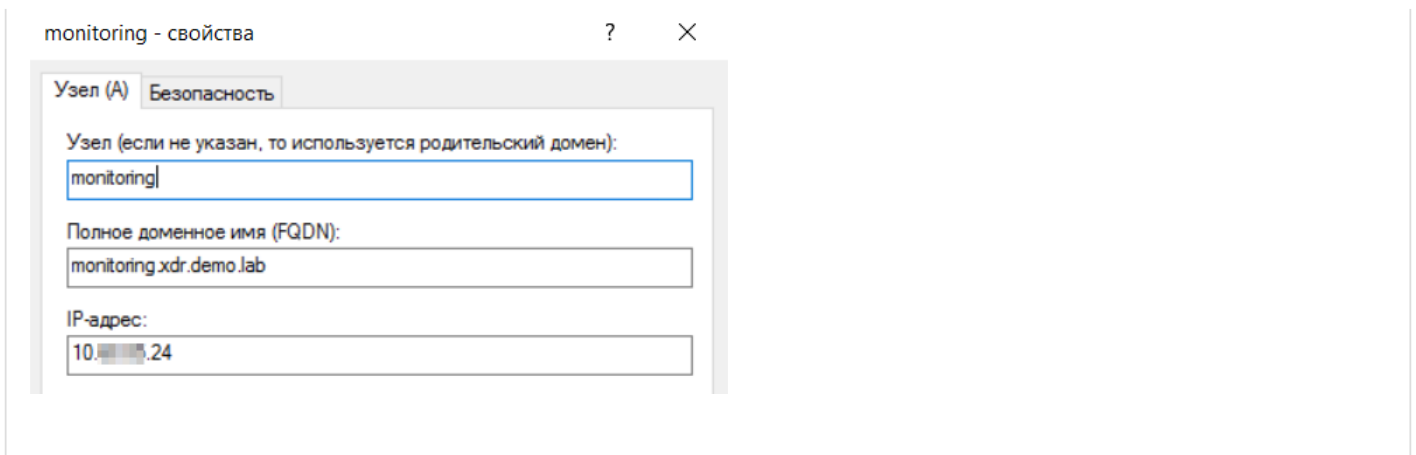


The screenshot shows the Windows DNS console with the following structure:

- DNS > XDR-DC > Зоны прямого просмотра > _msdcs.demo.lab > demo.lab > xdr > kuma

Название	Тип	Значение
(как папка верхнего уровня)	Узел (A)	10.0.0.24
*	Узел (A)	10.0.0.24

Пример отдельной записи



3. На хосте KDT (в случае single-node установке на целевом устройстве) создайте и распространите на все остальные устройства ssh-ключ пользователя, от имени которого будет производиться установка (либо от пользователя root, если установка будет происходить из-под root). Примеры добавления ssh-ключей: [ссылка](#)

???????????? ?????

Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

Официальная документация по данному разделу приведена в Онлайн-справке на продукт: <https://support.kaspersky.com/help/XDR/2.0/ru-RU/241223.htm>

????????? ?????? ?????????????????? ????? ? ?????????? ?? ?????????? ??
? ??????????????????

1. Подключитесь к СУБД

```
sudo -i -u postgres  
psql
```

2. Выполните команды

```
CREATE USER user WITH PASSWORD 'password';  
ALTER ROLE user CREATEDB;  
ALTER ROLE user CREATEROLE;  
ALTER ROLE user SUPERUSER;  
CREATE DATABASE user OWNER user;  
\q
```

где, `user` - имя нового пользователя

Альтернативно, можно использовать встроенного пользователя `postgres`, предварительно сменив ему пароль:

```
ALTER USER postgres WITH PASSWORD 'password';
```

????????????????????? ?????????????????? ??????

1. Откройте на редактирование файл конфигурации PostgreSQL:

```
vi /etc/postgresql/<БЕПСИЯ>/main/postgresql.conf
```

2. Найдите приведенные ниже параметры в файле конфигурации и переопределите их следующим образом:

```
port = 5432
listen_addresses = '*'
shared_buffers = <X>GB
max_stack_depth = <Y>MB
temp_buffers = 24MB
work_mem = 16MB
max_connections = 512
max_parallel_workers_per_gather = 0
maintenance_work_mem = 128MB
standard_conforming_strings = on
```

где, <X> - 25%-40% от общего числа ОЗУ на сервере

Как посмотреть объем оперативной памяти

```
free --giga
```

<Y> - размер стека минус 1MB

Как посмотреть размер стека

```
ulimit -s
```

Команда выведет размер стека в KB. Для приведения к MB необходимо полученное число разделить на 1024.

3. Разрешите удаленное подключение к СУБД через правку конфигурационного файла

```
vi /etc/postgresql/<ВЕРСИЯ>/main/pg_hba.conf
```

В секции `# IPv4 local connections:` переопределите значение или убедитесь, что значение соответствует следующему:

```
host all all 0.0.0.0/0 scram-sha-256
```

4. Перезапустите службу

```
systemctl restart postgresql
```

????????? ?????????????? ? ?????

1. Установите на другой хост клиент PostgreSQL:

```
sudo apt install postgresql-client
```

2. Проверьте подключение командой:

```
psql -U <user> -h <host> -p 5432  
\dx  
\q
```

где, `<user>` - имя пользователя, созданного для БД, а `<host>` - адрес хоста, на котором располагается СУБД.

???????????? ???? ???? ??????
? ??????????

Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

Официальная документация по данному разделу приведена в Онлайн-справке на продукт:

<https://support.kaspersky.com/help/XDR/2.0/ru-RU/271992.htm>

<https://support.kaspersky.com/help/XDR/2.0/ru-RU/249240.htm>

?????????

В SMP для установки необходимо заполнить 2 файла с конфигурацией: `param.yaml` и `inventory.yaml`. Первый файл отвечает за настройку и развертывание SMP, а второй - за развертывание KUMA.

Ниже приведем примеры конфигурационных файлов для установки, а также наиболее популярные сценарии переопределения стандартных параметров установки.

???????? ???? ???? ??????

Single node

Ниже приведен пример файла `param.yaml` для однонодовой инсталляции SMP с комментариями и советами по заполнению.

```
schemaType: ParameterSet
schemaVersion: 1.0.1
namespace: ""
name: bootstrap
project: xdr

nodes:
  - desc: cdt-1
```

type: primary-worker

host: 10.80.23.42 # Необходимо указать IP адрес целевого устройства

access:

ssh:

user: root # Пользователь, от имени которого будет производиться установка

key: /root/.ssh/id_rsa # Путь к ssh-ключу пользователя

parameters:

- name: psqldb

source:

Необходимо заполнить строку подключения к БД: user_db - имя пользователя, password_db - пароль,

(если пароль содержит спецсимволы их необходимо привести к URI кодировке)

host_db - адрес узла с БД (для Single node это адрес указанный в параметре выше, port_db - порт БД

(по умолчанию 5432)

value: "postgres://<user_db>:<password_db>@<host_db>:<port_db>"

- name: ingress_ip

source:

value: 10.80.23.182 # Резервированный IP адрес

- name: ssh_pk

source:

path: /root/.ssh/id_rsa # Значение должно совпадать с указанным выше значением key

- name: admin_password

source:

value: "AdminKit2015!" # Пароль для доступа к интерфейсу, от 8 до 256 символов
должен содержать буквы, цифры и спец. символы

- name: inventory

source:

value: "/home/user/inventory.yml" # Путь к файлу инвентаря для установки компонентов

KUMA, для установки только KEDR

укажите /dev/null

- name: license

source:

value: "/home/user/license.key" # Путь к файлу ключа лицензии KUMA

- name: smp_domain

source:

value: "smp.local" # Укажите домен для компонентов SMP

- name: pki_host_list

source:

```
    value: "admsrv api console kuma monitoring updater agentserver"

# По умолчанию размер тома для ядра - 4 Гб при low_resources = true
# Если хотите изменить это значение раскомментируйте блок параметров ниже
# И укажите требуемое значение

# - name: core_request_low_resources
#   source:
#     value: "100Gi"

# Начало блока, который не нужно изменять
- name: low_resources
  source:
    value: "true"
- name: default_class_replica_count
  source:
    value: "1"
- name: openbao_ha_mode
  source:
    value: "false"
- name: openbao_standalone
  source:
    value: "true"
```

Multi node

Ниже приведен пример файла `param.yaml` для многонодовой инсталляции SMP с комментариями и советами по заполнению.

```
schemaType: ParameterSet
schemaVersion: 1.0.1
namespace: ""
name: bootstrap
project: xdr

nodes:
  # одна нода контроллера
  - desc: cdt-primary1
    type: primary
    host: 10.80.23.42 # IP-адрес ноды контроллера. Данный адрес,
```

а также все адрес приведенные ниже, должны быть из одной подсети

access:

ssh:

user: root # Пользователь, от имени которого будет производиться установка

key: /root/.ssh/id_rsa # Путь к ssh-ключу пользователя

три рабочие ноды

- desc: cdt-w1

type: worker

host: 10.80.23.141 # IP-адрес 1-ой рабочей ноды

access:

ssh:

user: root # Пользователь, от имени которого будет производиться установка

key: /root/.ssh/id_rsa # Путь к ssh-ключу пользователя

- desc: cdt-w2

type: worker

host: 10.80.23.45 # IP-адрес 2-ой рабочей ноды

access:

ssh:

user: root # Пользователь, от имени которого будет производиться установка

key: /root/.ssh/id_rsa # Путь к ssh-ключу пользователя

- desc: cdt-w3

type: worker

host: 10.80.23.30 # IP-адрес 3-ей рабочей ноды

access:

ssh:

user: root # Пользователь, от имени которого будет производиться установка

key: /root/.ssh/id_rsa # Путь к ssh-ключу пользователя

kind: admsrv # Данный параметр необходимо указать для одной любой рабочей ноды

parameters:

Необходимо заполнить строку подключения к БД: user_db - имя пользователя, password_db - пароль,

(если пароль содержит спецсимволы их необходимо привести к URI кодировке)

host_db - адрес узла с БД (для Single node это адрес указанный в параметре выше, port_db - порт БД

(по умолчанию 5432)

- name: psql_dsn

source:

value: "postgres://<user_db>:<password_db>@<host_db>:<port_db>"

- name: ingress_ip

```

source:
  value: 10.80.23.182 # Зарезервированный IP адрес
- name: ssh_pk
source:
  path: /root/.ssh/id_rsa # Значение должно совпадать с указанным выше значением key
- name: admin_password
source:
  value: "AdminKit2015!" # Пароль для доступа к интерфейсу, от 8 до 256 символов
                        # должен содержать буквы, цифры и спец. символы
- name: core_disk_request
source:
  value: 512Gi # Укажите размер тома ядра
- name: inventory
source:
  value: "/home/user/inventory.yml" # Путь к файлу инвентаря для установки компонентов
KUMA
- name: license
source:
  value: "/home/user/license.key" # Путь к файлу ключа лицензии KUMA
- name: smp_domain
source:
  value: "smp.local" # Укажите домен для компонентов SMP
- name: pki_host_list
source:
  value: "admsrv api console kuma monitoring updater agentserver"

```

???????? ???? ???? ????????

All in one

Ниже приведен пример файла `inventory.yml` для **all in one** инсталляции KUMA в рамках SMP с комментариями и советами по заполнению.

```

all:
  vars:
    deploy_example_services: false
    ansible_connection: local # не изменять
    ansible_user: nonroot # не изменять
kuma:

```

```
vars:
  ansible_connection: ssh
  ansible_user: root # укажите имя пользователя, от имени которого будет производиться
установка
# В случае, если пользователь не root, раскомментируйте строки ниже
#  ansible_become: true
#  ansible_become_method: sudo
children:
  kuma_utils:
    hosts:
      kuma.example.com: # укажите FQDN сервера для компонентов KUMA
        ansible_host: 0.0.0.0 # укажите IP сервера для компонентов KUMA
  kuma_collector:
    hosts:
      kuma.example.com: # укажите FQDN сервера для компонентов KUMA
        ansible_host: 0.0.0.0 # укажите IP сервера для компонентов KUMA
  kuma_correlator:
    hosts:
      kuma.example.com: # укажите FQDN сервера для компонентов KUMA
        ansible_host: 0.0.0.0 # укажите IP сервера для компонентов KUMA
  kuma_storage:
    hosts:
      kuma.example.com: # укажите FQDN сервера для компонентов KUMA
        ansible_host: 0.0.0.0 # укажите IP сервера для компонентов KUMA
        shard: 1
        replica: 1
        keeper: 1
```

Distributed

Ниже приведен пример файла `inventory.yaml` для **distributed** инсталляции KUMA в рамках SMP с комментариями и советами по заполнению.

```
all:
  vars:
    deploy_example_services: false
    ansible_connection: local # не изменять
    ansible_user: nonroot # не изменять
  kuma:
    vars:
```

```
ansible_connection: ssh
ansible_user: root # укажите имя пользователя, от имени которого будет производиться
установка
# В случае, если пользователь не root, раскомментируйте строки ниже
#   ansible_become: true
#   ansible_become_method: sudo
children:
  kuma_utils:
    hosts:
      kuma-utils.example.com: # укажите FQDN сервера для компонентов KUMA
      ansible_host: 0.0.0.0 # укажите IP сервера для компонентов KUMA
  kuma_collector:
    hosts:
      kuma-collector.example.com: # укажите FQDN сервера для коллектора KUMA
      ansible_host: 0.0.0.0 # укажите IP сервера для коллектора KUMA
  kuma_correlator:
    hosts:
      kuma-correlator.example.com: # укажите FQDN сервера для коррелятора KUMA
      ansible_host: 0.0.0.0 # укажите IP сервера для коррелятора KUMA
  kuma_storage:
    hosts:
      kuma-storage-1.example.com: # укажите FQDN сервера для хранилища KUMA
      ansible_host: 0.0.0.0 # укажите IP сервера для хранилища KUMA
      shard: 1
      replica: 1
      keeper: 1
      kuma-storage-2.example.com: # укажите FQDN сервера для хранилища KUMA
      ansible_host: 0.0.0.0 # укажите IP сервера для хранилища KUMA
      shard: 1
      replica: 2
      keeper: 2
      kuma-storage-3.example.com: # укажите FQDN сервера для хранилища KUMA
      ansible_host: 0.0.0.0 # укажите IP сервера для хранилища KUMA
      shard: 2
      replica: 1
      keeper: 3
      kuma-storage-4.example.com: # укажите FQDN сервера для хранилища KUMA
      ansible_host: 0.0.0.0 # укажите IP сервера для хранилища KUMA
      shard: 2
      replica: 2
```

Указывайте в файле инвентаря только необходимое количество нод коллекторов, корреляторов и хранилищ.

????????? ??????????? ????????????

?????????? ??????????? ????? ?? ???????????

По умолчанию используются следующие имена: `kuma`, `admsrv`, `api`, `monitoring`, `console`. Если по какой-либо причине вы хотите их изменить, следуйте инструкции ниже.

kuma

Добавьте в файл `param.yaml` следующий блок, где `kuma-new` - новое доменное имя для kuma. Пример:

```
- name: kuma_host
  source:
    value: "kuma-new"
```

А также в блоке `pki_host_list` измените `kuma` на новое доменное имя. Пример:

```
- name: pki_host_list
  source:
    value: "admsrv api console kuma-new monitoring"
```

api

Добавьте в файл `param.yaml` следующий блок, где `api-new` - новое доменное имя для api. Пример:

```
- name: api_host
  source:
    value: "api-new"
```

А также в блоке `pki_host_list` измените `api` на новое доменное имя. Пример:

```
- name: pki_host_list
  source:
    value: "admsrv api-new console kuma monitoring"
```

monitoring

Добавьте в файл param.yaml следующий блок, где `monitoring-new` - новое доменное имя для monitoring. Пример:

```
- name: monitoring_host
  source:
    value: "monitoring-new"
```

А также в блоке `pki_host_list` измените `monitoring` на новое доменное имя. Пример:

```
- name: pki_host_list
  source:
    value: "admsrv api console kuma monitoring-new"
```

admsrv

Добавьте в файл param.yaml следующий блок, где `ksc-new` - новое доменное имя для ksc. Пример:

```
- name: ksc_host
  source:
    value: "ksc-new"
```

А также в блоке `pki_host_list` измените `ksc` на новое доменное имя. Пример:

```
- name: pki_host_list
  source:
    value: "ksc-new api console kuma monitoring"
```

console

Добавьте в файл param.yaml следующие блоки, где `console-new` - новое доменное имя для console. Пример:

```
- name: nwc_host
  source:
    value: "console-new"
- name: flow_host
  source:
    value: "console-new"
- name: hydra_host
  source:
    value: "console-new"
```

```
- name: login_host
  source:
    value: "console-new"
- name: console_host
  source:
    value: "console-new"
- name: gateway_host
  source:
    value: "console-new"
```

А также в блоке `pki_host_list` измените `console` на новое доменное имя. Пример:

```
- name: pki_host_list
  source:
    value: "admsrv api console-new kuma monitoring"
```

Ниже расположен пример файла инвентаря со всеми измененными доменными именами

Пример файла инвентаря с переопределенными именами

```
schemaType: ParameterSet
schemaVersion: 1.0.1
namespace: ""
name: bootstrap
project: xdr

nodes:
- desc: cdt-1
  type: primary-worker
  host: 10.80.23.42
  access:
    ssh:
      user: user
      key: /home/user/.ssh/id_rsa
parameters:
- name: psql_dsn
  source:
    value: "postgres://postgres:postgres@10.80.23.40:5432"
- name: ingress_ip
```

```
source:
  value: 10.80.23.182
- name: ssh_pk
  source:
    path: /root/.ssh/id_rsa
- name: admin_password
  source:
    value: "AdminKit2015!"
- name: inventory
  source:
    value: "/home/user/inventory.yml"
- name: license
  source:
    value: "/home/user/license.key"
- name: smp_domain
  source:
    value: "smp.local"
- name: core_request_low_resources
  source:
    value: "100Gi"
# Начало блока с изменение доменных имен
- name: pki_host_list
  source:
    value: "admsrv-new api-new console-new kuma-new monitoring-new"
- name: kuma_host
  source:
    value: "kuma-new"
- name: admsrv_host
  source:
    value: "admsrv-new"
- name: api_host
  source:
    value: "api-new"
- name: monitoring_host
  source:
    value: "monitoring-new"
- name: nwc_host
  source:
    value: "console-new"
```

```
- name: flow_host
  source:
    value: "console-new"
- name: hydra_host
  source:
    value: "console-new"
- name: login_host
  source:
    value: "console-new"
- name: console_host
  source:
    value: "console-new"
- name: gateway_host
  source:
    value: "console-new"
# Конец блока с изменение доменных имен

# Начало блока, который не нужно изменять
- name: low_resources
  source:
    value: "true"
- name: default_class_replica_count
  source:
    value: "1"
- name: openbao_ha_mode
  source:
    value: "false"
- name: openbao_standalone
  source:
    value: "true"
```

????????? ?????????????? web-?????????

Для изменения сертификатов публичных служб существует 2 способа: указание конечных сертификатов, либо промежуточных для генерации сертификатов самой консолью. В обоих случаях в параметр необходимо предоставить сертификат в формате PEM и незашифрованный закрытый ключ.

Важно! Указать пользовательский сертификат для внешних служб возможно только до первоначальной установки!

???????????????? ????????????????? ?????????????????

Добавьте следующие параметры в файл установки, где `/path/cert.pem` - полный путь к промежуточному сертификату с незашифрованным закрытым ключом в формате PEM. На основе данного сертификата консолью будут сгенерированы сертификаты для публичных служб.

```
- name: intermediate_enabled
  source:
    value: "true"
- name: intermediate_bundle
  source:
    value: "/path/cert.pem"
```

???????????????????? ????????????????? ?????????????????

Добавьте следующие параметры в файл установки, где `/console/cert.pem`, `/admsrv/cert.pem`, `/api/cert.pem` - пути к сетификатам с незашифрованными закрытыми ключами в формате PEM для консоли SMP, KSC и API соответственно.

```
- name: intermediate_enabled
  source:
    value: "false"
- name: console_bundle
  source:
    value: "/console/cert.pem"
- name: admsrv_bundle
  source:
    value: "/admsrv/cert.pem"
- name: api_bundle
  source:
    value: "/api/cert.pem"
```

???????????????? ????????????????? ?????? ??????????????

Изменение некоторых параметров возможно и после установки системы, например, доменных имен или IP-адресов. Ниже рассмотрим данный процесс.

1. Перейти в директорию, где располагается kdt

2. Выполняем экспорт текущих параметров:

```
./kdt ec -e new_param.yaml
```

3. Вносим необходимые изменения в файл и применяем параметры, где `/path/new_param.yaml` - полный путь к ранее экспортированному и измененному файлу конфигурации

```
./kdt apply -i /path/new_param.yaml
```

Внимание! Применение параметров может занимать более 10-20 минут!

????????? ??????????

Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

????????? ?????????? /tmp

1. Определение переменной окружения

```
export TMPDIR=<new_directory>/tmp
```

где, <new_directory>/tmp - новый путь к директории /tmp

????????? ?????????????? ? ?????? ?????????????????????? ? ????????????? ?????????????? ?????????????? ??? ?????? ???????

1. Добавление в группу суперпользователей

```
sudo usermod -aG sudo user
```

2. Редактирование файла суперпользователей

```
sudo vi /etc/sudoers
```

Пример строки, которую необходимо добавить файл для разрешения повышения привелегий без ввода пароля

```
user ALL=(ALL) NOPASSWD:ALL
```

где, `user` - имя пользователя для которого выполняются операции

????????? ? ?????????????????????? ssh-???????

Ниже приведена последовательность действий для создания пользователя, генерация ключа ssh и распространение его на сторонний хост. Предварительно на удаленном хосте должен быть создан пользователь с таким же именем.

```
useradd user  
passwd user
```

```
mkdir /home/user/.ssh/  
ssh-keygen -t rsa  
ssh-copy-id -i /home/<user>/.ssh/id_rsa user@<host>
```

где, <host> - имя или IP-адрес удаленного хоста, куда необходимо доставить ключ

????????? ? ?????????????????? ssh-?????? ??? root

```
ssh-keygen -t rsa  
ssh-copy-id -i /root/.ssh/id_rsa root@<host>
```

где, <host> - имя или IP-адрес удаленного хоста, куда необходимо доставить ключ

??????????? ?????????????????? ????????? ? ????????? chrony

1. Установка chrony

```
sudo apt install chrony
```

2. (опционально) Редактирование файла конфигурации для указания собственного сервера синхронизации времени

```
vi /etc/chrony.conf
```

3. Запуск службы

```
sudo systemctl enable --now chronyd
```

4. Проверка синхронизации

```
sudo timedatectl | grep 'System clock synchronized'
```