Информация, необходимая для установки программного обеспечения: InfraVision

InfraVision — платформа для построения динамически обновляемой карты гибридной инфраструктуры.

Состав системы

Система состоит из четырёх компонентов:

1. **Сервер (iv-server)** — веб-интерфейс и постоянное реляционное хранилище данных
2. **Агрегатор (iv-aggregator)** — передаёт данные из шины данных на сервер
3. **Discovery (iv-discovery)** — сбор данных с сетевого оборудования и систем виртуализации по протоколам SNMP, REST API, SSH, WMI, NETCONF, IPMI (186 коннекторов)
4. **Агент (iv-agent)** — сбор данных с конечных устройств

Для коммуникации между компонентами используется **Apache Kafka**. Все серверные компоненты поставляются в виде Docker-образов.

Система предназначена для установки в закрытый контур без входящего доступа из интернета.

Варианты конфигурации

1. All-in-one сервер + клиентские агенты
2. Отдельная инфраструктура, серверная часть с сервис-воркерами + клиентские агенты
3. Отдельная инфраструктура, серверная часть, отдельные сервис-воркеры + клиентские агенты

Системные требования — Сервер

Уровень	Устройств	CPU	RAM	Диск	Сеть
Минимальный (тестирование)	до 200	8 ядер	16 ГБ	80 ГБ	1 Гбит/с
Рекомендуемый (production)	200–10 000	16 ядер	32 ГБ	200 ГБ SSD	10 Гбит/с
Высоконагруженный	10 000+	32 ядра	128 ГБ	1 ТБ NVMe SSD	10 Гбит/с

ОС: Ubuntu 22.04 LTS, Debian 12, RHEL 9 и совместимые.
Для 10 000+ устройств рекомендуется распределённая архитектура Master-Slave.

Необходимая инфраструктура

- **PostgreSQL 17** — ≥ 4 ядер, ≥ 16 ГБ RAM, ≥ 20 ГБ быстрого хранилища
- **Redis / Valkey** — 2 инсталляции или 2 базы данных (≥ 1 ГБ RAM каждая)
- **Apache Kafka 3.9.1**
- **HashiCorp Vault** — хранилище секретов и учётных данных
- SMTP-сервер для отправки уведомлений
- Nginx — обратный прокси / балансировщик (опционально)
- Мониторинг: Grafana, VictoriaMetrics, node-exporter, kafka-exporter, Alertmanager, vmalert (рекомендуется)

Системные требования — Агрегатор

- **CPU** — 4 ядра
- **RAM** — 8 ГБ

Необходимая инфраструктура

- **Redis / Valkey** — 1 ядро, ≥ 1 ГБ RAM

Необходимые сетевые доступы

- Серверу InfraVision (HTTPS)
- Шине данных Kafka (TCP)

Лицензирование

Для работы iv-aggregator требуется действующая лицензия. Настройте публичный ключ лицензии через переменную окружения:

```
IV_LICENSE_PUBLIC_KEY=<base64-encoded-public-key>
```

После запуска загрузите ключ через веб-интерфейс: **Приложения → Лицензии → Добавить**.

1. Установка серверной части (all-in-one)

1.1. Получение доступа к Container Registry

Получите учётные данные и адрес реестра у команды InfraVision и выполните авторизацию:

```
docker login <адрес реестра предоставляется командой InfraVision>
```

1.2. Загрузка дистрибутива

Скачайте и распакуйте архив `iv-setup-main.zip`.

1.3. Настройка переменных окружения

В файле `.env` укажите реальный IP-адрес вашей виртуальной машины:

```
# Замените <IP_ВАШЕЙ_ВМ> на реальный IP
KAFKA_ADVERTISED_LISTENERS=INTERNAL://kafka:9093,EXTERNAL://<IP_ВАШЕЙ_ВМ>:9
094
KAFKA_HOST=<IP_ВАШЕЙ_ВМ>
```

1.4. Запуск установки

```
chmod +x install.sh
bash install.sh
```

В процессе установки скрипт запросит:

- **Salt-строку** для генерации паролей и сертификатов (произвольный набор символов)
- **Доменное имя** (если есть; иначе — пропустить)
- **IP-адрес** виртуальной машины

По завершении скрипт выведет сгенерированные учётные данные для всех сервисов. **Сохраните их в надёжном месте** — они понадобятся при дальнейшей настройке.

1.5. Запуск контейнеров

```
docker compose up -d
```

Дождитесь запуска всех контейнеров (~10 минут). Проверить статус:

```
docker ps
```

Все контейнеры должны иметь статус `Up` или `healthy`.

1.6. Проверка после запуска

1. Откройте веб-интерфейс InfraVision: `http://<IP_ВАШЕЙ_ВМ>:8000`
2. Войдите с учётными данными администратора, полученными на шаге 1.4
3. Перейдите в **Администратор → Токены API** и убедитесь, что токены для `service-iv-aggregator` и `service-iv-discovery` созданы. Если токенов нет — добавьте их вручную, значения возьмите из переменных `IV_SERVER_TOKEN` и `DISCOVERY_SERVER_TOKEN`
4. Убедитесь, что контейнеры агрегатора запущены и работают
5. Перейдите в **Настройки → IV Services** и убедитесь, что все сервисы активны

1.7. Настройка прокси (опционально)

Необходимо, если агенты не имеют прямого доступа в интернет. Настройте маппинг через Nginx:

```
location /ip {
    proxy_pass https://api.ipify.org;
    proxy_ssl_server_name on;
}
```

```
location /ip-api {
    proxy_pass http://ip-api.com/json;
}
```

---

2. Переустановка серверной части

Для полной переустановки выполните команды по очереди:

```
docker stop $(docker ps -qa)
docker rm $(docker ps -qa)
docker rmi -f $(docker images -qa)
docker volume rm $(docker volume ls -q)
docker network rm $(docker network ls -q)
```

Затем повторите все шаги из раздела 1.

---

3. Настройка Discovery

3.1. Регистрация сервиса

После запуска `iv-discovery` автоматически регистрируется в IVServer — сервис отправляет heartbeat в Kafka каждые 60 секунд, после чего Aggregator создаёт запись в IVServer. Новый сервис появится в разделе **Операции → Управление → IV сервисы**.

3.2. Настройка IP-диапазонов

1. Перейдите в **Операции → Управление → IV сервисы**
2. Выберите сервис Discovery
3. В разделе **IP Ranges** нажмите **+ Добавить**
4. Укажите диапазон в формате CIDR, например: `192.168.1.0/24`

Совет: для больших сетей используйте несколько диапазонов `/24` вместо одного `/16` — это позволяет контролировать прогресс сканирования.

3.3. Процесс обнаружения

Процесс	Интервал	Описание
Discovery	24 часа	Полное сканирование всех IP-диапазонов
Polling	20 минут	Обновление данных известных устройств
SNMP Trap	По событию	Приём трапов от устройств (приостанавливается во время Discovery)

Первое сканирование начинается автоматически после настройки IP-диапазонов.

3.4. Поддерживаемые протоколы Discovery

Протокол	Порт	Коннекторов	Применение
SNMP v1/v2c/v3	UDP 161	68	Сетевое оборудование, принтеры, ИБП
SNMP Trap	UDP 162	—	Приём событий от устройств
REST API	TCP 443	61	Виртуализация, СХД, облака, SDN, контейнеры
SSH	TCP 22	29	Linux, сетевое оборудование
NETCONF	TCP 830	14	Cisco, Juniper, Huawei
IPMI	UDP 623	14	Серверы BMC (iLO, iDRAC, IPMI)
WMI	TCP 5985	—	Windows Server

3.5. Настройка учётных данных (Vault)

Учётные данные для подключения к оборудованию хранятся в HashiCorp Vault. Поиск выполняется в порядке приоритета: конкретный IP → подсеть → платформа по умолчанию → общее значение по умолчанию.

Поддерживаемые типы учётных данных:

Протокол	Поддерживаемые методы аутентификации
SNMP	v1/v2c (community string), v3 (username + auth/priv пароли, SHA/AES и другие)
SSH	Логин/пароль, приватный ключ, enable-пароль (Cisco/Huawei), jump host
API	Логин/пароль по типу платформы (VMware, OpenStack, Proxmox и другие)
WMI	Логин/пароль, домен, HTTPS
IPMI / NETCONF	Логин/пароль

Подробное описание структуры секретов — в документации по HashiCorp Vault.

3.6. Приоритет источников данных

Если устройство доступно по нескольким протоколам, используются данные источника с наивысшим приоритетом:

Источник	Приоритет	Описание
SSH	5	Минимальные данные
SNMP	10	Полные данные для сетевых устройств
IPMI	15	Базовые данные BMC
API	20	REST API коннекторы (базовый)
WMI	21	Windows (выше базового API)
NETCONF	25	Структурированные данные (YANG)

Источник	Приоритет	Описание
API:VMware, API:OpenStack	25	Платформенные API коннекторы
API:Redfish	30	Redfish API (наивысший приоритет)

3.7. Поддерживаемые платформы Discovery

- **Виртуализация:** VMware vCenter/ESXi, OpenStack, Proxmox VE, oVirt/zVirt/RHEV/ROSA, XenServer, Nutanix, KVM, SCVMM, VirtualBox
- **Контейнеры:** Kubernetes, OpenShift, Rancher, Docker/Swarm, Nomad, Portainer, AWS ECS/EKS, Azure AKS, GKE
- **Hardware BMC:** HPE iLO, Dell iDRAC, Lenovo XClarity, Cisco UCS, Supermicro, YADRO BMC, Vegman BMC (Redfish + IPMI)
- **СХД:** NetApp ONTAP, Dell EMC PowerStore/Unity, HPE 3PAR/Nimble, Huawei Dorado, Pure Storage, Infinidat, IBM FlashSystem, Hitachi VSP, MinIO, Ceph, VAST, Weka, DDN, YADRO Tatlin, TrueNAS, Synology/QNAP, Lenovo SANtricity
- **SDN:** VMware NSX, Cisco ACI/DNAC, Juniper Contrail, Arista CloudVision, Nokia NSP, Huawei Agile Controller, OpenDaylight
- **Облака:** OpenStack, Azure, GCP, Alibaba Cloud, Yandex Cloud, Selectel, VK Cloud, Hetzner, DigitalOcean, Oracle Cloud
- **Сетевое оборудование:** Cisco, Juniper, Huawei, MikroTik, Arista, Fortinet, Nokia, Extreme, Brocade, QTECH, Moxa (SNMP + SSH + NETCONF)
- **ОС:** Windows Server 2012+ (WMI), Linux (SSH), macOS (SSH)

3.8. Проверка работы Discovery

```
docker logs iv-discovery -f
```

Метрики Prometheus доступны по адресу: `http://<IP_СЕРВЕРА>:3112/metrics`

Типичные проблемы:

- **Устройство не обнаруживается** — проверьте сетевую доступность и учётные данные в Vault
- **"Unknown user name" (SNMPv3)** — неверное имя пользователя, проверьте настройки на устройстве
- **"Unsupported security level"** — Discovery автоматически понижает уровень: authPriv → authNoPriv → noAuthNoPriv
- **Vault недоступен** — при недоступности активируется circuit breaker на 5 минут

4. Установка агента (iv-agent)

Агент поставляется в виде нативных инсталляторов. Перед установкой скопируйте TLS-сертификаты, сгенерированные на этапе установки серверной части (путь: `/iv-setup-main/services/kafka`).

Системные требования агента

- **Минимальные:** 1 ядро CPU, 512 МБ RAM, 10 ГБ HDD; Windows 7 / Windows Server 2008 и выше; Linux glibc > v2.24
- **Рекомендуемые:** 2 ядра CPU, 1 ГБ RAM

Протестированные платформы

Платформа	Формат
Windows 7, 8, 10, 11	EXE
Windows Server 2008R2, 2012R2, 2016, 2019	EXE
Debian GNU/Linux 9, 10, 11, 12	DEB
Ubuntu 18.04, 20.04, 22.04, 24.04	DEB
Astra Linux 1.7 Воронеж, Орёл	DEB
SelectOS 1.1	DEB
macOS начиная с 10.14	PKG
Android начиная с 8.0 (ARM64, ARMv7, x86_64)	APK

Windows

```
C:\path\to\Infravision_<version>.exe /quiet /qn /norestart /log
C:\path\to\install.log ^
  PROPERTY1=value1 PROPERTY2=value2
```

Также поддерживается установка через файл `config.xml` с предустановленными параметрами. При установке создаётся пользователь `iv-user` с ограниченными правами на запуск службы `iv-agent`.

Linux (DEB)

```
sudo ./install.sh \
  --kafka-host <KAFKA_HOST> \
  --kafka-port 9094 \
  --kafka-user <KAFKA_USER> \
  --kafka-pass <KAFKA_PASS> \
  --cert-dir /opt/infravision/certs
```

Linux (RPM)

```
sudo rpm -i --force iv-agent-<version>.linux.x86_64.rpm

# Или через скрипт:
sudo ./install.sh \
  --kafka-host <KAFKA_HOST> \
  --kafka-port 9094 \
  --kafka-user <KAFKA_USER> \
  --kafka-pass <KAFKA_PASS> \
  --cert-dir /opt/infravision/certs
```

macOS

```
sudo ./install.sh \
  --pkg path/to/iv-agent-vX.Y.Z.pkg \
```

```
  --kafka-host <KAFKA_HOST> \
  --kafka-port 9094 \
  --kafka-user <KAFKA_USER> \
  --kafka-pass <KAFKA_PASS> \
  --kafka-ssl true
```

Android

Поддерживаются четыре метода конфигурации (в порядке приоритета):

Приоритет	Метод	Применение
1 (высший)	ADB Broadcast	Тестирование и отладка
2	MDM Managed Configuration	Корпоративное развёртывание
3	JSON конфигурационный файл	Ручная настройка
4 (низший)	Переменные окружения	Legacy-совместимость

Поддерживаемые MDM-системы: Microsoft Intune, VMware Workspace ONE, Samsung Knox, MobileIron, SOTI MobiControl.

Параметры установки агента

Параметр	Обязательный	По умолчанию	Описание
`--kafka-host`	✅	—	Адрес Kafka
`--kafka-user`	✅	—	Пользователь Kafka
`--kafka-pass`	✅	—	Пароль Kafka
`--kafka-port`		9094	Порт Kafka
`--timeout`		60 сек	Интервал сбора данных
`--status-timeout`		10 сек	Интервал отправки статуса
`--cert-dir`		`./cert`	Директория с сертификатами

---

5. Массовое развёртывание агентов через MDM

Платформа	Поддерживаемые инструменты
Windows	Microsoft Intune, SCCM, Group Policy (GPO)
macOS	Jamf Pro, Microsoft Intune
Linux	Ansible, Puppet, Chef
Android	Samsung Knox, VMware Workspace ONE, MobileIron

---

6. Проверка установки

После развёртывания агентов:

1. Откройте веб-интерфейс: `http://<IP_СЕРВЕРА>:8000`

2. Перейдите в **Devices → All Devices**
3. Убедитесь, что устройства с агентами появились в списке и поле **Last Seen** обновляется

Проверка статуса агента на устройстве

Windows:

```
Get-Service -Name "iv-agent"
Get-Process -Name "iv-agent"
```

Linux:

```
sudo systemctl status iv-agent
sudo journalctl -u iv-agent -f
```

macOS:

```
sudo launchctl list | grep infravision
sudo launchctl print system/pro.infravision.agent
```

Android:

```
adb logcat -s InfraVisionAgent:V
```

Мониторинг

- **Grafana:** `http://<IP_СЕРВЕРА>:3000` — дашборд мониторинга агентов и сервисов
- **Kafka UI:** `http://<IP_СЕРВЕРА>:8080` — просмотр топиков с данными и статусами устройств