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

- Как работают RFID и NFC картридеры
- Протоколы сетевой печати — IPP, RAW, LPD
- Как работает SNMP в контексте мониторинга принтеров
- Как работает Kerberos — основы для инженера Принтум
- Как работает SAML 2.0 — основы для инженера Принтум
- Как работает LDAP — основы для инженера Принтум
- Как работает HAProxy в кластере Принтум
- Как работает Redis Sentinel
- Зачем нужен NFS в кластерной конфигурации
- Как устроен SNMP и что Принтум получает от принтеров

??? ???????? RFID ? NFC ??????????

??? ???????? RFID ? NFC ??????????

??????????

Картридеры используются для идентификации пользователя по бесконтактной карте. В контексте Принтум — для авторизации на МФУ.

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

Стандарт	Частота	Тип карты	Дальность
EM-Marine (EM4100/EM4102)	125 кГц	Пассивная, read-only	до 10 см
HID Prox	125 кГц	Пассивная, read-only	до 10 см
MIFARE Classic	13,56 МГц	Пассивная, ISO 14443-A	до 10 см
MIFARE DESFire	13,56 МГц	Пассивная, ISO 14443-A	до 10 см
HID iCLASS	13,56 МГц	Пассивная	до 10 см
NFC (ISO 18092 / ISO 14443)	13,56 МГц	Пассивная/активная	до 20 см

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

Ридер генерирует переменное электромагнитное поле. Карта — пассивное устройство: она получает энергию через индуктивную связь и возвращает ответ с UID или шифрованными данными.

```
Ридер  →  RF-поле (13,56 МГц или 125 кГц)  →  Карта
Карта  →  модулированный сигнал с UID       →  Ридер
```

Ридер передаёт UID (или расшифрованные данные) по одному из интерфейсов:

- Wiegand 26/34/37 — стандарт для СКУД и МФУ;
- USB HID — эмуляция клавиатуры;
- RS-232 / RS-485 — последовательный интерфейс;
- OSDP — современный защищённый протокол.

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

Причина	Объяснение
Несовпадение стандарта	Ридер 125 кГц не читает MIFARE (13,56 МГц)
Слишком большое расстояние	Превышена рабочая дальность ридера
Помехи	Металлические поверхности экранируют поле
Повреждённая антенна карты	Физическое повреждение внутренней катушки
Конфликт UID	На заводских MIFARE Classic UID может совпасть у разных карт
Блокировка сектора	MIFARE Classic: сектор заперт неизвестным ключом
Карта в чехле с другими картами	Коллизия при антиколлизионной процедуре

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

ПринтМенеджер получает UID карты от ридера и сопоставляет его с учётной записью пользователя. Сопоставление может быть выполнено:

- через синхронизацию из LDAP/AD (атрибут карты в профиле пользователя);
- через Личный кабинет — пользователь самостоятельно привязывает карту.

После идентификации пользователь получает доступ к своим заданиям на МФУ.

????????? ??????? ?????? — IPP, RAW, LPD

Протоколы сетевой печати — IPP, RAW, LPD Обзор Протокол Порт RFC Тип соединения IPP 631 RFC 8011 HTTP/HTTPS (двунаправленный) RAW (JetDirect) 9100 — TCP (однонаправленный поток) LPD/LPR 515 RFC 1179 TCP (устаревший) IPP (Internet Printing Protocol) IPP построен поверх HTTP (порт 631) или HTTPS (IPPS, порт 443/631). Возможности: двунаправленная коммуникация: МФУ сообщает статус задания, уровень тонера и ошибки; аутентификация (Basic, Digest, Negotiate); шифрование через TLS; управление очередью: пауза, отмена, приоритет; поддержка IPP Everywhere — печать без установки драйверов. IPP используется в macOS (CUPS), Android, Windows 10+. RAW (JetDirect / порт 9100) Простой TCP-поток: клиент открывает соединение на порт 9100 и передаёт PDL (PCL, PostScript, PDF). Особенности: нет встроенной обратной связи о статусе; максимальная совместимость — поддерживают практически все МФУ; минимальные накладные расходы протокола; драйвер на стороне клиента формирует полный PDL. LPD/LPR Протокол 1980-х годов (RFC 1179). Используется в legacy-окружениях. Особенности: порт 515, TCP; очередь задаётся именем (queue name); ограниченные возможности управления; не поддерживает шифрование. Когда что использовать Ситуация Рекомендуемый протокол Современный МФУ, нужен статус задания IPP Максимальная совместимость, legacy-устройство RAW (9100) Печать из UNIX/Linux без CUPS LPD/LPR Защищённая сеть, нужно шифрование IPPS (IPP over TLS) Как Принтум использует эти протоколы ПринтМенеджер принимает задания от клиентов через установленные порты. Для отправки задания на физическое устройство ПринтМенеджер использует RAW (порт 9100) или IPP в зависимости от конфигурации МФУ. Протокол зависит от вендора и задаётся в карточке устройства (вкладка «Драйвер»). Для большинства устройств используется socket (порт 9100, аналог RAW): HP, Kyocera, Pantum, Sharp, Sindoh. Konica Minolta и Xerox поддерживают socket и ipp . Avison, Brother, Epson, Lexmark, Ricoh — socket и http . Протокол обновляется автоматически при изменении в интерфейсе и применяется после синхронизации. в Принтум.

??? ???????? SNMP ? ????????? ??????????? ?????????

Как работает SNMP в контексте мониторинга МФУ Что такое SNMP SNMP (Simple Network Management Protocol) — протокол для мониторинга и управления сетевыми устройствами. Работает по UDP, порт 161 (агент) и 162 (trap-сообщения). Версии Версия Аутентификация Шифрование Статус SNMPv1 Community string (plaintext) Нет Устаревший SNMPv2c Community string (plaintext) Нет Широко используется SNMPv3 Имя пользователя + пароль AES/DES Рекомендуется Community string Community string в SNMPv1/v2c — текстовая строка, выполняющая роль пароля. public — стандартное read-only значение; private — стандартное read-write значение; Передаётся в открытом виде (нет шифрования). Важно: многие устройства поставляются с дефолтными community strings. Их следует изменить. OID и MIB OID (Object Identifier) — иерархический идентификатор объекта мониторинга. .1.3.6.1.2.1.43.11.1.1.9.1.1 — уровень тонера (Printer MIB, RFC 3805) .1.3.6.1.2.1.43.10.2.1.4.1.1 — счётчик страниц .1.3.6.1.2.1.1.1.0 — sysDescr (описание устройства) MIB (Management Information Base) — база описаний OID. Для МФУ: Printer-MIB (RFC 3805), Host Resources MIB (RFC 2790). Операции SNMP Операция Описание GET Запрос значения одного OID GET-NEXT / GET-BULK Обход дерева OID SET Изменение значения (требует write-доступа) TRAP / INFORM Уведомление от устройства (push) Почему устройства иногда не отвечают Причина Решение Неверный community string Проверить настройки устройства SNMP отключён на устройстве Включить в настройках МФУ Брандмауэр блокирует UDP 161 Открыть порт Устройство поддерживает только SNMPv3, запрашивается v2c Привести версии в соответствие Устройство перегружено и не успевает ответить Увеличить timeout на стороне менеджера IP-фильтр на устройстве Добавить IP сервера мониторинга в разрешённые Как Принтум использует SNMP Принтум опрашивает МФУ по SNMP для получения: текущего статуса устройства (готов, ошибка, офлайн); уровней расходных материалов (тонер, барабан); счётчиков страниц; информации о лотках и бумаге. Принтум поддерживает SNMP v1, v2c и v3 . По умолчанию устройства опрашиваются по SNMP v1/v2c (community string). Для SNMP v3 необходимо отдельно указать логин и пароль — в параметрах сканирования сети в панели администратора Мониторинга (либо в карточке конкретного устройства). При наличии учётных данных v3 все устройства локации переводятся на опрос только по SNMP v3. и как настраивается community string для устройства. Почему разные вендоры показывают разные данные Стандарт Printer MIB (RFC 3805) определяет структуру данных, но не способ их передачи. Производители реализуют его по-своему, что приводит к различиям в поведении устройств. Производители по-разному реализуют Printer MIB (RFC 3805). Одни передают оставшийся ресурс в процентах, другие — в страницах, третьи не передают вообще и возвращают -1 или 0 . Некоторые устройства передают некорректные значения (253 , 254) — Принтум их игнорирует. Принтум компенсирует отсутствие данных: если устройство не

передаёт ресурс детали — Принтум рассчитывает его самостоятельно по счётчику отпечатков. Расчётные значения помечаются символом * , чтобы отличать их от данных, полученных напрямую от устройства. Разные детали используют разные счётчики: чёрный тонер — общий счётчик отпечатков, цветной тонер — счётчик цветных отпечатков, ролик АПД — счётчик АПД. Это нормальное поведение — не баг системы, а особенность реализации SNMP конкретными вендорами.

??? ???????? Kerberos — ?????? ??? ???????? ???????

??? ???????? Kerberos — ?????? ??? ???????? ???????

??????????

Kerberos — сетевой протокол аутентификации на основе тикетов (RFC 4120). Позволяет пользователям доказывать свою идентичность сервисам без передачи пароля по сети.

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

Компонент	Описание
KDC (Key Distribution Center)	Центр выдачи тикетов, обычно совмещён с AD
AS (Authentication Service)	Выдаёт TGT после проверки пароля
TGS (Ticket Granting Service)	Выдаёт тикеты сервисов на основе TGT
TGT (Ticket Granting Ticket)	Первичный тикет, подтверждает личность пользователя
ST (Service Ticket)	Тикет для доступа к конкретному сервису
Realm	Административный домен Kerberos (обычно = DNS-домен, UPPER CASE)

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

```
1. Клиент  →  AS:   AS-REQ (имя пользователя)
2. AS      →  Клиент: AS-REP (TGT, зашифрован ключом KDC)
3. Клиент  →  TGS:  TGS-REQ (TGT + имя сервиса)
4. TGS     →  Клиент: TGS-REP (Service Ticket)
```

```
5. Клиент  →  Сервис: AP-REQ (Service Ticket)
6. Сервис  →  Клиент: AP-REP (подтверждение)
```

?????? ???? ??????????? (SSO)

После получения TGT (шаги 1–2, выполняются при входе в Windows/Linux) все последующие обращения к сервисам (шаги 3–6) происходят автоматически — пользователь не вводит пароль повторно.

TGT имеет срок жизни — как правило, 8–10 часов (настраивается в политиках домена). По истечении требуется повторная аутентификация или продление (renewal).

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

- Пароль пользователя **никогда** не передаётся по сети.
- Используется симметричное шифрование (AES-256 в современных реализациях).
- Все тикеты ограничены по времени (timestamp + skew до 5 минут).
- Расхождение системных часов более 5 минут приводит к ошибке `KRB_AP_ERR_SKEW`.

????? ? SSO ? ???????

Принтум может использовать Kerberos/GSSAPI для прозрачной аутентификации пользователей в среде Active Directory — пользователь, вошедший в домен, получает доступ к сервисам Принтум без повторного ввода пароля.

TODO: уточнить — поддерживается ли Kerberos SSO напрямую или только через промежуточный IdP (например, ADFS/Keycloak).

??? ???????? SAML 2.0 — ?????? ??? ???????? ???????

??? ???????? SAML 2.0 — ?????? ??? ???????? ???????

??????????

SAML 2.0 (Security Assertion Markup Language) — стандарт обмена данными аутентификации и авторизации между IdP и SP на основе XML (OASIS, 2005).

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

Роль	Описание	Пример
IdP (Identity Provider)	Аутентифицирует пользователя и выдаёт assertions	ADFS, Keycloak, Okta, Azure AD
SP (Service Provider)	Принимает assertions и предоставляет доступ	Принтум, веб-приложение
Пользователь (User Agent)	Браузер, который перенаправляется между IdP и SP	Браузер пользователя

???? assertions

Тип	Содержимое
Authentication assertion	Кто пользователь, когда и как аутентифицирован
Attribute assertion	Атрибуты пользователя (email, группы, роль)
Authorization assertion	Права доступа (используется редко)

SP-initiated flow (???????? ????????????)

```
1. Пользователь  →  SP:    обращается к защищённому ресурсу
2. SP            →  Браузер: HTTP 302, AuthnRequest (SAML Request)
3. Браузер       →  IdP:   перенаправление с AuthnRequest
4. IdP           →  Пользователь: форма входа (если нет сессии)
5. IdP           →  Браузер: HTTP 200 + форма с SAML Response (POST binding)
6. Браузер       →  SP:    POST с SAML Response (assertion)
7. SP            →  Пользователь: доступ разрешён
```

Bindings

Binding	Механизм передачи
HTTP Redirect	AuthnRequest передаётся в URL (GET), подписывается query string
HTTP POST	Response передаётся в теле формы (Base64-encoded XML)
HTTP Artifact	Ссылка на assertion; SP забирает напрямую у IdP

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

- Assertions подписываются IdP с помощью X.509-сертификата.
- SP проверяет подпись перед обработкой.
- Assertion содержит `NotBefore` и `NotOnOrAfter` — временное окно действия.
- Для защиты от replay-атак используется `InResponseTo` и однократное использование assertion ID.

????? ? SSO ? ???????

Принтум (в роли SP) может принять аутентификацию от корпоративного IdP по SAML 2.0. Пользователь, уже вошедший в корпоративный IdP, получает доступ к Личному кабинету без ввода пароля.

TODO: уточнить — какие атрибуты assertion Принтум использует для сопоставления пользователя (NameID, email, UPN).

??? ???????? LDAP — ?????? ??? ???????? ???????

??? ???????? LDAP — ?????? ??? ???????? ???????

??????????

LDAP (Lightweight Directory Access Protocol, RFC 4511) — протокол доступа к каталогу. Чаще всего используется для хранения учётных записей пользователей (Microsoft Active Directory, OpenLDAP, FreeIPA).

?????

Режим	Порт
LDAP (plaintext / StartTLS)	389
LDAPS (TLS с первого байта)	636
AD Global Catalog	3268 / 3269 (LDAPS)

????????? ???????? (DIT)

Каталог — дерево записей (Directory Information Tree).

```
dc=example,dc=com
├── ou=Users
│   ├── cn=Ivan Petrov
│   └── cn=Anna Sidorova
├── ou=Groups
│   └── cn=PrintUsers
└── ou=Computers
```

```
    └── cn=WS001
```

Атрибут	Расшифровка	Пример
DC	Domain Component	`dc=example,dc=com`
OU	Organizational Unit	`ou=Users`
CN	Common Name	`cn=Ivan Petrov`
DN	Distinguished Name	`cn=Ivan Petrov,ou=Users,dc=example,dc=com`
SN	Surname	`sn=Petrov`
UID	User ID (OpenLDAP)	`uid=ipetrov`
sAMAccountName	Логин Windows (AD)	`ipetrov`
userPrincipalName	UPN (AD)	`ipetrov@example.com`

Bind (??????????????)

Bind — операция аутентификации клиента на LDAP-сервере.

Тип	Описание
Simple bind	DN + пароль в открытом виде (требует TLS)
Anonymous bind	Без аутентификации (ограниченный доступ)
SASL / GSSAPI	Kerberos или другие механизмы

Сервис, читающий каталог, выполняет bind с отдельной учётной записью (service account / bind DN).

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

Фильтры определяют, какие записи возвращает запрос.

Фильтр	Значение
`(objectClass=user)`	Все объекты типа user
`(sAMAccountName=ipetrov)`	Конкретный пользователь
`(&(objectClass=person)(mail=*))`	Все люди с заполненным email
`(|(ou=IT)(ou=Finance))`	Записи из OU IT или Finance
`(memberOf=cn=PrintUsers,ou=Groups,dc=example,dc=com)`	Члены группы PrintUsers

??? ??????? ??????????? LDAP

Принтум подключается к LDAP/AD для:

- синхронизации учётных записей пользователей;
- получения атрибутов (имя, email, номер карты, группы);
- проверки членства в группах для управления доступом;
- аутентификации пользователей через bind с их учётными данными.

Синхронизация выполняется периодически. При отказе LDAP Принтум продолжает работу с последними синхронизированными данными (degraded mode).

TODO: уточнить — какие именно атрибуты AD/LDAP синхронизируются в Принтум и как часто выполняется синхронизация.

??? ???????? HAProxy ? ???????? ???????

??? ???????? HAProxy ? ???????? ???????

??????????

HAProxy (Отказоустойчивая конфигурация Proxy) — программный балансировщик нагрузки и прокси уровней L4/L7. В кластере Принтум используется для распределения запросов между узлами ПринтМенеджера.

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

Алгоритм	Описание	Когда применять
`roundrobin`	Запросы распределяются поочерёдно	Однородная нагрузка, равноценные узлы
`leastconn`	Запрос идёт к узлу с наименьшим числом соединений	Долгоживущие соединения
`source`	Хэш по IP клиента, клиент всегда попадает на один узел	Sticky sessions без cookies
`uri`	Хэш по URI	Кэшируемые сервисы
`random`	Случайный выбор из N лучших	Упрощённое распределение
`first`	Первый доступный узел в порядке списка	Active-passive

Health Check

HAProxy периодически проверяет доступность backend-узлов.

Тип	Описание
TCP check	Проверяет только открытие TCP-соединения
HTTP check	Отправляет HTTP-запрос, проверяет статус ответа
Agent check	Опрашивает внешний агент на узле для получения веса

Параметры проверки:

- `inter` — интервал между проверками (например, 2s);
- `rise` — количество успешных проверок для возврата в пул;
- `fall` — количество неудачных проверок для исключения из пула.

Failover

```
Штатная работа:
  Клиент → HAProxy → [PM1] [PM2]

Отказ PM1:
  HAProxy детектирует неудачу health check (после N попыток)
  PM1 помечается DOWN
  Все новые запросы → PM2

Восстановление PM1:
  HAProxy детектирует успех health check (после M попыток)
  PM1 возвращается в пул
  Запросы снова распределяются между PM1 и PM2
```

Во время failover активные соединения к PM1 прерываются — клиент получает ошибку или retry.

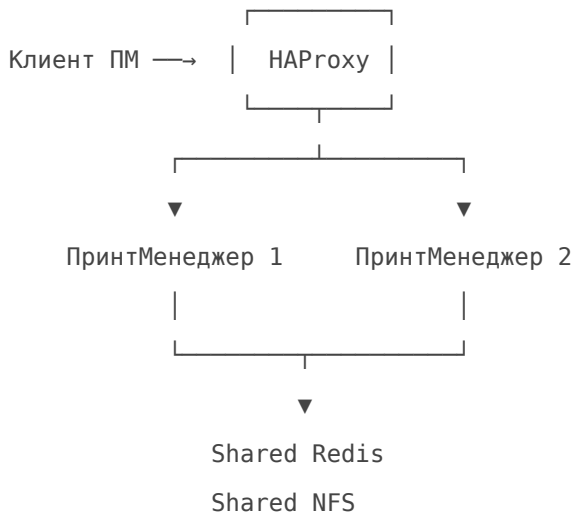
Sticky Sessions

Если ПринтМенеджер хранит сессионное состояние локально, необходима привязка клиента к узлу:

- `cookie` — HAProxy вставляет cookie с идентификатором узла;
- `source` — привязка по IP.

В Active-Active кластере Принтум сессионное состояние хранится в Redis — sticky sessions не требуются.

???? HAProxy ? Active-Active ????????????? ???????



- Оба узла активны, обрабатывают запросы одновременно.
- HAProxy распределяет входящие HTTP-запросы (Клиент ПМ, API, Встроенное приложение).
- При отказе одного узла HAProxy автоматически направляет весь трафик на второй.

В стандартной конфигурации используется алгоритм **Round Robin** с включением **Sticky Session**. Параметры таймаутов: сервера — 30 сек, клиента — 30 сек, соединения — 5 сек. Повторы (retries) не требуются. Принтум.

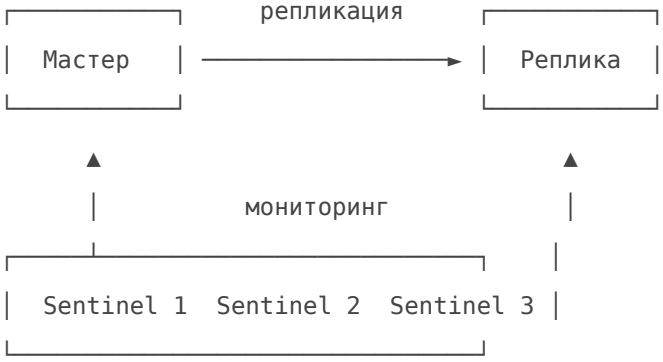
??? ???????? Redis Sentinel

??? ???????? Redis Sentinel

??????????

Redis Sentinel — система мониторинга и автоматического failover для Redis. Обеспечивает высокую доступность без ручного вмешательства при отказе мастера.

?????????



Минимальная рекомендуемая конфигурация: **3 Sentinel-процесса** (нечётное число для кворума).

?????? (quorum)

Кворум — минимальное количество Sentinel-узлов, которые должны согласиться, что мастер недоступен, прежде чем начнётся выборы нового мастера.

Параметр	Рекомендация
Число Sentinel	3
Кворум	2

Если кворум не достигнут, failover не выполняется — защита от split-brain.

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

1. Sentinel-ы обнаруживают, что мастер не отвечает в течение `down-after-milliseconds`.
2. Каждый Sentinel помечает мастер как **SDOWN** (субъективно недоступен).
3. Когда `quorum` Sentinel-ов помечают мастер — он становится **ODOWN** (объективно недоступен).
4. Один из Sentinel-ов избирается **лидером** через алгоритм Raft.
5. Лидер выбирает лучшую реплику для повышения до мастера.

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

- Реплика должна быть online.
- Предпочитается реплика с наименьшим `slave-priority`.
- При равном приоритете — с наибольшим `replication_offset` (меньше отставание).
- При равном offset — с наименьшим runid (детерминированный выбор).

??????? failover

```
1. Лидер отправляет SLAVEOF NO ONE выбранной реплике → она становится мастером
2. Остальные реплики переключаются на нового мастера (SLAVEOF <new-master>)
3. Клиенты получают уведомление через Pub/Sub или переспрашивают адрес мастера
4. Старый мастер (после восстановления) становится репликой нового
```

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

Клиент подключается к Sentinel для получения адреса текущего мастера:

```
SENTINEL get-master-addr-by-name <master-name>
→ возвращает IP:PORT актуального мастера
```

Redis-клиенты с поддержкой Sentinel (Jedis, StackExchange.Redis, redis-py) делают это автоматически.

???? Redis Sentinel ? ???????? ??????????????

Redis используется в кластере Принтум для:

- хранения пользовательских сессий;
- очередей заданий между узлами ПринтМенеджера;
- кэширования данных конфигурации.

Redis Sentinel обеспечивает автоматическое переключение при отказе Redis-мастера, чтобы оба узла ПринтМенеджера сохраняли доступ к данным.

В стандартной конфигурации разворачиваются **3 Sentinel-процесса** — по одному на каждый узел ПринтМенеджера (переменная REDIS_SENTINEL_LIST содержит 3 IP-адреса). TODO: уточнить точное значение кворума — в документации явно не указано. в стандартной конфигурации Принтум.

????? ????? NFS ? ?????????? ????????????

Зачем нужен NFS в кластерной конфигурации Назначение NFS (Network File System) — сетевой протокол для совместного доступа к файловой системе (RFC 7530 для v4). В кластере Принтум используется как общее хранилище заданий печати. Версии NFS Версия Статус Ключевые особенности NFSv3 Широко используется Stateless, UDP/TCP, порт 2049 NFSv4 Рекомендуется Stateful, только TCP, порт 2049, ACL, Kerberos NFSv4.1/4.2 Современный pNFS, параллельный доступ, улучшенная производительность Принцип работы NFS-сервер экспортирует директорию: /exports/printum-jobs NFS-клиенты (узлы ПринтМенеджера) монтируют её: mount -t nfs nfs-server:/exports/printum-jobs /var/printum/jobs Оба клиента видят одну и ту же файловую систему: ПринтМенеджер 1: /var/printum/jobs/ ←—— один каталог ——→ ПринтМенеджер 2 Почему задания нужно хранить централизованно В Active-Active кластере любой узел может получить запрос от любого пользователя. Без общего хранилища: Пользователь отправил задание через ПринтМенеджер 1. Он подходит к МФУ и его запрос попадает на ПринтМенеджер 2. ПринтМенеджер 2 не видит задания — release-печать невозможна. С NFS: Задание сохраняется в общую директорию. Любой узел читает задание и отправляет его на МФУ. Роль NFS в Active-Active кластере Принтум

ПринтМенеджер 1 ПринтМенеджер 2 /var/printum/jobs /var/printum/jobs (одна и та же физическая директория) Файлы заданий создаются и читаются через единую точку монтирования. HAProxy определяет, какой узел обработает запрос release, но оба могут прочитать задание. NFS-сервер должен быть отказоустойчивым сам по себе (например, HA NFS с DRBD, Pacemaker или Ceph). Производительность и надёжность Аспект Рекомендация Сеть Выделенная сеть хранилищ (10 Гбит/с) или отдельный VLAN NFS-опции монтирования rsize=1048576,wsize=1048576,hard,intr,timeo=600 Блокировки (locks) NFSv4 использует встроенные блокировки; NFSv3 требует rpc.lockd Отказ NFS При недоступности NFS (hard mount) операции I/O зависают — критично для ПринтМенеджер Рекомендуемая версия NFS — 4 (nfsvers=4). Допустимые значения nfsvers: 3, 4, 4.2. Стандартные опции монтирования: addr=NFS_ADDR,nolock,soft,rw . Для явного указания версии: addr=NFS_ADDR,nolock,soft,rw,nfsvers=4 — задаётся в переменной DRIVER_OPTS_O в файле .env ПринтМенеджера.

??? ??????? SNMP ? ??? ??????? ????????? ?? ?????????

Кратко SNMP (Simple Network Management Protocol) — протокол опроса сетевых устройств. Принтум использует SNMP для получения данных от МФУ и МФУ без установки агентов на устройства. Как устроена информация в МФУ Каждый МФУ содержит базу данных объектов — MIB (Management Information Base) . Каждый объект в MIB имеет уникальный адрес — OID (Object Identifier) . OID — это числовой путь, например: 1.3.6.1.2.1.43.11.1.1.9.1.1 Структура OID иерархична — как путь в файловой системе: 1.3.6.1 — интернет 1.3.6.1.2 — management 1.3.6.1.2.1 — mib-2 (стандартные объекты) 1.3.6.1.2.1.43 — Printer MIB (RFC 3805) 1.3.6.1.2.1.43.11 — prtMarker (данные о расходных материалах) Что можно получить по OID Данные OID-ветка Примечание Модель устройства 1.3.6.1.2.1.25.3.2.1.3 hrDeviceDescr Серийный номер 1.3.6.1.2.1.43.5.1.1.17 prtGeneralSerialNumber Счётчик отпечатков 1.3.6.1.2.1.43.10.2.1.4 prtMarkerLifeCount Оставшийся ресурс тонера 1.3.6.1.2.1.43.11.1.1.9 prtMarkerSuppliesLevel Максимальный ресурс 1.3.6.1.2.1.43.11.1.1.8 prtMarkerSuppliesMaxCapacity Статус устройства 1.3.6.1.2.1.25.3.5.1.1 hrPrinterStatus IP-адрес 1.3.6.1.2.1.4.20.1.1 ipAdEntAddr Версии SNMP Версия Аутентификация Когда использовать SNMPv1 Community string (открытый текст) Старые устройства SNMPv2c Community string (открытый текст) Большинство современных устройств SNMPv3 Логин + пароль + шифрование Повышенные требования к безопасности Принтум поддерживает SNMPv1 и SNMPv2c. Community string по умолчанию: public . Почему разные вендоры показывают разные данные Стандарт Printer MIB (RFC 3805) описывает общую структуру. Но производители реализуют его по-своему: Одни передают оставшийся ресурс в процентах. Другие — в страницах. Третьи не передают вообще и возвращают -1 или 0 . Некоторые устройства передают некорректные значения (253 , 254). Принтум компенсирует это: Если устройство передаёт данные — использует их напрямую. Если не передаёт — рассчитывает самостоятельно по счётчику отпечатков. Расчётные значения помечаются символом * . Как проверить SNMP вручную Команда snmpwalk позволяет просмотреть все данные устройства: snmpwalk -v 2c -c public <ip-адрес-МФУ> Получить конкретный OID: snmpget -v 2c -c public 1.3.6.1.2.1.43.10.2.1.4.1.1 Это полезно при диагностике — если snmpwalk не отвечает, устройство не будет обнаружено Сетевым агентом. Связанные страницы Как рассчитывается ресурс деталей МФУ не обнаружен при сетевом сканировании Сетевой агент — справка по компоненту