ООО «Астра Консалтинг»

Астра ИС МД (Инфраструктурные Сервисы)

Модуль Astra Секреты

ИНСТРУКЦИЯ ПО УСТАНОВКЕ ЭКЗЕМПЛЯРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПРЕДОСТАВЛЕННОГО ДЛЯ ПРОВЕДЕНИЯ ЭКСПЕРТНОЙ ПРОВЕРКИ

Москва, 2026

Содержание

1. Область применения

Настоящий документ предназначен для установки экземпляра программного обеспечения «Astra Секреты» (Модуль управления секретами), представленного для проведения экспертной проверки.
Документ описывает процесс установки, инициализации, конфигурации и проверки работоспособности Системы на демонстрационном стенде.

2. Установка системы

2.1. Технические требования к оборудованию

Для локальной инсталляции требуется виртуальный сервер (ВС) с минимальными техническими характеристиками, представленными в Таблице 1, и операционной системой Astra Linux Special Edition (сертификат соответствия ФСТЭК России № 2557).

Таблица 1 — Минимальные требования к ВС

Вид ресурса	**Ед. измерения**	**Требуемый объем**
Частота процессора vCPU	ГГц	2.3
vCPU количество процессоров и ядер	шт.	4
Объем оперативной памяти	Гб	8
Объем жесткого диска	Гб	100
Сетевой интерфейс	шт.	1
Скорость сетевого интерфейса	Мб/с	1000

Примечание: для HA-кластера минимальное количество узлов — 3. Каждый узел должен удовлетворять требованиям Таблицы 1.

2.2. Минимальные требования к сторонним компонентам и/или системам, необходимым для установки и работы ПО

Для установки Системы на стенде используется система контейнерной изоляции приложений Docker, минимальная версия v24.0.2.
Установка системы контейнерной изоляции приложений Docker осуществляется по инструкции:
https://wiki.astralinux.ru/pages/viewpage.action?pageId=158601444

Дополнительные компоненты (опционально):

- PostgreSQL версии 15 и выше — при использовании PostgreSQL в качестве Storage Backend;
- HSM-модуль с поддержкой PKCS#11 — при использовании Auto-Unseal через аппаратный модуль;

LDAP-сервер (ALDPro / FreeIPA / SambaDC) — при использовании аутентификации через LDAP;
Kubernetes-кластер — при использовании Kubernetes Auth Method и CSI Provider.

2.3. Архитектура развёртывания

Система поддерживает следующие варианты развёртывания:

Одиночный узел (standalone) — для тестирования и экспертной проверки;
HA-кластер (3+ узла) — для продуктивной эксплуатации с обеспечением отказоустойчивости.

В рамках экспертной проверки развёртывается одиночный узел в контейнере Docker с Storage Backend Raft.

3. ВС на Стенде

Технические характеристики демонстрационного стенда представлены в Таблице 2.
Таблица 2 — Характеристики демонстрационного стенда

Машина	**Параметр**	**Требование**
astra-secrets-demo	OS	Astra Linux SE 1.8
	CPU	4
	RAM	8 Gb
	HDD	100 Gb

4. Доступ к Стенду

UI доступна извне по адресу http://158.160.196.133:9200/ui
Токен: s.9xWwwHuteTdPycdCcDirVMyn

5. Доступ к Стенду через REST API

Доступ к REST API для управления Системой осуществляется с использованием root-токена, пример выполнения запроса через утилиту curl:

```
curl --request GET \
  --url http://158.160.196.133:9200/v1/sys/health \
  --header "X-Vault-Token: <ROOT_TOKEN>"
```

API Системы доступно по адресу http://158.160.196.133:9200/.

8. Проверка работоспособности

Для проверки работоспособности Системы откройте в браузере:

https://158.160.196.133:9200/ui/vault/dashboard

На главной странице отображается статус кластера: Initialized = true, Sealed = false, версия, тип хранилища.

Также проверьте доступность REST API:
GET https://158.160.196.133:9200/v1/sys/health

Ожидаемый ответ (HTTP 200):
{"initialized":true,"sealed":false,"standby":false,...}

9. Просмотр логов работы системы

Для просмотра аудит-логов всех операций с секретами откройте в браузере:
https://158.160.196.133:9200/ui/vault/audit

В журнале отображается: кто выполнил действие, когда, какой тип операции, с какого IP-адреса, результат выполнения.

Также аудит-логи доступны через REST API:
GET https://158.160.196.133:9200/v1/sys/audit
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn

10. Настройка методов аутентификации

Все методы аутентификации настраиваются через Web UI или через REST API. Для доступа к API используйте токен, полученный при авторизации в UI.

10.1. Просмотр включённых методов

Откройте в браузере:
https://158.160.196.133:9200/ui/vault/access

В разделе Auth Methods отображаются доступные методы аутентификации: Token, LDAP, AppRole, Kubernetes, OIDC и другие.

10.2. Включение метода LDAP

Через UI: Access → Auth Methods → Enable new method → LDAP → Enable.

Параметры подключения (URL, binddn, userdn, groupdn, starttls) указываются в форме настроек метода в UI.

Через REST API:
POST https://158.160.196.133:9200/v1/sys/auth/ldap
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn

POST https://158.160.196.133:9200/v1/auth/ldap/config
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn
Body: {
  "url": "ldaps://ldap.example.com:636",
  "binddn": "cn=admin,dc=example,dc=com",
  "bindpass": "<LDAP_PASSWORD>",
  "userdn": "ou=users,dc=example,dc=com",
  "groupdn": "ou=groups,dc=example,dc=com",
  "insecure_tls": false,
  "starttls": true
}

10.3. Включение метода AppRole

Через UI: Access → Auth Methods → Enable new method → AppRole → Enable.

Для проверки работы AppRole создайте тестовую роль:

Через UI: Access → Auth Methods → AppRole → Create role.

Через REST API:
POST https://158.160.196.133:9200/v1/sys/auth/approle
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn

POST https://158.160.196.133:9200/v1/auth/approle/role/cicd
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn
 Body: {
  "token_policies": "cicd-policy",
  "token_ttl": "1h",
  "token_max_ttl": "4h"
 }

GET https://158.160.196.133:9200/v1/auth/approle/role/cicd/role-id
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn

POST https://158.160.196.133:9200/v1/auth/approle/role/cicd/secret-id
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn

10.4. Включение метода Kubernetes

Через UI: Access → Auth Methods → Enable new method → Kubernetes → Enable.

Настройка подключения к K8s-кластеру через UI (форма настроек метода) или через REST API:

```
POST https://158.160.196.133:9200/v1/sys/auth/kubernetes
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn

POST https://158.160.196.133:9200/v1/auth/kubernetes/config
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn
Body: {
  "kubernetes_host": "https://kubernetes.default.svc:443",
  "kubernetes_ca_cert": "<CA_CERT>",
  "token_reviewer_jwt": "<REVIEWER_JWT>"
}

---
```

11. Включение Secrets Engines

Все Secrets Engines настраиваются через Web UI или через REST API.

11.1. Просмотр включённых движков

Откройте в браузере:
https://158.160.196.133:9200/ui/vault/secrets

Отображаются доступные Secrets Engines: KV, Database, PKI, Transit, SSH и другие.

11.2. Включение KV (хранилище ключ-значение)

Через UI: Secrets → Enable new engine → KV → Next → Enable.

Для проверки создайте тестовый секрет через UI (кнопка Create secret) или через REST API:

```
POST https://158.160.196.133:9200/v1/secret/data/test
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn
Body: {
  "data": {
    "username": "demo",
    "password": "demo-password"
```

```
  }
 }
```

Проверьте чтение:
GET https://158.160.196.133:9200/v1/secret/data/test
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn

11.3. Включение Database (динамические учётные данные)

Через UI: Secrets → Enable new engine → Database → Next → Enable.

Настройка подключения к PostgreSQL через UI (раздел Configure) или через REST API:
POST https://158.160.196.133:9200/v1/database/config/postgresql
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn

```
 Body: {
   "plugin_name": "postgresql-database-plugin",
   "allowed_roles": ["readonly", "readwrite"],
   "connection_url":
"postgresql://{{username}}:{{password}}@postgres:5432/mydb?sslmode=disable",
   "username": "admin",
   "password": "<DB_ADMIN_PASSWORD>"
 }
```

Создание роли:
POST https://158.160.196.133:9200/v1/database/roles/readonly
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn

```
 Body: {
   "db_name": "postgresql",
   "creation_statements": "CREATE ROLE \"{{name}}\" WITH LOGIN PASSWORD
'{{password}}' VALID UNTIL
 '{{expiration}}'; GRANT SELECT ON ALL TABLES IN SCHEMA public TO \"{{name}}\";",
   "default_ttl": "1h",
   "max_ttl": "24h"
 }
```

11.4. Включение Transit (шифрование как сервис)

Через UI: Secrets → Enable new engine → Transit → Next → Enable.

Создание ключа шифрования через UI (кнопка Create encryption key) или через REST API:
POST https://158.160.196.133:9200/v1/transit/keys/my-key
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn

Шифрование данных:

```
POST https://158.160.196.133:9200/v1/transit/encrypt/my-key
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn
Body: {
  "plaintext": "SGVsbG8gV29ybGQ="
}
```

Расшифрование:

```
POST https://158.160.196.133:9200/v1/transit/decrypt/my-key
Headers: X-Vault-Token: s.9xWwwHuteTdPycdCcDirVMyn
Body: {
  "ciphertext": "<CIPHERTEXT_FROM_RESPONSE>"
}
```