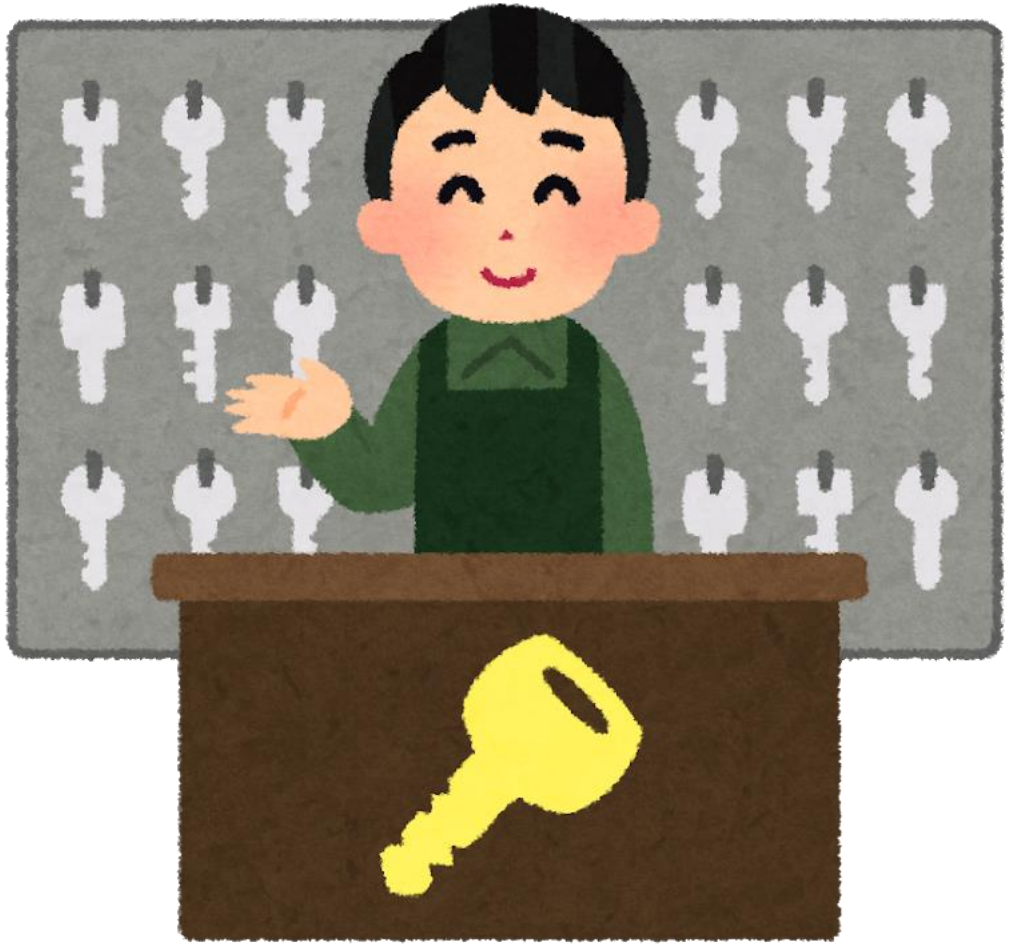
HashiCorp
Vault Enterprise概要資料

ネットワールド　HashiCorpチーム

もくじ

1. シークレットとは？
2. 今のシークレット管理と問題点
3. Vaultとは？
4. Vaultの機能説明
5. Vault Enterprise
6. まとめ



シークレットとは？

Vault とは？

シークレットを管理するソリューションです

シークレットって管理してますか？

そもそもシークレットってなんですか？

シークレットにはこんなものがあります

クラウドの IAM (アクセスキーとパスワード)

トークン / TLS証明書 / ユーザー名とパスワード

流出や紛失してはいけないもの

パスワードは管理してますよ(笑)

自動保存 オン

ファイル ホーム 挿入 ページ レイアウト 数式 データ 校閲 表示 ヘルプ

貼り付け 切り取り コピー 書式のコピー/貼り付け クリップボード

Yu Gothic 11 フォント

E7 gseyh006

	A	B	C	D	E	F	G
1	#	ID	架空の名前	架空の住所	パスワード		
2	1	84831	安部 知世	福島県	d88ik001		
3	2	30215	北田 兼	山口県	72wyc002		
4	3	15212	秋元 勇	福井県	xr7b2003		
5	4	59739	中尾 まさし	愛知県	ra2d4004		
6	5	57179	末永 ケンイチ	東京都	th467005		
7	6	64816	天野 優	愛知県	gseyh006		
8	7	11473	金山 智花	東京都	mgfyp007		
9	8	61040	野田 奈央	新潟県	mj74u008		
10	9	90143	今泉 寛治	兵庫県	vsfc8009		
11	10	36382	大村 宏行	北海道	ut3u500a		
12	11	85614	板垣 小雁	千葉県	d5wj600b		
13	12	78453	小峰 恵美	岡山県	b8dkp00c		

今（よくある）シークレットの管理

- 必要なユーザーからのメールもしくはワークフローで問い合わせ
- 管理者が手動で発行
- 有効期限は都度確認するが、実質Expireされない
- 作成した履歴や値はExcel管理

今（よくある）シークレット管理の問題

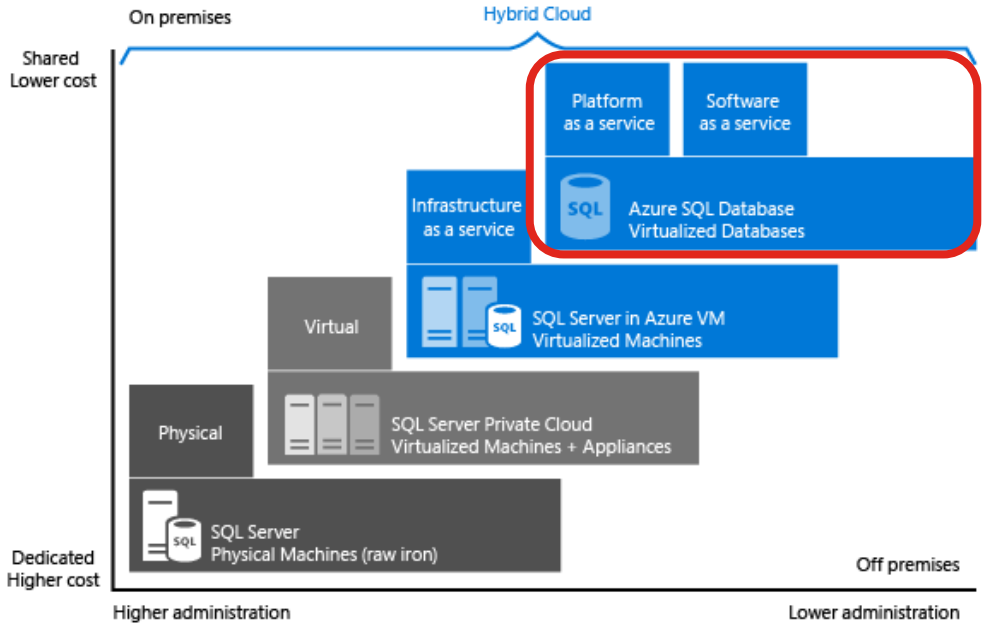
- 発行までに時間がかかる
- 手動で行うフローが多い
- 発行したシークレットに有効期限を設定できない（しづらい）
- 有効期限後に利用されているかわからない
- 発行後のアクセスコントロールができない（しづらい）
- シークレット管理のための管理表（Excel）などの保護、管理
- ログ管理

利用者側のシークレットの使い方の問題

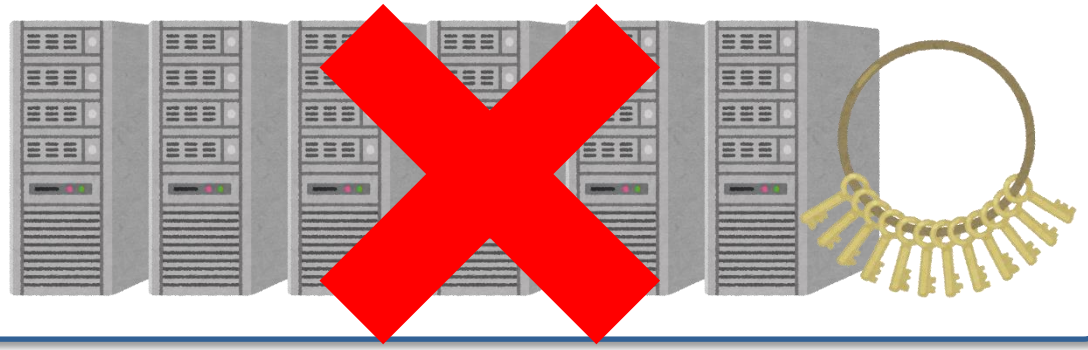
- シークレットの使い回し、長期利用
- アプリの設定にシークレットを記入
- シークレットの乱立、管理が煩雑

そして、クラウド全盛の現代では

PaaS：Platform as a Service で提供されるデータベースなど



現実的には、従来のように仮想マシン単位でパスワードを管理するような手法は通用しない

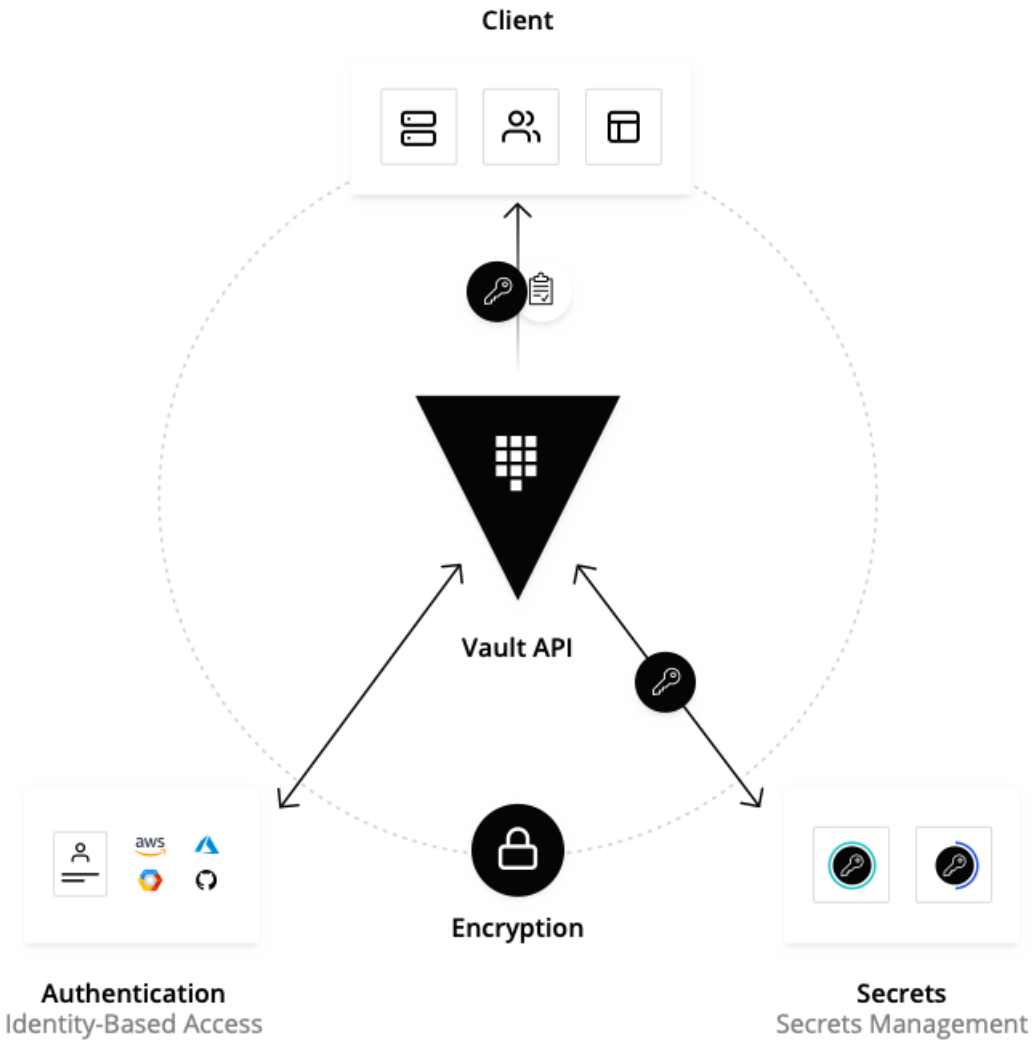


そこで…

HashiCorp Vault

HashiCorp Vault

Vaultとは？　https://www.vaultproject.io



- シークレットライフサイクルの集中管理
- データプロテクション: API-Drivenな暗号化
- Advanced データプロテクション

Vaultとは？

1. Secretの中央管理（Centralization）
2. 暗号化（Encryption）
3. 認証（Authentication）
4. 認可（Authorization）
5. 鍵交換（Rotation）

これ全部Vaultで出来ます

Vaultによるシークレット管理

Before

- シークレットの使い回し、長期利用
- アプリの設定にシークレットを記入
- シークレットの乱立、管理が煩雑
- アクセスコントロールの設計が困難

シークレット管理の手間や、
同じシークレットを長期間利用し続ける
ことのリスク

After

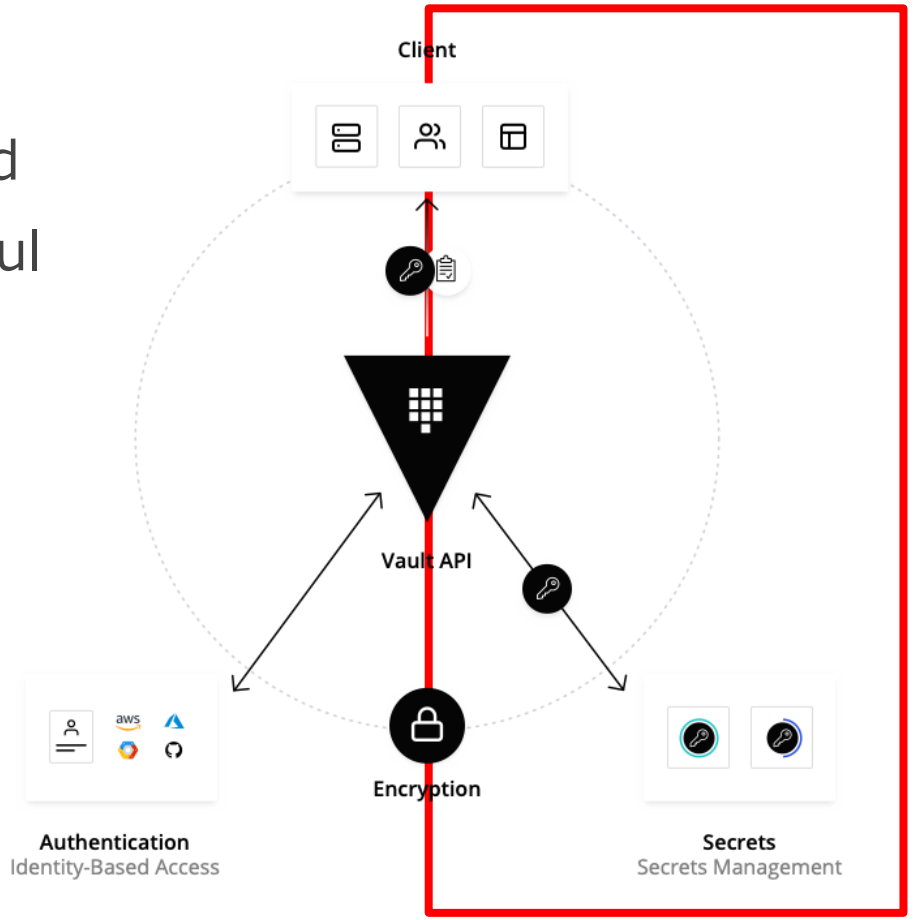
- シークレットのシングルレポジトリ
- Vaultから様々なシークレットを発行
- 期限(TTL)付与しシークレットのライフサイクルをコントロール
- 細かな権限管理

**シークレット管理を改善し、
短期間で新しいシークレットを発行**

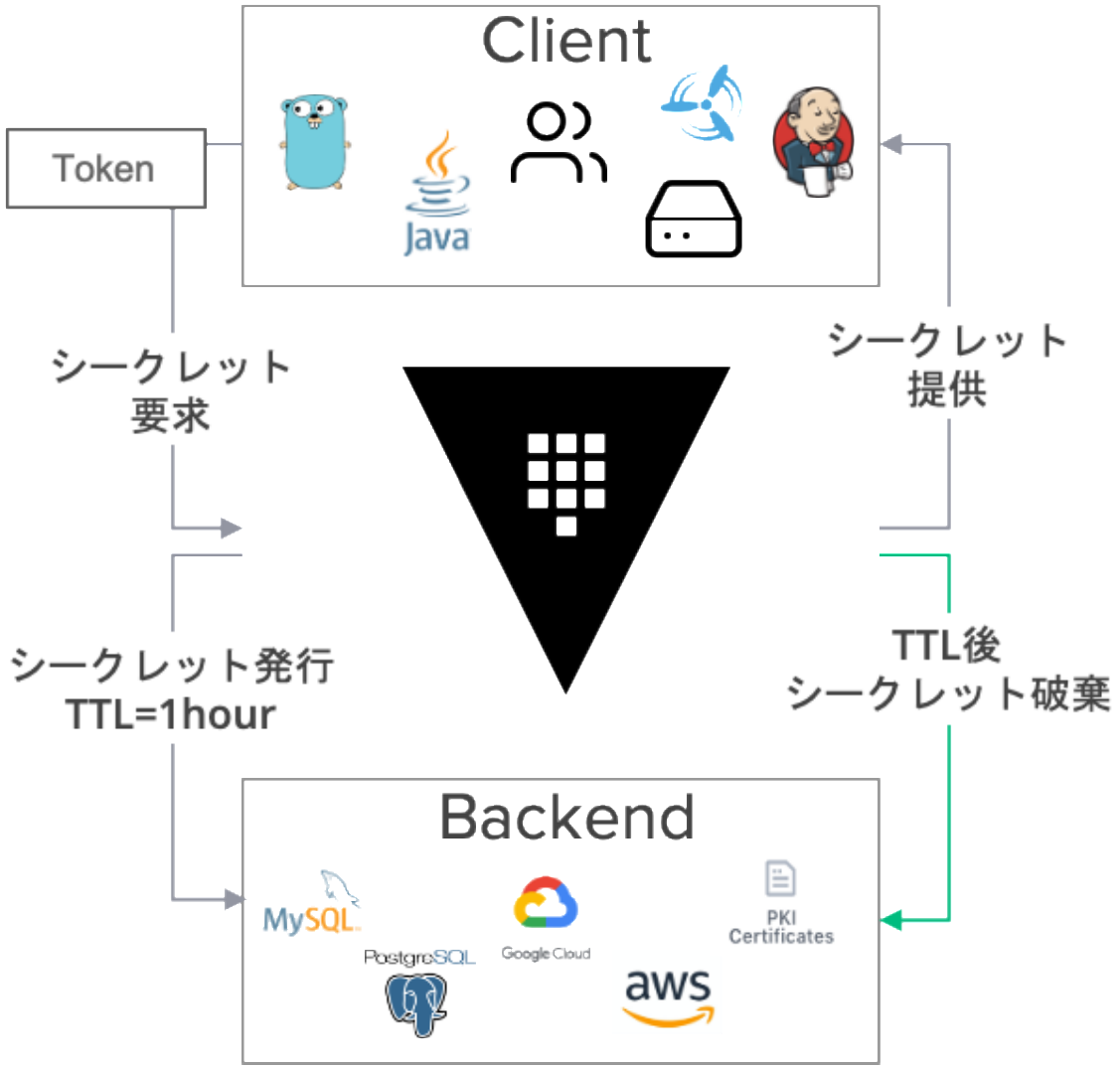
シークレットエンジンの種類と利用

シークレットエンジンを利用することで、認証されたクライアントがシークレット（何かにアクセスするために必要な「何か」）を使用する際の管理ワークフローが実装できます

- **動的シークレット（Vaultが差別化されている点）**
  - Public Cloud: AWS / Azure / GCP / Oracle / AliCloud
  - Middleware: Database / RabbitMQ / Nomad / Consul
  - Active Directory
  - Open LDAP
  - PKI (証明書)
  - SSH (ワンタイムパスワード / CA認証)
- **静的シークレット**
  - KV Secret Engine



Dynamic(動的)シークレットのワークフロー



Terminal

```
$ vault read database/creds/mysql-role
→ VaultがDatabaseのSQLを実行し、ユーザとパスワードを発行
$ vault read aws/creds/vpc-admin
→ VaultがAWS API実行し、IAMキーを発行
$ vault read ssh/role/otp
→ VaultがVM上でシェルを実行し、SSHパスワードを発行
$ vault write pki_intermediate/issue/kabuctl-dot-run ¥
common_name="blog.kabuctl.run"
→ Vaultが認証局となり、証明書を発行
```

Dynamic Secret：SSH　現状の運用

ID,Password認証

- 接続対象上にSSH用のユーザーを作成
- SSH接続時には作成したユーザーのID,Passwordを利用して接続

- ID,Passwordのみで接続できてしまうため最低でも複雑なパスワードによる運用が必要
- パスワードのローテーションなどはサーバー個別に実施する必要がある

公開鍵認証

- 接続元でキーペアを作成
- 接続先サーバーに作成した公開鍵を配置
- 接続時に配置した公開鍵と接続元にある秘密鍵をつかい認証する

- 接続先サーバーに接続する接続元機器を追加するたびに公開鍵の設定が必要

Dynamic Secret：SSH

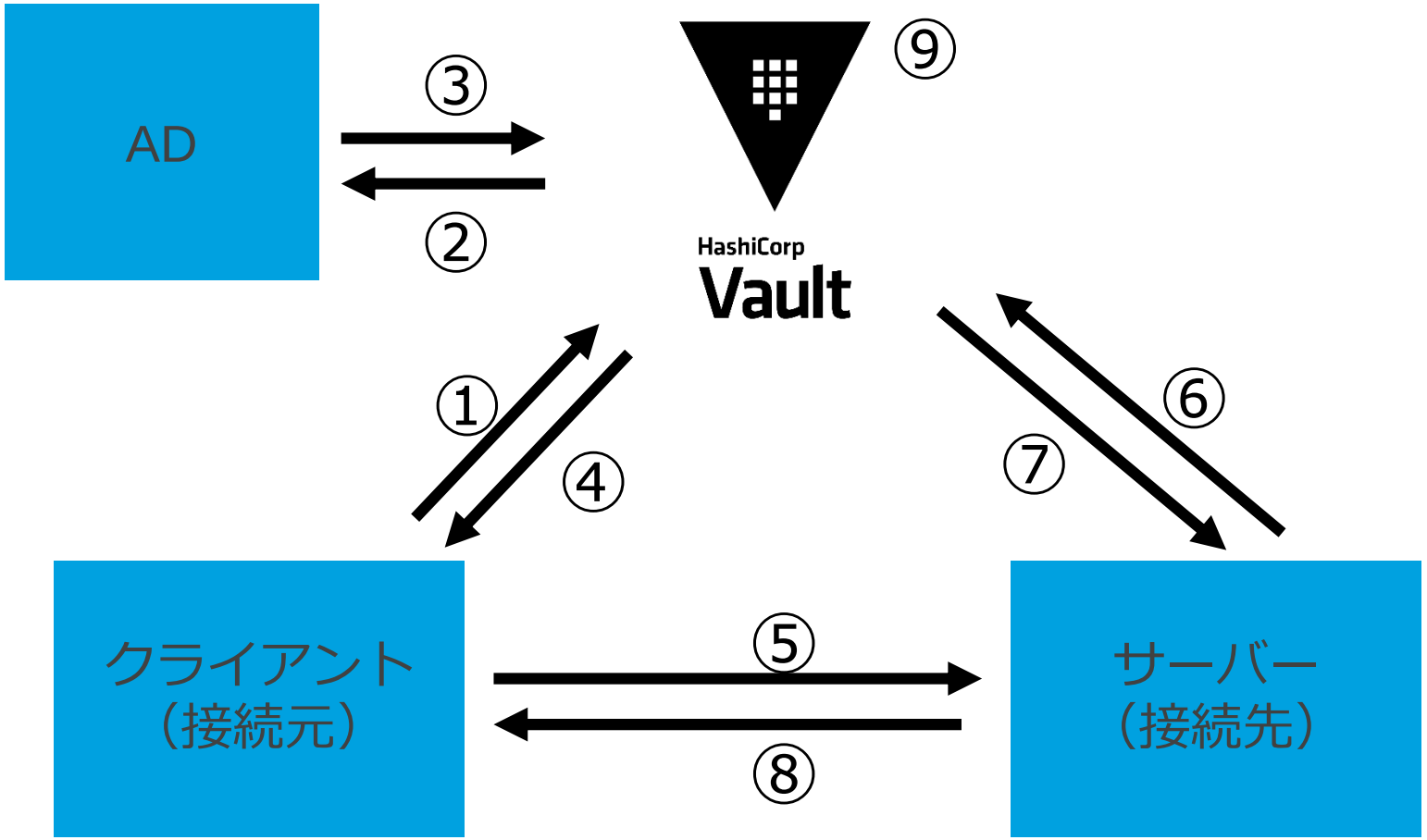
CA認証

- VaultサーバーをCAとして利用
- SSH接続対象にはCA公開鍵を登録
- クライアント(接続元)はキーペアを作成し、公開鍵をCA(Vaultサーバー)で署名

ワンタイムパスワード認証

- SSH接続対象のsshd.confの設定を変更し、チャレンジレスポンス認証とPAMを利用
- vault-ssh-helperを利用して、ワンタイムパスワードをVaultサーバーに対して認証
- ワンタイムパスワードはVaultサーバーで発行

Dynamic Secret：SSH ワンタイムパスワード



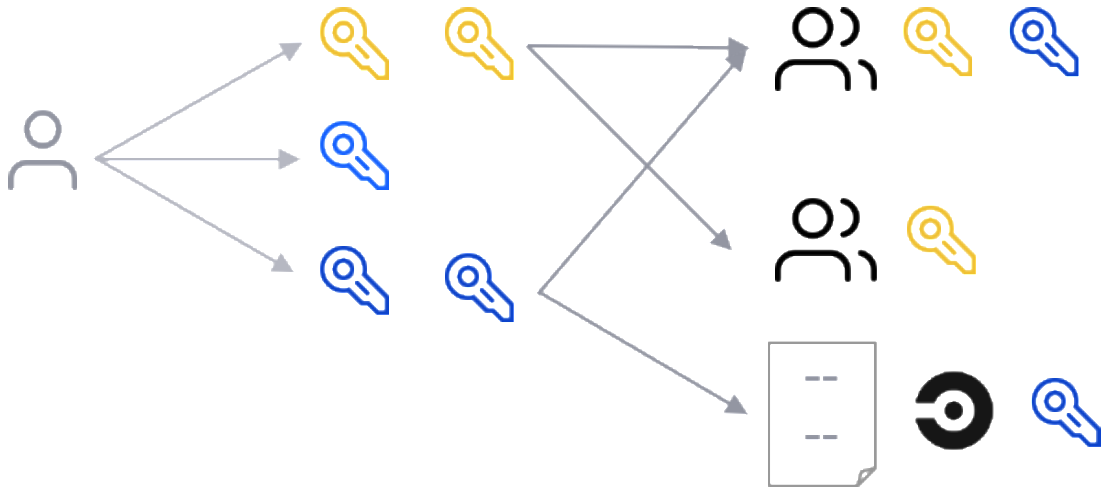
①認証＆OTPの発行依頼
②Vaultへのユーザー認証要求
③認証
④OTPの発行
⑤OTPを使ってログイン
⑥OTPをVaultに確認要求
⑦確認
⑧ログイン許可
⑨ログイン確認後OTPを消去

※OTP = ワンタイムパスワード

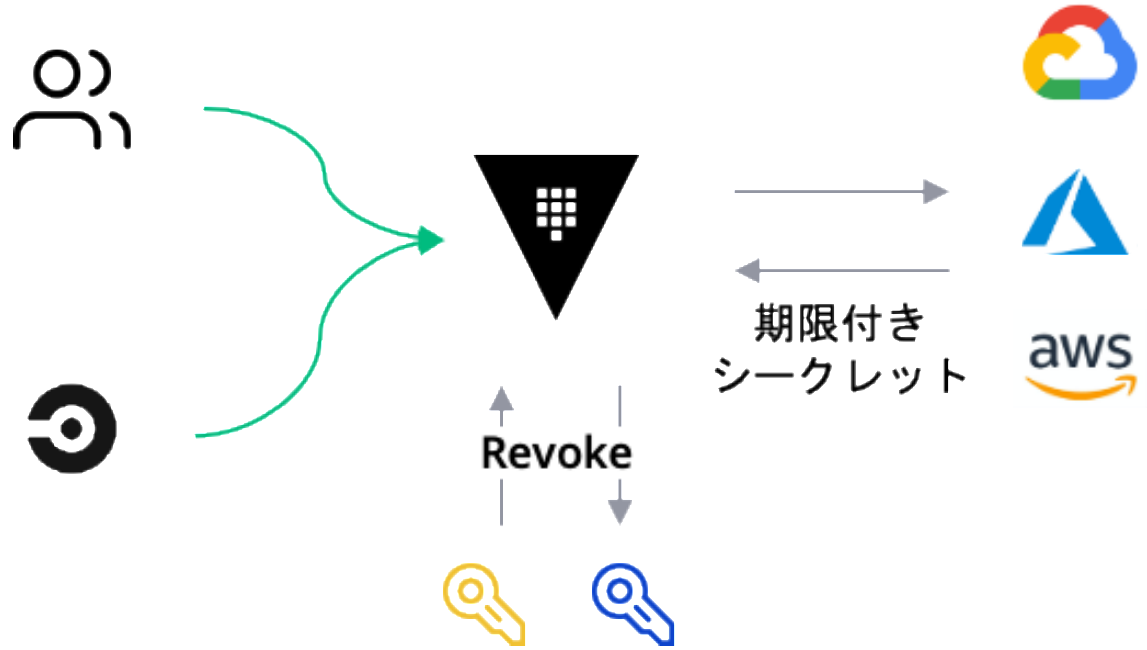
Dynamic Secret：Public Cloud

一般的な運用

- 手動でのキー発行
- 一度発行したキーを長期間利用
- かつ同じキーを使い回し
- アカウントが乱立しmanageabilityが低下
- 権限管理が複雑に
- 設定ファイルへの直記入



Vaultによるシークレット管理

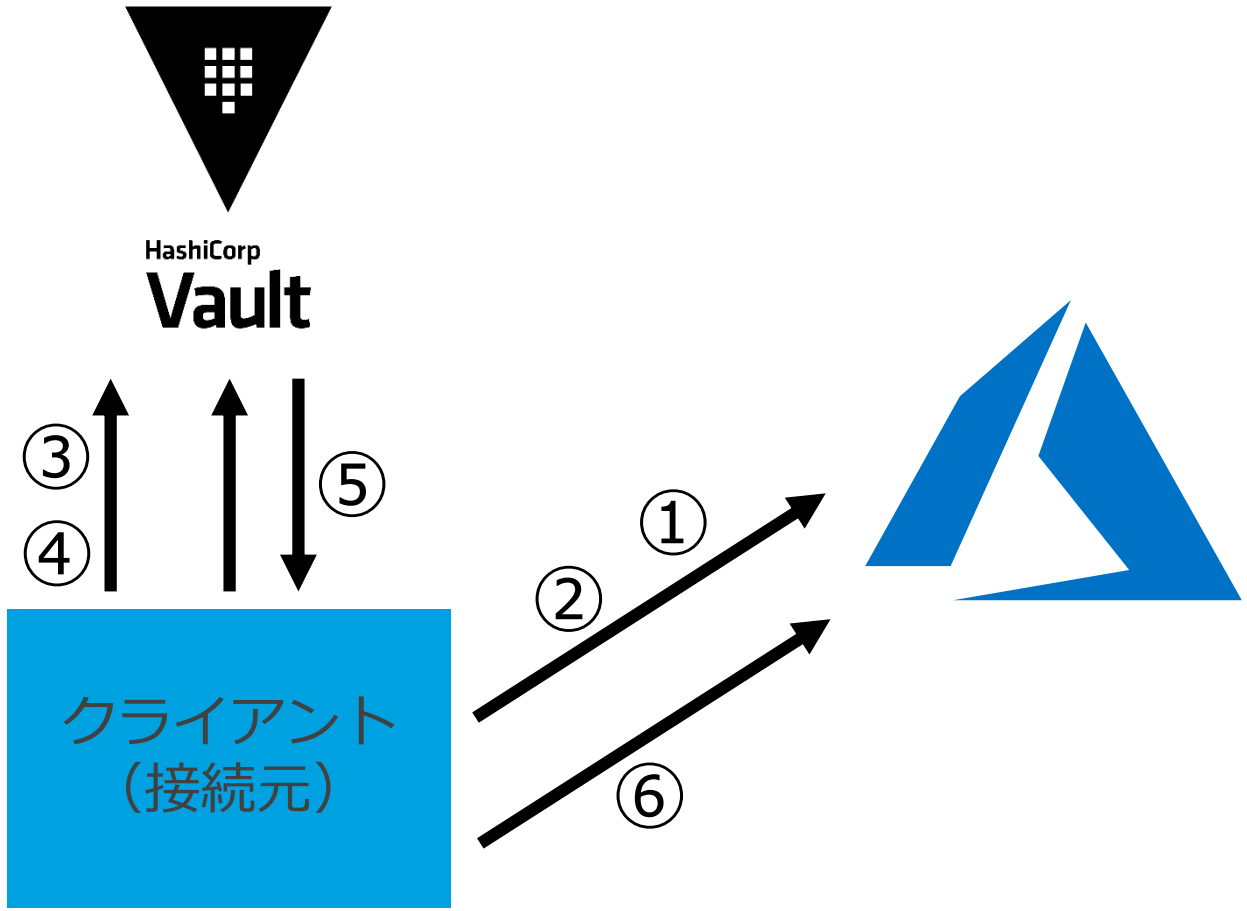


Dynamic Secret：Public Cloud の利点

クラウドシークレットキーを動的に発行

- 必要なキーを必要な時に生成し、自動で削除
- 常時新しいキーを利用可能に
- 設定ファイル等にシークレットの記述が不要でリスクを低減
- 発行するユーザやクライアントに対してクラウドへの権限を柔軟に設定可能
  - Principle of least privilege

Dynamic Secret：Public Cloud(Azure)

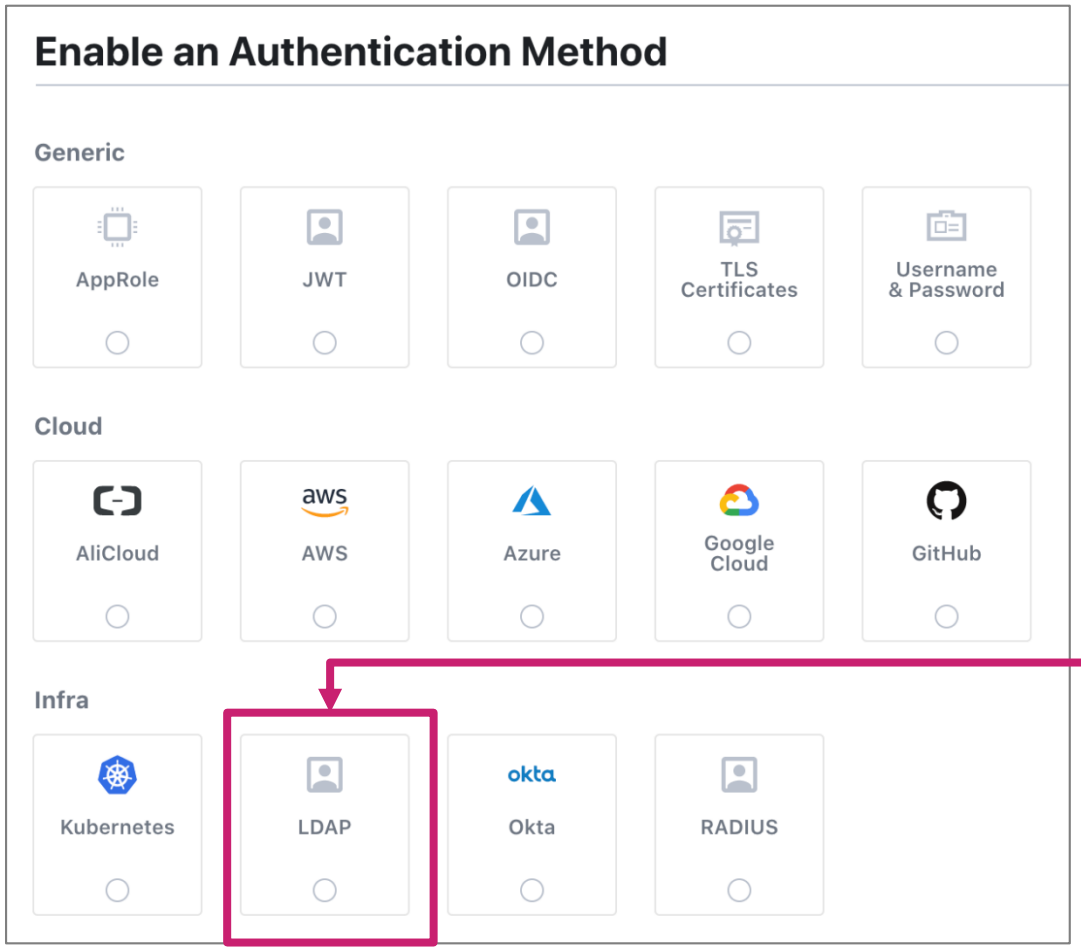


①連携に必要な設定の実行
②連携に必要な情報の入手
③Azure Secret Engineの有効化と設定
④Azureの権限をもつRoleの作成
⑤シークレットの発行
⑥シークレットを使ってログイン

連携に必要なAzureの情報
- サブスクリプションID
- ディレクトリ（テナント）ID
- アプリケーション（テナント）ID
- クライアントシークレット

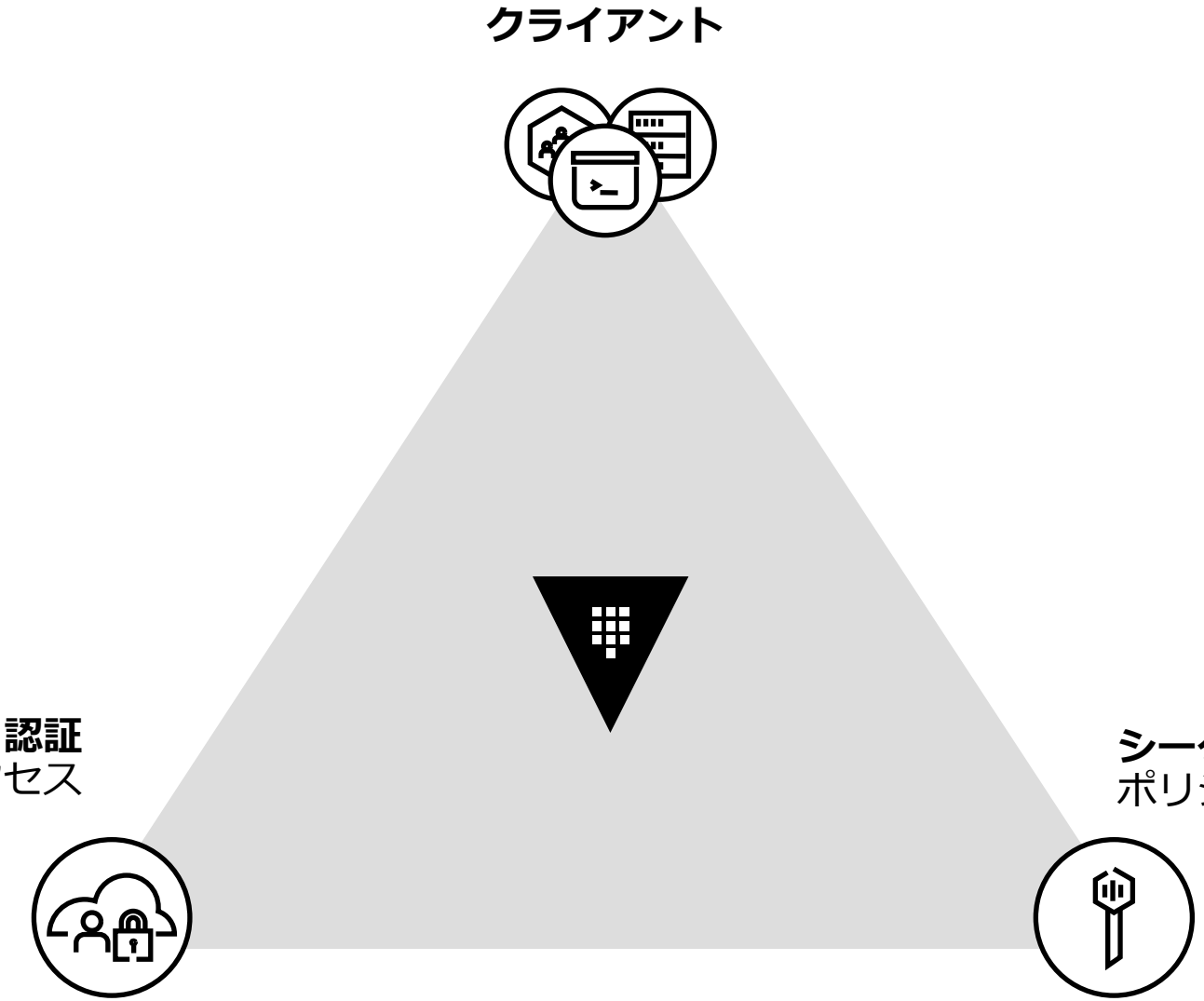
Vaultのユーザー管理

- ユーザー管理（認証)として複数のオプションがあります。



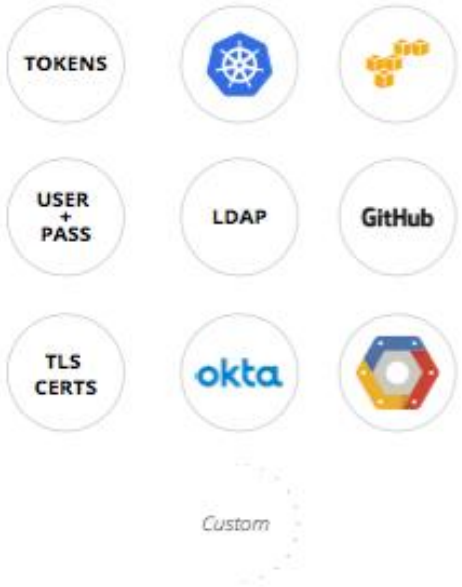
SSHのデモではLDAPと連携していました。

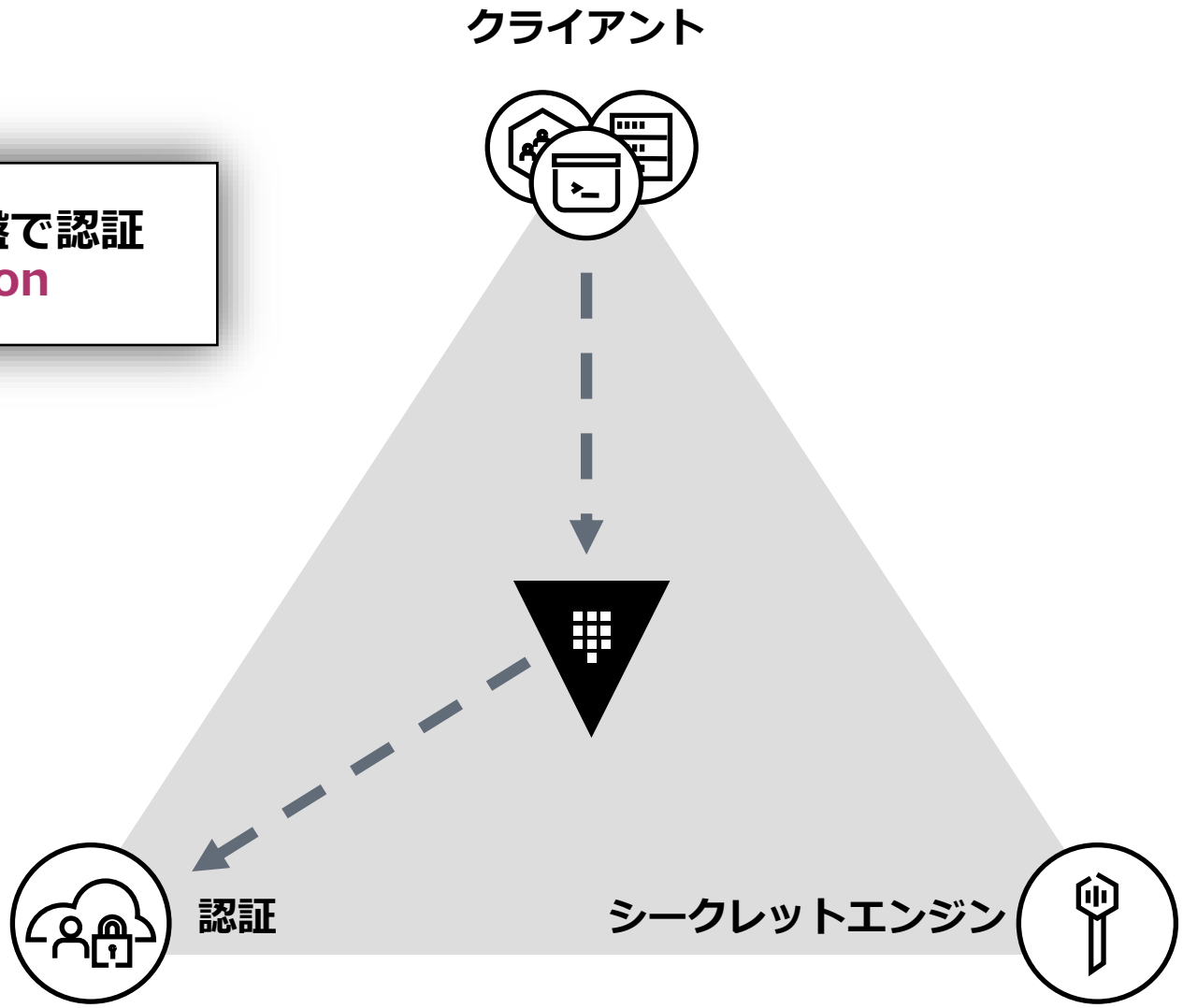
Vault を使ったシークレットの発行手順



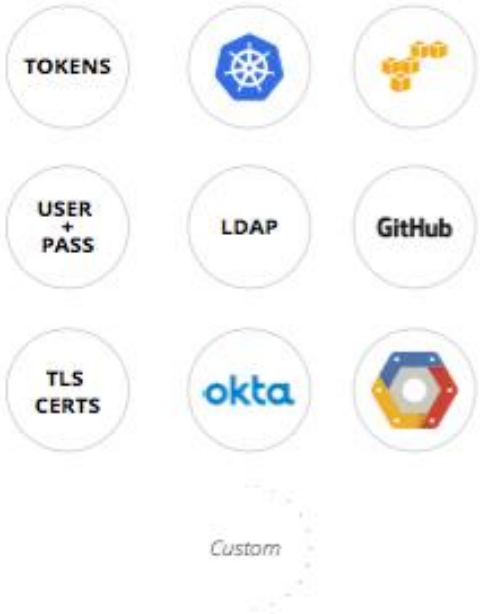
信頼できるIDPとの連携

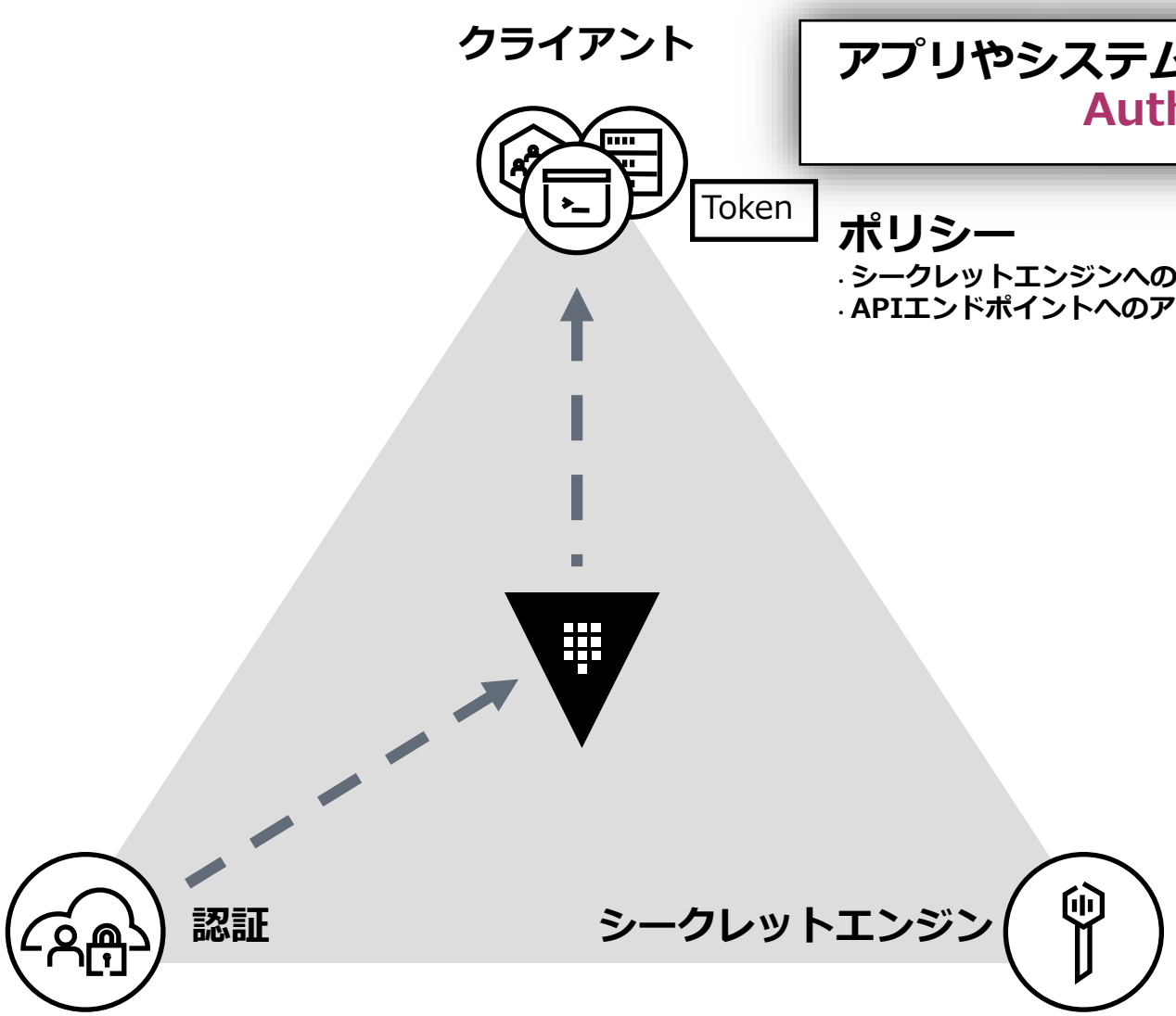
信頼できる認証基盤で認証
Authentication





Tokenの取得

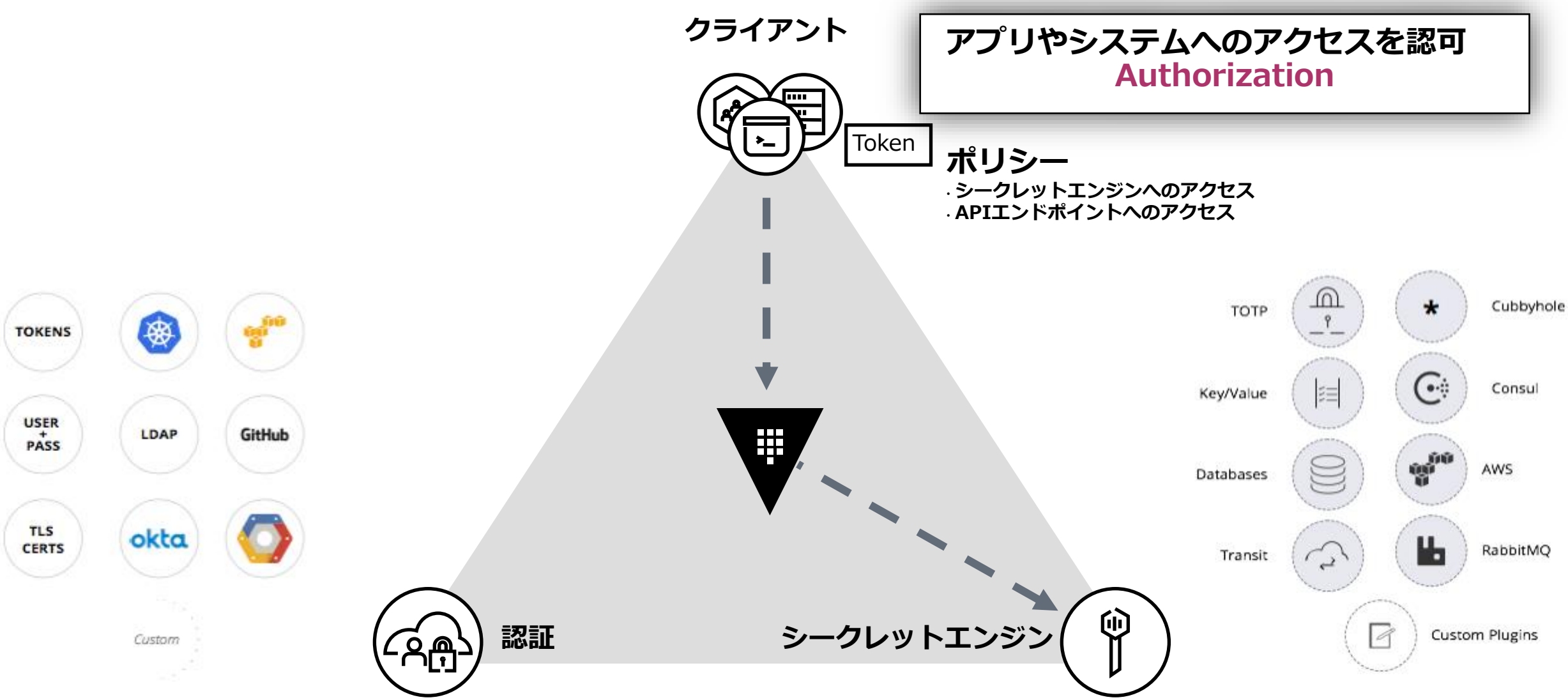




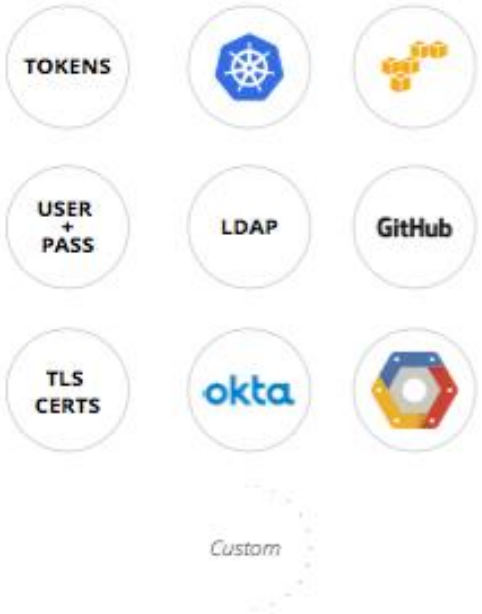
アプリやシステムへのアクセスを認可
Authorization

ポリシー
- シークレットエンジンへのアクセス
- APIエンドポイントへのアクセス

シークレットのリクエスト



シークレットの取得



クライアント

アプリやシステムへのアクセスを認可
Authorization

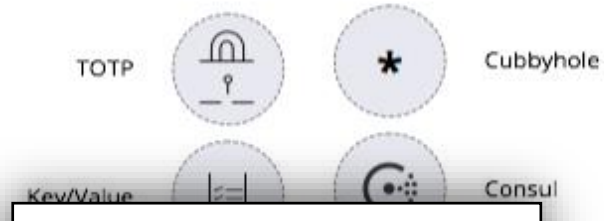
Token

ポリシー
- シークレットエンジンへのアクセス
- APIエンドポイントへのアクセス

シークレット

認証

シークレットエンジン



サービスの提供
- 動的シークレット
- 静的シークレット
- 証明書
- 暗号化サービス

Vault 導入 の Before & After

Before

- シークレットの使い回し、長期利用
- アプリの設定にシークレットを記入
- シークレットの乱立、管理が煩雑
- アクセスコントロールの設計が困難

つまり…

- **パスワードを個別に管理する必要あり**
- **同じパスワードだから漏れても追えない**
- **ひとつ漏れると全部突破される**
- **強すぎる権限など不適切なコントロール**

After

- シークレットのシングルレポジトリ
- Vaultから様々なシークレットを発行
- 期限(TTL)付与しシークレットの
  ライフサイクルをコントロール
- 細かな権限管理

つまり…

- **単一のシステムでシークレットを集中管理**
- **シークレットは必要な分を自動的に生成**
- **個別シークレットだから漏洩元がわかる**
- **漏れてもすぐ有効期限切れになる**
- **適切な権限だけを与えられる**

Vault Enterprise

Vault Enterprise

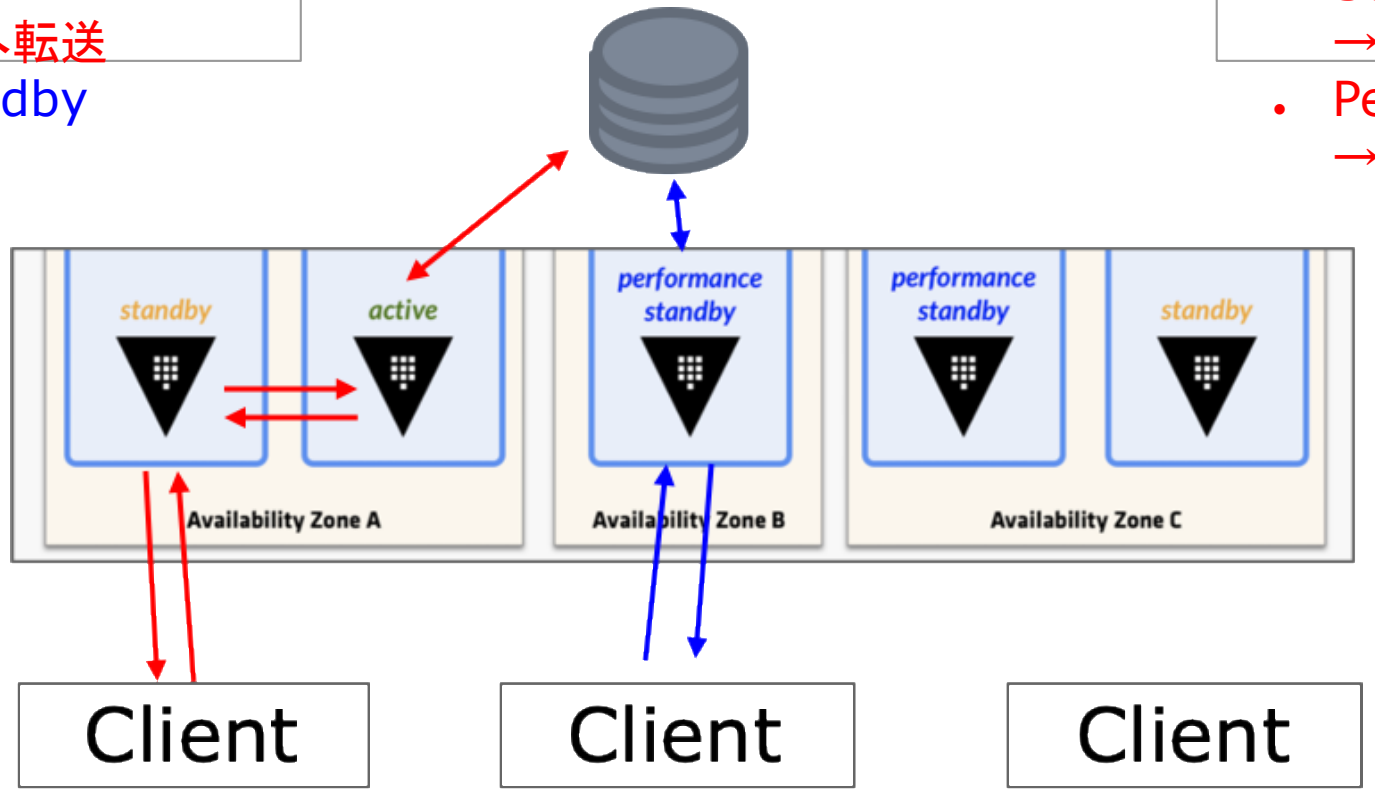
機能名	概要
Namespace	論理的なマルチテナンシー
DR Replication	Vaultクラスター間でトークン、シークレットやキーを含めたレプリケーションをし可用性を向上
Performance Replication	Vaultクラスター間でシークレットなどをレプリケーションし、複数クラスタでリードを処理しパフォーマンスを向上
Performance Standby	1クラスタ内で複数のリードノードを立てパフォーマンスを向上
Control Groups	Response Wrapping Tokenにアクセスする際に認証フローを入れセキュリティを向上
HSM Auto-unseal	Hardware Security Moduleによる自動unseal
Replication Filters	クラスタ間でレプリケーションするデータの条件を指定してフィルタリングをする
Policy as Code (Sentinel)	SentinelによるVault APIコール等に関するポリシーの設定
Multi Factor Authentication	Vaultへの多要素の認証
KMIP	Key Management Interoperability Protocol
HashiCorp Support	• Solutions Engineer, Technical Account Managerによるヘルスチェックや定例MTG • 24 * 365のサポート

Performance Standby

- 1クラスタ内で複数のリードオンリーのノードを立てパフォーマンスを向上

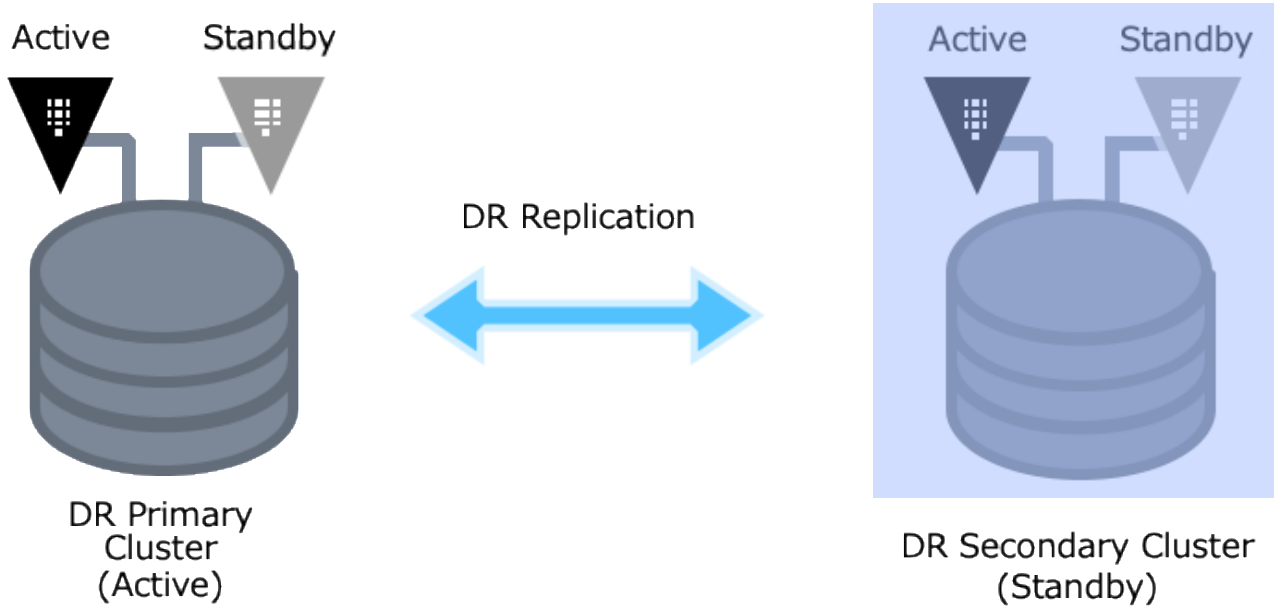
Read operation (Transit含む)
- Standby
  → Activeノードへ転送
- Performance standby
  → 自分で処理

Write operation
- Standby
  → Activeノードへ転送
- Performance standby
  → Activeノードへ転送



DR Replication

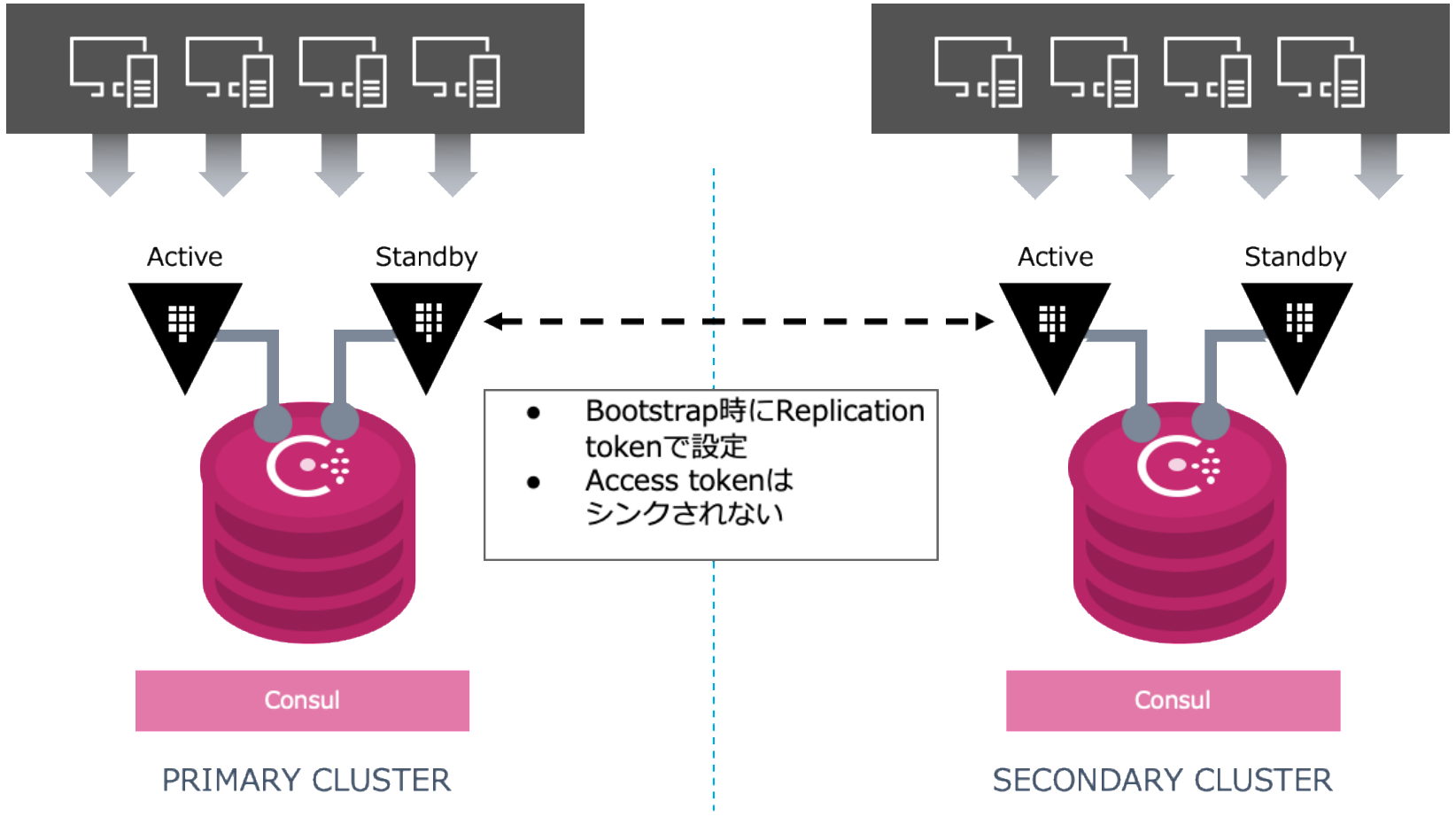
- Vaultクラスター間でトークン、シークレットやキーを含めたレプリケーションをし可用性を向上



- 全てのデータがシンクされる
- PrimaryへのPromotionは手動で行なう
- Vault 1.4でBatch DR tokenによりPromotion作業が非常に容易に

Performance Replication

- Vaultクラスター間でシークレットなどをレプリケーションし、複数クラスタでリードを処理しパフォーマンスを向上



まとめ

Vaultを導入すると・・・・

- シークレットを統合管理し、
- 必要な人に必要な時、必要な部分だけを提供
- だれがいつ使ったかの証跡も確認可能
- 認証を既存を連携することでユーザーの手間は変わらず
- クラウドへの対応もバッチリ！
- **Enterprise** を使うことで最重要な基盤となりうるVaultを
  安心・安全運用が可能！！

Networld