

クラウドのマルチテナンシー

オープン API で請求とチャージバックが簡単に

企業やクラウド サービス プロバイダーの多くは、高い拡張性を備えた自動化サービスをオンデマンドで提供するために、優れたクラウド アーキテクチャーを採用したいと考えています。リソース共有、迅速なスケーリング、自動化、セルフサービスといったクラウドのメリットと、高度なセキュリティ、制御性、柔軟性の両方のメリットを提供するのが、テナントのエンタープライズ クラウドです。また、クラウド アーキテクチャーにとって重要なのは、共有環境でアプリケーションの実行を行う各テナントを分離する、マルチテナンシー機能です。マルチテナント環境では、あるテナントが制御するリソースを他のテナントから物理的または論理的に分離することによってセキュリティを確保することになります。多くの場合、テナントの分離に加えて、テナント単位でのレポート作成やクォータ管理も重要になります。

クラウドにとってマルチテナンシーが重要な理由

高い俊敏性とスケーラビリティを実現: マルチテナンシーなら、新機能のロールアウトは、顧客が共有するインフラストラクチャー全体に 1 度行うだけで済みます。一方、顧客ごとに専用のハードウェアを用意している場合は、変更があるたびにすべての顧客環境で大規模なオーケストレーションを行うことが必要になります。

コスト効率が良い: コストを最適化するには、インフラストラクチャーの使用率を最大限に高める必要があります。マルチテナンシーでは、複数のテナントでインフラストラクチャーを共有できるため、コストを大幅に削減できます。ストレージのオーバープロビジョニングによって無駄が生じることもありません。

テナントのエンタープライズ クラウドを支える 4 つの柱



自律運用

エンド ユーザーは、一貫性や予測可能性を重視していますが、共有インフラストラクチャーでこれを実現するのは簡単なことではありません。多くの場合、複数のテナントのアプリケーションを LUN 間またはボリューム間で入れ替えるために、頻繁な介入が必要となります。また、細分化されたサービス レベルごとに、異なるサービス品質 (QoS) 要件を維持する必要もあります。しかし、テナントのエンタープライズ クラウドでは、以下のことが簡単に実現されます。

- ▶ VM レベルのパフォーマンスを分離する組み込み機能により、アプリケーションのパフォーマンスを保証 - 手動でのチューニングは必要ありません。
- ▶ 高負荷のワークロードが混在する場合でも、アプリケーション間の共有リソースの競争を回避 - ノイジー ネイバー (うるさい隣人) 問題は生じません。



分析

すべてのアプリケーションを完全に見える化し、インフラストラクチャー全体をリアルタイムに把握することで、ホスト、ネットワーク、ストレージ全体の問題をトラブルシューティングできます。テナントが実現するマルチテナント環境では、以下のことが可能になります。

- ▶ 監視、チャージバック/ショーバック、レポート生成を適切なレベルで実行できる
- ▶ 予測分析機能によって影響をモデル化することで、根拠があやふやな予測に頼ることなく、ストレージのオーバープロビジョニングを防止できる



自動化

これまで手動で行っていた作業を自動化することで、ストレージによるクラウドのボトルネックを解消できます。テナントによって以下のことが可能になります。

- ▶ REST API、PowerShell Toolkit、Python SDK、vRealize Orchestrator (vRO) プラグインを VM レベルで使用できる
- ▶ LUN の構築やチューニングを行うことなく、VM を瞬時にプロビジョニングできるほか、データ保護ポリシーや QoS ポリシーも自動的に適用される



セルフサービス

データセンターの IT セネナリストはストレージのフットプリントを管理し、事業部や DevOps のエンド ユーザーは独自のアプリケーションをすばやく簡単に管理できます。テナントのオープン API を使用することで、以下のことが可能になります。

- ▶ Slack などのメッセージング サービスを介してテナントとチャットできる
- ▶ エンド ユーザーがさまざまな操作を実行するためのオンライン セルフサービス ポータルを構築できる



ティントリはマルチテナント環境と非常に相性が良く、まるでサービスプロバイダー専用設計されたかのようです。

Whipcord、テクノロジー担当バイスプレジデント、Dan Hamilton 氏

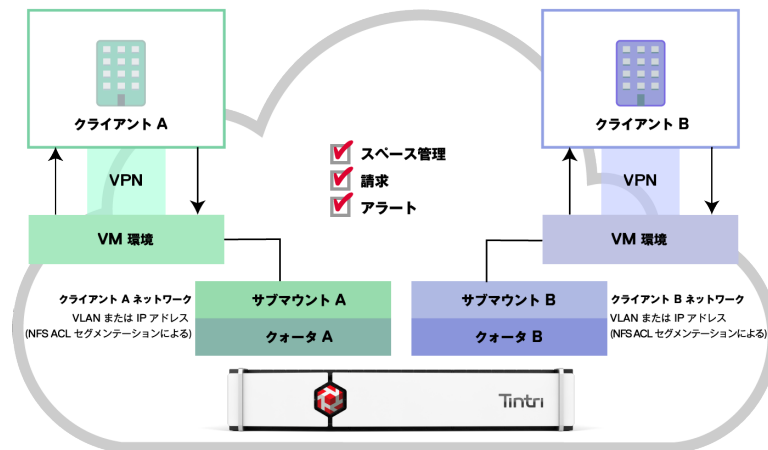


図 1. ティントリのマルチテナンシーとクォータ管理

マルチテナントに対するティントリのアプローチ

特に、複雑で扱いにくい従来のストレージを使用している場合、マルチテナント環境では、ストレージシステムに対して多大な需要が生じることがあります。ティントリの CONNECT アーキテクチャーでは、最も重要なアプリケーションに合わせることで、そうした複雑さを回避できます。

管理レイヤーにおけるマルチテナンシー

クラウドサービスプロバイダーの多くは、管理ソリューションを利用して IaaS や DaaS (Desktop as a Service) などのマルチテナントサービスを展開しています。また企業においても、社内の複数のグループに IT as a Service を提供している場合があります。よく使われるのは、IaaS では VMware vCloud Director や OpenStack、DaaS では VMware Horizon などのツールです。これらの管理パッケージでは、マルチテナント環境向けのフレームワークとツールが提供されますが、一方、ティントリは特定のテナントの VM 情報を提示することで、レポート作成やチャージバックを簡単に行えるようにします。

ホスト型プライベートクラウドとクォータ管理

プライベートクラウドの展開方法によっては、テナントがインフラストラクチャーにアクセスできる場合があります。専用インフラストラクチャーはコスト効率が悪くないため、クラウドサービスプロバイダーや企業は、複数のプライベートクラウドを共有インフラストラクチャーに展開し、マルチテナンシー機能を使用して各テナントを分離する必要があります。

一方、ティントリは、ホスト型プライベートクラウド環境をサポートしているため、テナントの安全な分離、データ暗号化、サービス保証に加えて、請求、チャージバック、カスタマイズに使用できる VM 単位の分析が可能です (図 1 参照)。

- ▶ **テナントの分離:** RBAC によるアクセス制御や VLAN とサブマウントの組み合わせにより、各テナントはそれぞれの専用サブマウントのみを表示、アクセスできます。サブマウントには容量クォータポリシーが適用されるため、各クライアントが割り当てられた容量を超過することはできません。また、サブマウントに定義したクォータについて、カスタムのしきい値に到達した場合に容量のアラートを表示するように構成することもできます。
- ▶ **データ暗号化:** Tintri SecureVM では、容量やパフォーマンスに影響を与えずに、保存データを暗号化します。
- ▶ **サービス保証:** ワークロードが混在する場合でも、各 VM のパフォーマンスが保証されます。ティントリの VM 単位の QoS によって、CSP は SLA の複数のパフォーマンスレベルを提供することができます。クォータを設定することで、クライアントによる容量の使用状況を簡単に把握、制御できます。
- ▶ **VM 単位の分析:** 詳細な VM 単位のデータが提供され、REST API を使用して取得することができます。この分析データは、請求、チャージバック、トラブルシューティングなどに活用できます。