L-024

IT/NW 統合アクセス制御方式の OpenStack 環境への適用
Integrated Access Control Policy Management for OpenStack

森田 陽一郎†
Yoichiro MORITA

1. はじめに

近年，一般に IaaS (Infrastructure as a Service) などと呼ばれるクラウドコンピューティング基盤の提供サービスや，その基盤を構築・運用するための管理ソフトウェアが急速に普及している． このような環境を，業務システムの基盤として複数の組織や部門で共用するには，特定の組織・部門や，その中の特定の役割を持つユーザにのみ，特定の業務システムの稼働するマシンやネットワークを操作できるようセキュリティ設定を行う機能が必要である．

筆者らは，適切な権限を適切なユーザに限定して割り当てるセキュリティ設定運用の徹底を目的として，IT/NW 統合アクセス制御方式を研究開発している[1]． 本方式は，ユーザの持つロール (役割) に基づくアクセス制御ポリシーによって，IT リソースやネットワークリソースに対するアクセス制御設定の一元化と一括設定の自動化を実現するものである (図 1).

本稿では，クラウドコンピューティング基盤を構築・運用するための代表的なオープンソースソフトウェアである OpenStack を対象として，IT/NW 統合アクセス制御方式によるアクセス制御設定の一元管理を実現する方法について説明する． 特に，本方式でポリシー管理を担う，統合アクセス制御情報管理 (IAM; Integrated Access Control Manager) との連携に際して必要となる追加機能等について述べる．

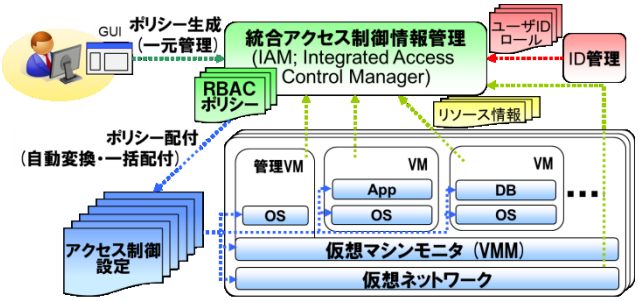

図 1: IT/NW 統合アクセス制御方式

2. IT/NW 統合アクセス制御方式の適用方針

2.1 クラウドコンピューティング基盤

IT/NW 統合アクセス制御方式による管理の対象となる，クラウドコンピューティング基盤の機能について述べる．

IaaS などのクラウドコンピューティング基盤の提供サービスは，主に仮想マシン (VM) を管理する仮想マシンモニタ (VMM) の機能を対象としており，利用者に対しては，Web フォームや Web API を介した管理機能を提供している． VMM の持つ管理機能としては，例えば，VM およびその構成の作成/削除，VM の起動/終了などを扱う． また，VM 間を接続するための仮想ネットワークの管理機能を提供している場合もある． 仮想ネットワークの管理機能としては，例えば，ネットワークセグメントおよびその設定の作成/削除，ネットワークセグメントと VM の持つ仮想 NIC (Network Interface Card) との紐付けや設定の作成/削除などを扱う．

このようなクラウドコンピューティング基盤を構築・運用するための代表的なソフトウェアの 1 つとして OpenStack がある． 商用・非商用を問わず急速に普及が進み，代表的な機器・ソフトウェアの多くが管理・連携可能となりつつある． VM や仮想ネットワークなどの管理機能を，幾つかの機能モジュールに分割して構成しているため，提供したいサービスの範囲に合わせた機能モジュールの選択や組み合わせが可能である． オープンソースのため，機能の分析や追加も容易である．

2.2 要件定義

クラウドコンピューティング基盤に IT/NW 統合アクセス制御方式を適用する際に求められる要件について述べる．

① ロールベースアクセス制御 (RBAC; Role Based Access Control) による一元管理

ユーザの権限に関し，ロール単位のアクセス制御ポリシーによって，一元的に管理が可能であることが望ましい．企業などの組織や部門のセキュリティ基準や業務ルールにおいては，職務分掌など，ユーザの役割に応じた適切な権限の付与と制限を行うよう統制が求められるため，RBAC による一元管理は その徹底に有用である．

② アクセス権の 3 つ組相当の粒度でのアクセス制御

アクセス権は，一般に，{サブジェクト，リソース，アクション} の 3 つ組で構成されるため，同等の粒度でのアクセス制御設定を可能とする必要がある． なお RBAC を用いる場合は，ユーザがサブジェクト，権限 (パーミッション) が {リソース，アクション} の組，ロールが権限の集合にあたる． アクセス制御設定としては，ユーザとロールとの関係と，ロールと権限集合との関係とを分けて指定し，これらを併せて解釈することで，3 つ組相当のアクセス制御設定を可能とする必要がある．

③ リソースのグループ化と名前付け

ロールと {リソース，アクション} の組との関係をアクセス制御設定として記載する際，VM や NIC など個々のリソースの持つ機械的な ID をすべて把握して列挙するのは手間がかかる． そのため，個々のリソースや その集合について，別名やグループとして意味を持つ分かりやすい名称を付けることで抽象化し，アクセス制御設定でのリソース指定に使用できることが望ましい．

④ 多種類のリソースやアクションへの対応

クラウドコンピューティング基盤におけるアクセス制御では，VM や NIC など多種類のリソースを扱う． VM というリソースに対しては起動や終了，NIC というリソースに対してはネットワーク設定など，リソースに対応するアクション (操作) も異なる． これら多種類のリソースやアクションに対してアクセス制御を実現する必要がある．

⑤ 将来のバージョンに対して適用容易

基盤を構築・運用するソフトウェアは，今後のニーズの拡がりに応じて頻繁なバージョンアップが予想される． 本

† 日本電気(株) クラウドシステム研究所
Cloud System Research Laboratories, NEC Corporation

方式を適用するにあたっては，新バージョンでも同様の適用方法が使用できることが望ましい．

3. 実装方法

3.1 OpenStack のアクセス制御機能の概要

OpenStack の持つアクセス制御機能について述べる．

OpenStack を構成する各機能モジュールは，それぞれの機能を提供するための API (Application Programming Interface) を持ち，API を介して，ユーザからのサービス要求の受付けや他の機能モジュールとの連携を実現する．OpenStack のアクセス制御機能としては，Keystone と呼ばれる機能モジュールが中心となって各機能モジュールが連携し，アクセス制御を実現している． Keystone は，ユーザ認証とトークン発行，トークン照会の機能を持ち，ユーザ ID・パスワードや，ユーザとロールとの関係，発行済みトークンの情報などを管理している． 各機能モジュールは，サービス実行，権限確認の機能を持ち，ロールと権限との関係などを，各機能モジュールが持つ policy.json というファイルで管理している．

OpenStack では，上記の機能を連携させることにより，以下の手順で，簡単な RBAC を実現している．

i. ユーザは，Keystone に対し，ユーザの認証を要求し，トークンを取得．
ii. ユーザは，機能モジュールに対し，トークンを添えてサービスの実行を要求．
iii. 機能モジュールは，Keystone に対し，トークンの照会を要求し，正規トークンであるとの結果を取得し，ロールなどユーザの情報も取得．
iv. 機能モジュールは，ユーザの情報と policy.json とを参照して，サービス実行に必要な権限が そのユーザに付与されているかを，ロールなどに基づいて確認し，権限有りとの結果を取得．
v. 機能モジュールは，ユーザの要求したサービスを実行．

3.2 OpenStack のアクセス制御機能の分析

3.1 節の機能構成を分析し，OpenStack のアクセス制御機能に対して IT/NW 統合アクセス制御方式を適用するために必要となる拡張について，以下の通り整理した．

① アクセス制御ポリシーの一元管理機能の追加

Keystone で一元管理されているのは，ユーザとロールとの関係のみで，ロールと権限との関係の管理は，個々の機能モジュールに分散しており，一元管理されていない．

② リソースの指定機能の追加

policy.json に権限の情報を記載する上で，ロールとアクションは指定できるが，リソースは指定できず，アクセス権の 3 つ組相当のアクセス制御を行うことができない．

③ リソースの抽象化機能の追加

リソースが指定できないため，グループ化や名前付けによる抽象化ができない．

④ リソースとアクションとの関係の管理機能の追加

アクションは指定できるが，リソースは指定できないため，その間の関係が管理されていない．

⑤ ①〜④ を出来る限り無改造で実現

現バージョンに無い機能が多いため，①〜④を実現するにあたって改造が必要である． しかし，機能追加のために OpenStack を大規模に改造すると，将来のバージョンとの差異が大きくなりすぎ，新バージョンでは現バージョンと同様の実装方法が使用できなくなる可能性が高い．

3.3 IT/NW 統合アクセス制御方式の適用

3.1 節の機能の詳細を分析し，以下の拡張方法を導出した． これらの拡張により，OpenStack への IT/NW 統合アクセス制御方式の適用を可能とし，2.2 節の要件を満たした．

① Keystone のロール情報と連携し，policy.json を IAM (Integrated Access Control Manager) で一元管理

IT/NW 統合アクセス制御方式により仮想マシンと仮想ネットワークとを統合管理する IAM を用いて，Keystone の管理するロールに対応するアクセス制御ポリシーを管理し，ポリシーから個々の policy.json 向けのアクセス制御設定を生成して，IAM から各機能モジュールに配付することで，システム全体のアクセス制御設定の一元管理を実現した．

② アクセス制御設定の解釈を行う共通コードの小改造と，簡単な判定用コードの外部追加

policy.json の解釈を扱う共通コード common.policy.Brain において，リソースを判定するための関数呼出しのみを追加し，呼び出される関数のコードを外部に追加した． この関数は，リソースに関する判定結果が許可の場合は，関数を追加しない状態と同様に何もせず，拒否の場合は，既存の関数と同じエラー (例外) を投げる． これにより，既存の判定コード自体には手を加えることなく，簡単な判定コード追加のみで，判定機能の拡張が可能である．

③ 各機能モジュールの DB にテーブルを追加し，リソースのグループなどの属性を外部追加

VM などのリソースの情報を管理する DB に関して，機能モジュールが本来持つテーブルには存在しない別名やグループなどの属性を，別テーブルに定義し，リレーショナル DB の機能を用いて本来のテーブルと紐付けた． これにより，本来のテーブルを変更することなく，任意の属性が追加可能である．

④ IAM において，リソースタイプとアクションとを紐付け

リソースを種類別に分類するために，リソースやリソースグループに対してリソースタイプ属性を紐付けた． リソースタイプには例えば VM や NIC などがある． その上で，各リソースタイプに対応するアクションを IAM で管理し，IAM でのアクセス制御ポリシーの編集時に，編集者が選択したリソースグループ等に合わせて適切なアクションのみを提示する． これにより，policy.json 向けに生成されるアクセス制御設定を，適切なリソースとアクションの組のみに限定することが可能である．

⑤ 本来の情報や機能は無改造や小改造で抑え，単純な情報や機能を外部追加

①〜④のように，OpenStack 本来の情報や機能は出来るだけ変更せず，その外部に汎用性の高い単純な情報や機能を複数追加して，IAM との接点とした． これにより，OpenStack の新バージョンでコードに変更があった場合でも，同様の方法で情報や機能を挿し込むだけでよいため，旧バージョンと新バージョンとの間で高い互換性を持つ．

4. おわりに

OpenStack を用いたクラウドコンピューティング基盤に，最小限の改造で IT/NW 統合アクセス制御方式を適用し，ロールに基づくアクセス制御設定の一元管理を実現した．

参考文献

[1] 森田, 山形, 佐々木, 中江, "IT/NW 統合アクセス制御方式におけるポリシー生成・配付機能", SCIS2012, Jan 2012.