

クラウド基盤構築演習

第二部: Eucalyptusによるクラウド基盤構築

第9回: Eucalyptusの基礎知識

ver1.2 2012/06/23



目次

- Eucalyptusの概要
- Eucalyptusの機能
- Eucalyptusのコンポーネント
- 動作に必要な環境
- Eucalyptusのディレクトリ構造
- バージョン3.1での構成例



本講で学ぶこと、実施すること -1-

■ Eucalyptusの概要を知る

- Eucalyptusが作られた経緯や歴史を知ること、Eucalyptusがどのような段階を経て今の機能を有しているかを理解することができます

■ Eucalyptusの機能を知る

- Eucalyptusの各機能を知ること、Eucalyptusでどのようなことができるかを理解することができます



本講で学ぶこと、実施すること -2-

- Eucalyptusの各コンポーネントを知る
 - どのようなコンポーネントによってEucalyptusが構成されているかを知ることができます
 - 各コンポーネントがどのような機能を提供しているかを理解することができます
- Eucalyptusのディレクトリ構造を知る
 - 各コンポーネントがどこにどのようなファイルやディレクトリを有しているかを知ることができます
- 動作に必要な環境を知る
 - EucalyptusによるIaaS環境を構築する際にどのようなハードウェアやネットワーク環境を用意すべきかを知ることができます



EUCALYPTUSの概要



Eucalyptusとは？

- Amazon EC2/S3互換のプライベートクラウド環境 (IaaS環境)を構築できるOSS
- 名前の由来は“**E**lastic **U**tility **C**omputing **A**rchitecture for **L**inking **Y**our **P**rograms **T**o **U**seful **S**ystem”の略で日本語に無理矢理訳すと「プログラムを実用的なシステムにつなぐための伸縮可能なユーティリティコンピューティングの仕組み」
- カリフォルニア大学サンタバーバラ校のコンピュータサイエンス学科の研究プロジェクトとして開発がスタートし、現在はEucalyptus Systems, Incが開発
- ライセンスはGPLv3
 - コミットする場合はCLAを結ぶ

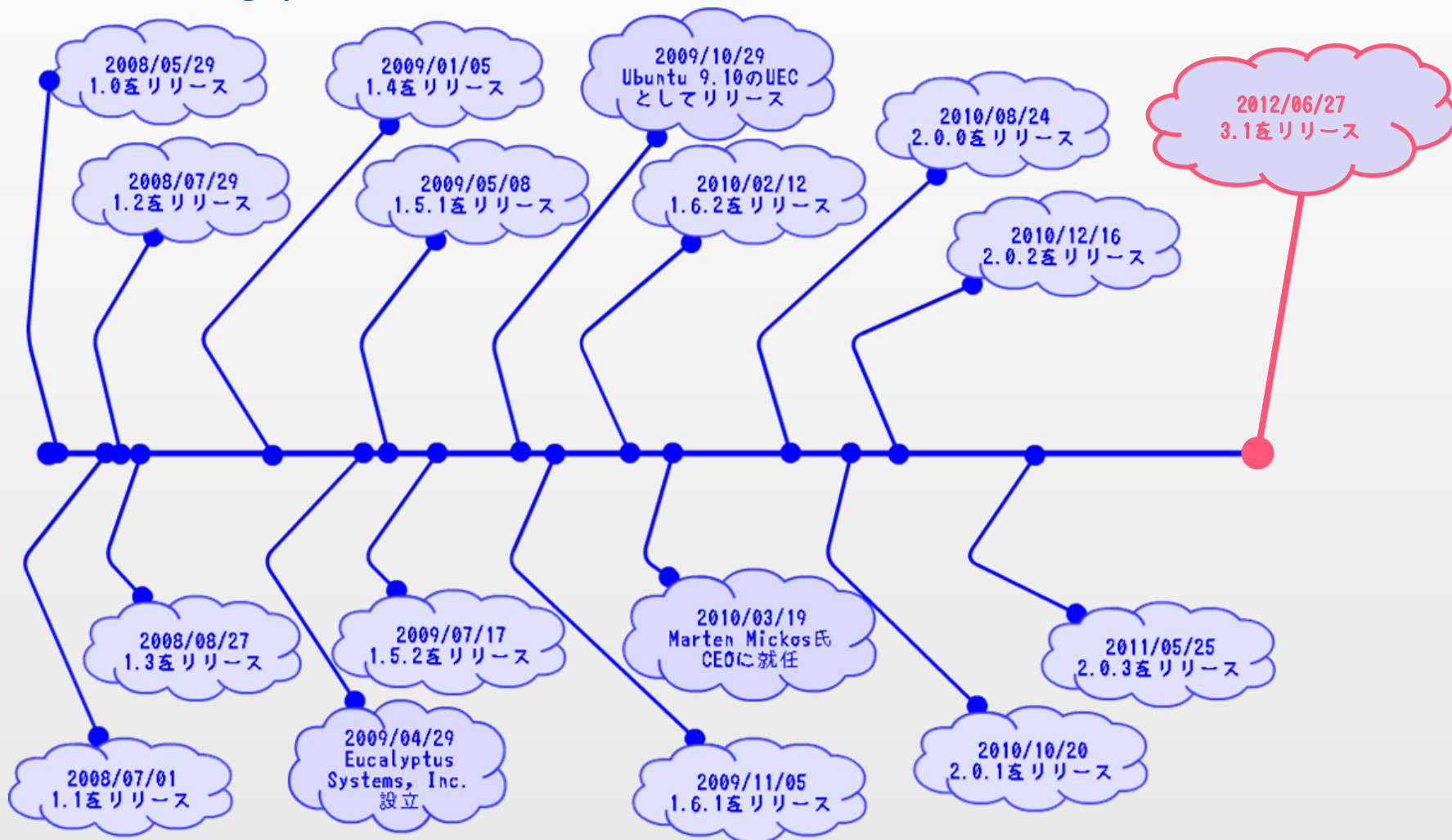


どのぐらいAmazon EC2/S3互換なのか？

- “EC2/S3 互換”ではあるが、プライベートクラウドでは必須ではない機能などは実装していない
 - 課金やモニタリングなど
- 基本的には2009-04-04時点のAmazon EC2 APIに準拠している
 - v3.1ではAPIによって2010-08-31とか2011-01-01もある
- S3 APIに関するWSDLは2006-03-01のAPIのようだが、実装についてはAPI毎に異なる様子



Eucalyptusの歴史





Eucalyptusの主な歴史 ～1～

■ v1.1

- Web管理画面を大幅に改良
- 各コンポーネント間の通信にWS-securityを使用
- Query APIを実装

■ v1.2

- インスタンス起動時間の短縮のためにイメージキャッシュ機能を追加
- インスタンスに関するネットワーク設定をeucalyptus.confに追加

■ v.1.3

- ec2tools (ec2 api tools, ec2 ami tools)の最新版に対応

■ v1.4

- ElasticIP機能を実装
- S3互換機能のWalrusを実装
- メタデータを実装



Eucalyptusの主な歴史 ～2～

■ v1.5系

- EBS機能を実装
- ネットワークモードにMANAGED-NOVLANを追加
- RightScaleに対応
- ハイパーバイザーにKVMが利用可能
- Ubuntu用のバイナリパッケージ



Eucalyptusの主な歴史 ～4～

■ v1.6系

- マルチクラスタを実装
- 各コンポーネントを別ホストに配置可能
- 運用監視ツールとしてnagiosやgangliaを利用するための簡易スクリプトを提供
- WebUIにテーマ機能追加（ビルド時に指定）
 - 実際はUEC向けの拡張であり、そのためのドキュメント類は未整備
- DB接続をover SSL化
- 何気に頑張ればHA化も可能な感じに



Eucalyptusの主な歴史 ～5～

■ v2.0系

- iSCSI対応
- KVMのVirtioに対応
- S3のバージョニング機能に対応
- Walrus内のオブジェクト名(ファイル名)をハッシュで保持するように変更
- 管理用コマンドの増加
 - ビルド時にハードコーディングされていた内部パラメータが変更可能に
- 商用版ではハイパーバイザーにVMware (ESXi)が利用可能



Eucalyptusの主な歴史 ～6～

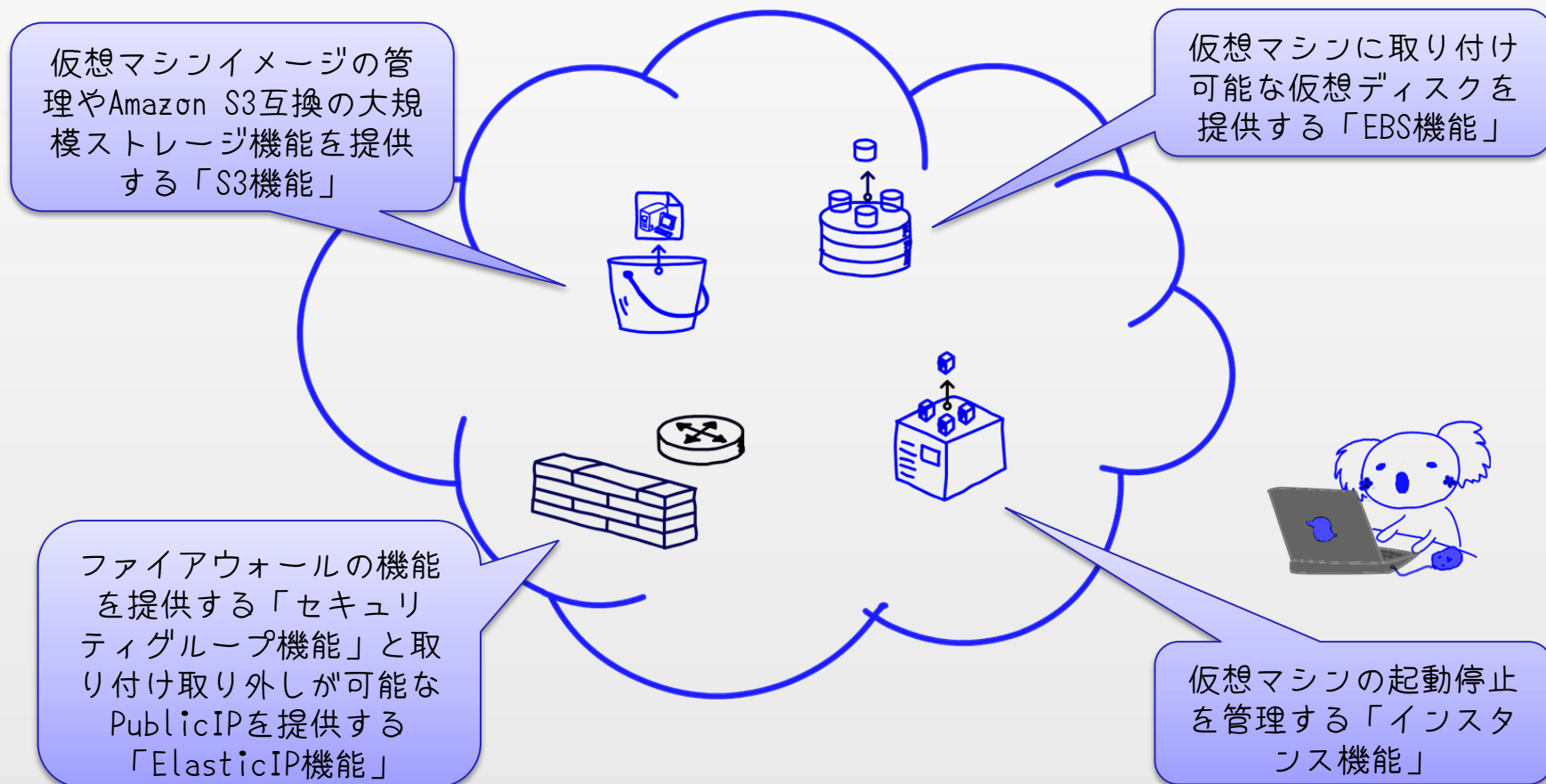
■ v3.x 系

- 高可用性 (HA構成)
- IAM機能を実装
- AD/LDAP連携
- Windowsゲスト対応
- EBS起動を実装
- オープンソース版でもハイパーバイザーに
VMware (ESXi)が利用可能
- SAN/NAS対応



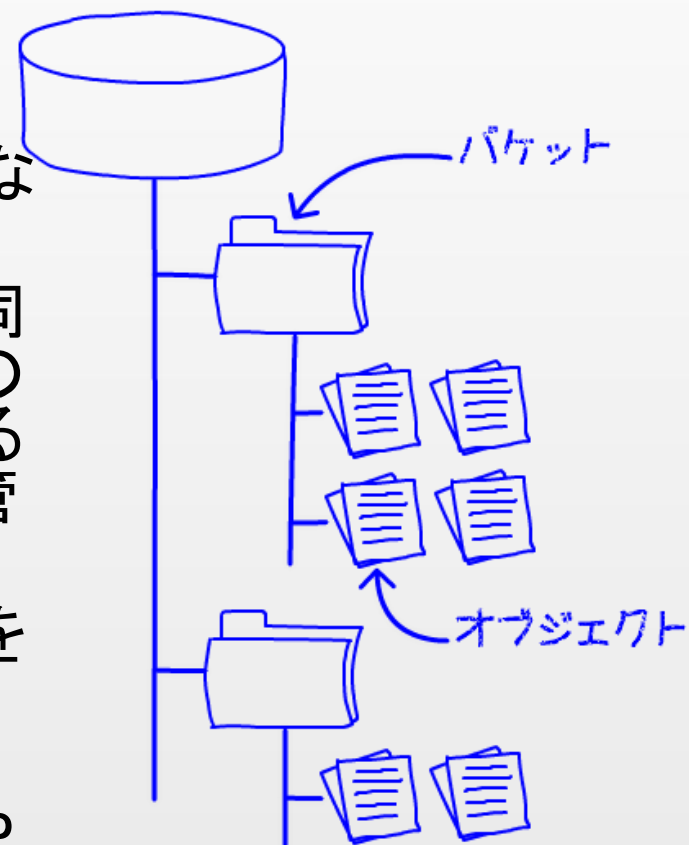
EUCALYPTUSの機能

Eucalyptusの主な機能



S3機能 -1-

- EC2の機能とは別の機能だが、EC2の機能を利用するためには欠かせない機能
- 基本的にはOSのファイルシステム同様、バケットと呼ばれるディレクトリのようなものと、オブジェクトと呼ばれるファイルのようなものによって構成管理が行なわれる
- Eucalyptusでは主にマシンイメージを管理・提供するために実装されている
- S3機能はAmazon S3と同様にSOAPおよびRESTによるAPIで利用





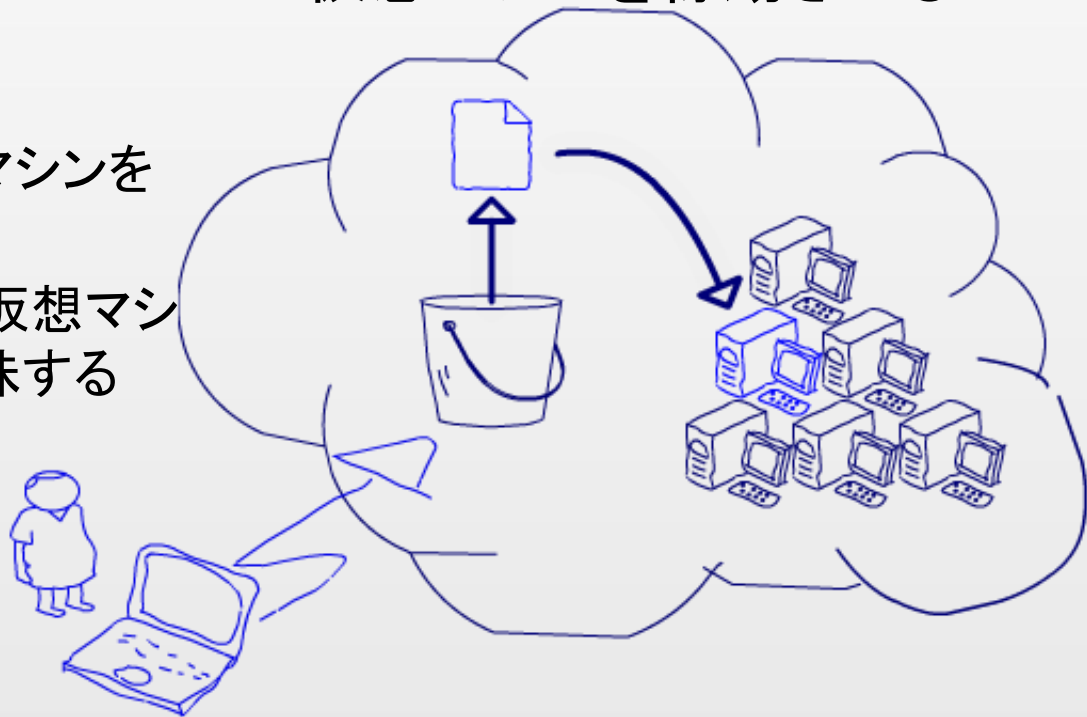
S3機能 -2-

- バケット/オブジェクトに対する主な操作は以下
 - バケット/オブジェクトの作成/追加
 - バケット/オブジェクトの削除
 - オブジェクトの複製
 - バケット/オブジェクトの一覧
 - オブジェクトの取得
 - バケット/オブジェクトに対する権限の設定
 - バケット/オブジェクトに対する権限情報の取得
- 上記以外にもロギングやバージョンングの機能が利用可能だが、REST APIでのみの実装である

インスタンス機能 -1-

- インスタンス機能とは、S3機能が管理するマシンイメージを使ってコンピュートノード上で仮想マシンを稼動させる機能。

- ✓ 「インスタンス」とは仮想マシンを意味する
- ✓ 「コンピュートノード」とは仮想マシンが稼動するノードを意味する





インスタンス機能 -2-

■ イメージキャッシュ

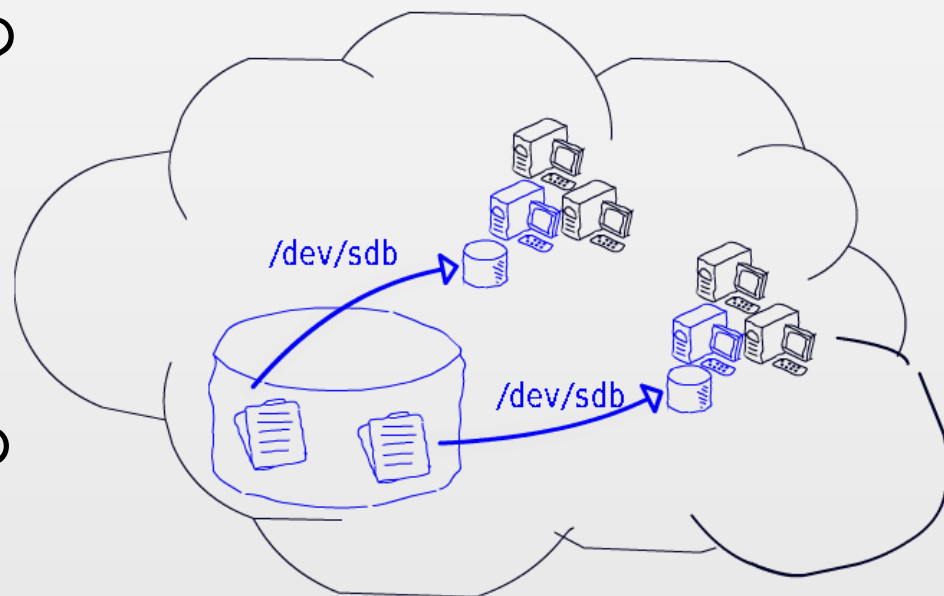
- インスタンスの起動時間を短縮するために、一度起動させたマシンイメージはキャッシュされる
- キャッシュされたマシンイメージは利用頻度の少ないものから削除される「設計」である
 - Eucalyptusのバージョンによっては古いイメージから先に削除される「実装」だったこともある

■ キーペア

- あらかじめ作成しておいたキーペアを用いて、起動するインスタンスにSSHの公開鍵を設定することにより、起動したインスタンスに対してセキュアに接続することが可能
- SSHの公開鍵はEucalyptusがインスタンスに直接埋め込むが、マシンイメージにcloud-initなどのツールを仕込むことによりメタデータから取得して設定することも可能
- もちろん、インスタンスにパスワードを設定しパスワード認証でログインすることも可能。

EBS(Elastic Block Store)機能 -1-

- S3形式のインスタンスを停止するとインスタンス内のデータは削除されるため、必要なデータはEBS機能によって提供される外部ストレージに保存しておくことで、データの永続化が可能
- インスタンスからはローカルデバイス(/dev/sdXや/dev/vdX)のように見えるため、特別なドライバやソフトウェアは不要
 - ただし使用しているハイパーバイザがXenの場合にはxenblkドライバが必要
 - ハイパーバイザーがKVMの場合、Eucalyptusの設定によってはvirtioが利用可能



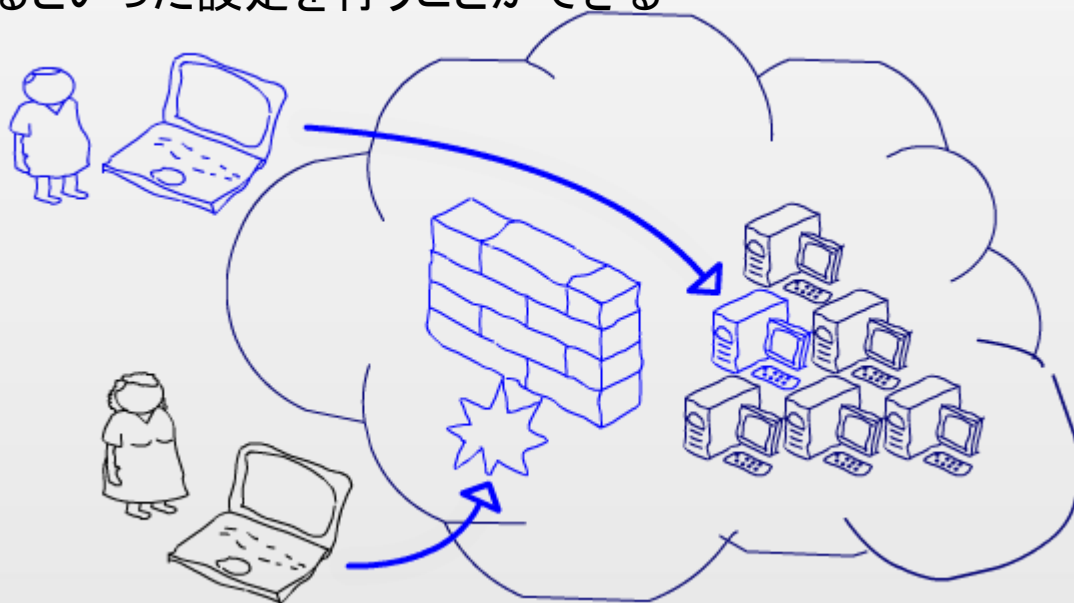


EBS(Elastic Block Store)機能 -2-

- EBS機能はEBSボリュームとEBSスナップショットという2つの要素からなる
 - EBSボリュームはネットワーク経由で提供されるブロックストレージ
 - EBSスナップショットはEBSボリュームのスナップショットを取得する機能
- EBSボリュームは新規で空のボリュームを作成することが可能だが、EBSスナップショットから生成することも可能
- EBSスナップショットはEBSボリュームのように直接インスタンスへ取り付けすることはできない。ただしWalrusにアップロードされたデータをダウンロードして利用することは可能
- v3.x系からはEBS起動のインスタンスを利用することが可能になった

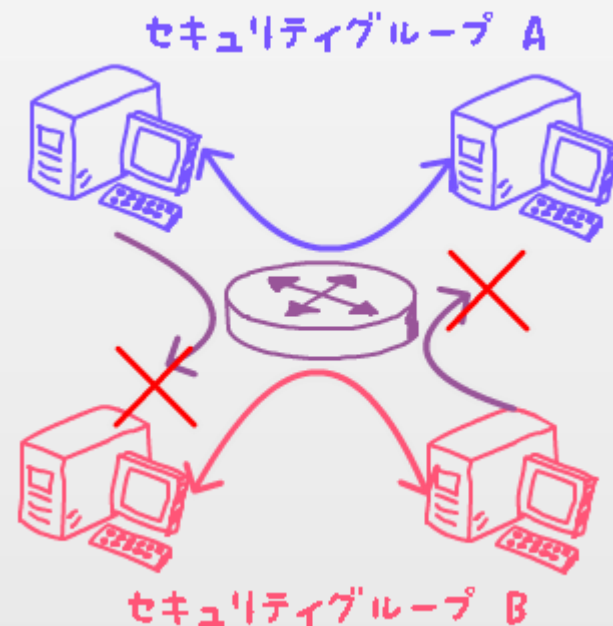
セキュリティグループ機能 -1-

- インスタンスを起動する際は必ず特定のセキュリティグループに所属した状態で起動
- セキュリティグループを設定しないと、起動したインスタンスに対するファイアウォールはすべてのポートを閉じた状態となる
- セキュリティグループを設定することにより、特定の条件下でファイアウォールの特定ポートを開放するといった設定を行うことができる



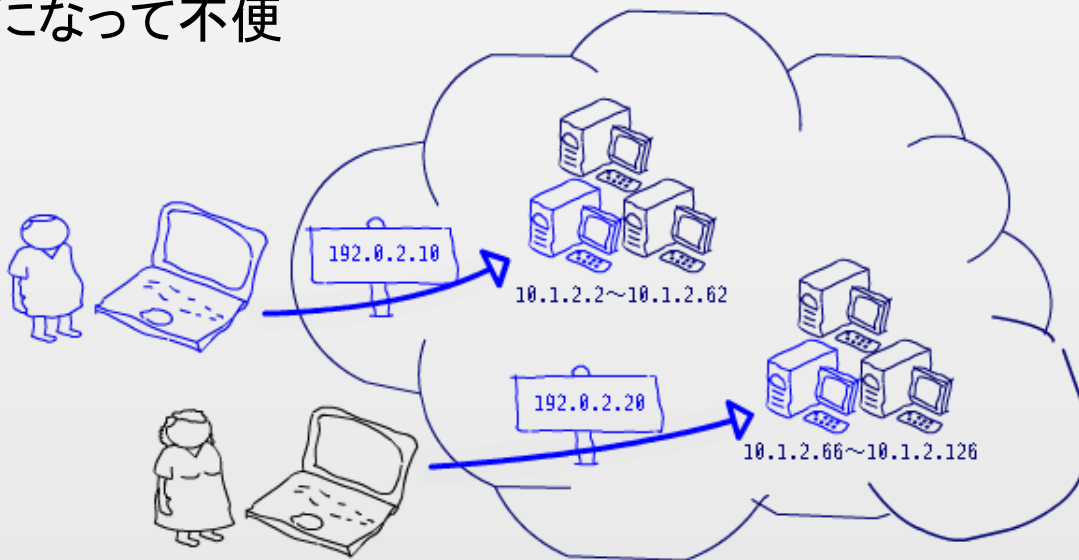
セキュリティグループ機能 -2-

- タグVLANにより以下を実現
 - 他のセキュリティグループのインスタンスとは通信できないようになっている
 - 通信させたい場合はセキュリティグループのルールに適宜設定を行なう
 - ルールは「グループ全体を許可する」か「プロトコルとポート毎に許可」する
 - 他のセキュリティグループから通信内容を傍受されることを防ぐ



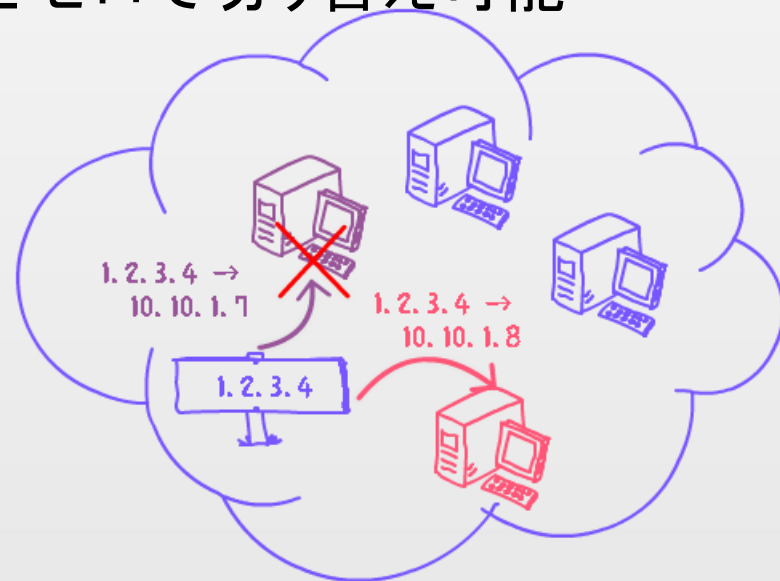
ElasticIP機能 -1-

- インスタンスを起動するとPublicIPが設定されるが、PublicIPはインスタンス停止で開放される
 - つまり他の利用者が利用していたIPアドレスを付番されることもあるし、自分が利用していたIPアドレスが他人のインスタンスに付番されることもある
- これでは外部からアクセスされる用途のサーバではPublicIPが変更される度に通知が必要になって不便
- よって、あらかじめPublicIPを確保することで、インスタンスが停止して再度起動しても同じIPアドレスが利用できるようにした



ElasticIP機能 -2-

- インスタンスに何か問題が発生して、別のインスタンスに切り替えなければならない場合、データなどはEBS機能で複製および取り外し&取り付けが可能だが、PublicIPもこのElasticIPを使用することで、別インスタンスに再Associateすることで、ダウンタイムがほとんどゼロで切り替え可能
- 既にインスタンスで利用されているElasticIPを他のインスタンスにAssociateした場合は、元のインスタンスのIPアドレスは前にAssociateされていたIPアドレスかもしくはPublicIPのプールで未使用のIPアドレスがAssociateされる





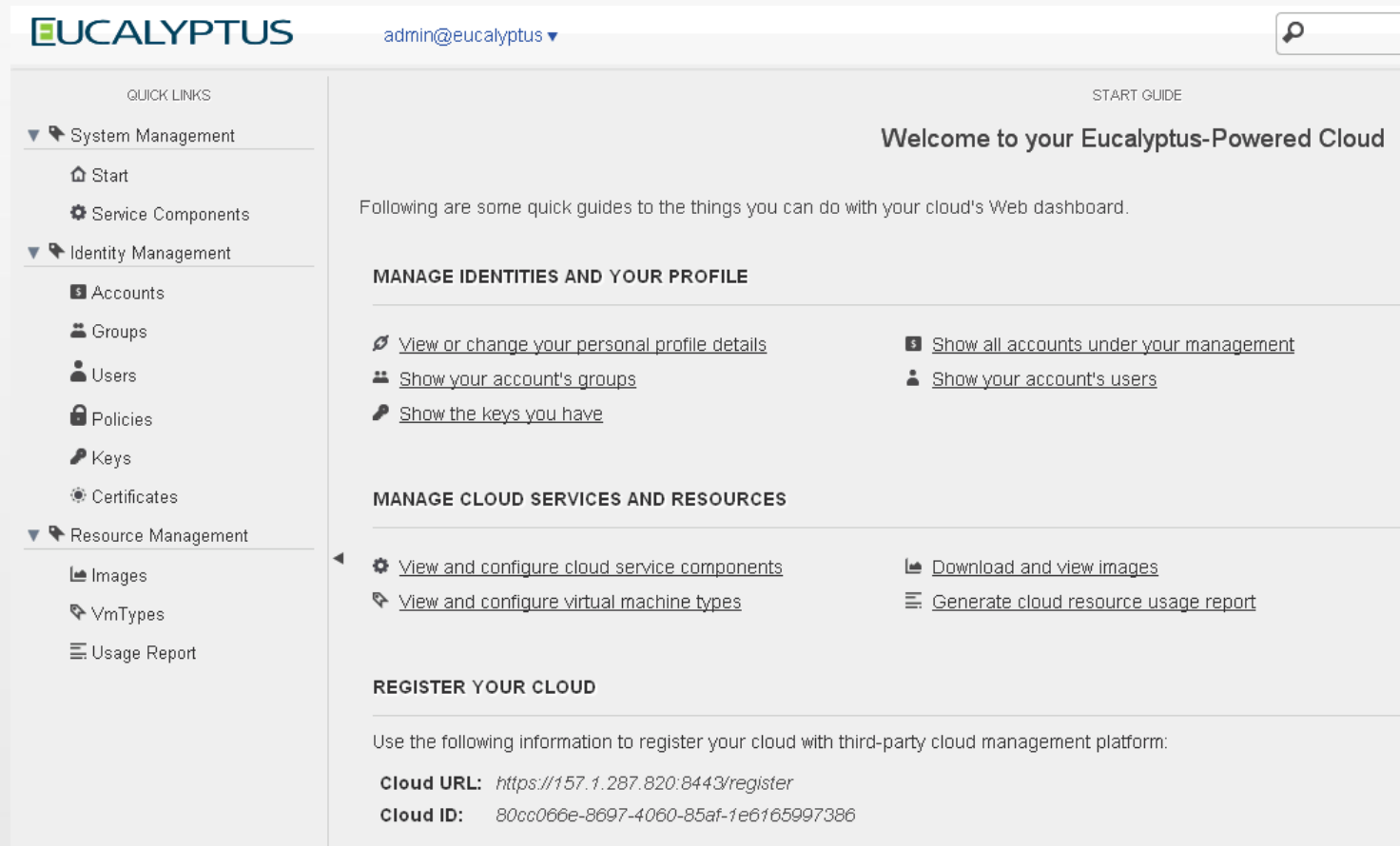
その他のEucalyptusの機能

「WebUI」-1-

- 他のIaaS基盤(OpenStack, CloudStack, Wakame-VDCなど)と比べると限定的な機能しかないが、WebUIによる管理画面には以下の機能(メニュー構成)がある
 - 認証情報
 - イメージ一覧
 - ユーザー一覧
 - 各コンポーネントの設定
 - Extra
 - Eucalyptus社配布のマシンイメージへのリンク
 - クライアントツールへのリンクなど
- v3.0からはWebUIの見た目と一部機能が変更され、大きな違いとしては、以下がある
 - 各コンポーネントの状態情報
 - IAM対応なユーザ管理ページ
 - 利用状況レポートページ

その他のEucalyptusの機能

「WebUI」-2-

The screenshot shows the Eucalyptus WebUI dashboard. At the top, there's a header with the "EUCALYPTUS" logo, a user dropdown menu showing "admin@eucalyptus", and a search icon. The main content area is divided into a left sidebar and a main panel. The sidebar, under "QUICK LINKS", has sections for "System Management" (Start, Service Components), "Identity Management" (Accounts, Groups, Users, Policies, Keys, Certificates), and "Resource Management" (Images, VmTypes, Usage Report). The main panel, under "START GUIDE", has a heading "Welcome to your Eucalyptus-Powered Cloud" and a paragraph: "Following are some quick guides to the things you can do with your cloud's Web dashboard." Below this are three sections: "MANAGE IDENTITIES AND YOUR PROFILE" with links like "View or change your personal profile details", "Show your account's groups", "Show the keys you have", "Show all accounts under your management", and "Show your account's users"; "MANAGE CLOUD SERVICES AND RESOURCES" with links like "View and configure cloud service components", "View and configure virtual machine types", "Download and view images", and "Generate cloud resource usage report"; and "REGISTER YOUR CLOUD" with instructions and fields for "Cloud URL" (https://157.1.287.820:8443/register) and "Cloud ID" (80cc066e-8697-4060-85af-1e6165997386).



その他のEucalyptusの機能 「iSCSI対応」

- EBS機能をAoEではなくiSCSIで提供する機能
- AoEはエンタープライズ用途には色々と問題があるため、iSCSI対応はEucalyptus 1.6系から実装されたが、正式に使えるようになったのは2.0系から
 - AoEには認証機構が実装されていないため、ネットワーク的にリーチ可能な場合にデータを読み書きできてしまう。とはいえ、インスタンスからはリーチ不可能である。
- エンタープライズ版およびEucalyptus3からはEqualLogicなどの市販ストレージも扱えるようになった
 - AoEと同じようにSCのローカルストレージをiSCSI機能で利用する場合、SCのスペックが高くないと性能が出ない。加えて市販ストレージのほうがボリューム作成やスナップショットの取得などの操作が早い(ものがある)
 - v3.0から実装されたEBS起動では市販ストレージを利用するほうが性能の面でも運用の面でも良い(場合がある)



その他のEucalyptusの機能 「マルチクラスタ」

- ネットワークのアーキテクチャ変更により、Eucalyptus 1.5系では一時使用できなくなったマルチクラスタ機能
 - Eucalyptus 1.6系から復活
 - ただし1.4系でと1.6系とではアーキテクチャが異なる
- Eucalyptus 1.6系以降のアーキテクチャではvtunを使用することで複数クラスタに跨ったセキュリティグループの利用が可能



その他のEucalyptusの機能 「高可用性(HA)」

- Eucalyptus2系でも一部のコンポーネントでは高可用性対応が可能だったが、公式なメソッドではなかった
- Eucalyptus3系からは機能として高可用性(HA)に対応
 - 各コンポーネントで故障が発生した際に自動でリカバリされるようになった
 - クラウド管理者に対する自動通知機能も提供される
 - 各コンポーネントの状態監視やリカバリを容易にするためのツールなども提供
- Eucalyptus3系ではDRDBを利用してデータのレプリケーションを実施



その他のEucalyptusの機能 「Eucalyptus版IAM(EIAM)」

- EIAMはAWSのIAMと互換性を持った、認証およびアカウント管理システム
- EIAM機能はユーザIDやグループIDの管理や、それらのアカウント情報に対してリソース割り当てやアクセス制御などを提供
- EIAM機能の一部として、大規模なレポーティング管理フレームワークを組込んだ
- このフレームワークは、ユーザやグループが使用したリソースについてのレポートをクラウド管理者が利用できるようになる



その他のEucalyptusの機能 「AD/LDAP統合」

- Eucalyptus3ではActiveDirectoryもしくはLDAPと連携するディレクトリサービス統合モジュールが提供される
- このモジュールを使うことによって、組織内に既に存在しているディレクトリサービスと連携することが可能
- 既存のユーザやグループの情報をEucalyptusでも使用できるようにし、アカウント管理のコストを減らすことができる



その他のEucalyptusの機能 「Windowsゲスト対応」

- EucalyptusでWindowsゲストを稼働させる機能
- Windowsのマシンイメージ(EMI)の作成を簡素化するいくつかの支援ツールも提供
 - これらのツールのなかには、システム管理者がエンタープライズ向けのWindowsライセンスサーバー(例えばKMS)を利用して、インスタンスに対するボリュームアクティベーションを管理するのを支援するツールもある



その他のEucalyptusの機能 「EBS起動」

- EBS起動機能は、(インスタンスとして起動可能な)EBSボリュームを用いてインスタンスを起動する機能を提供
- この機能はAmazon EC2におけるEBS起動のように、インスタンスデータの永続化およびインスタンスを起動(再開)する時間を短縮することが可能

ネットワークモードについて -1-

- Eucalyptusには4つのネットワークモードがあります
 - SYSTEMモード、STATICモード、MANAGEDモード、MANAGED-NOVLANモード
- SYSTEMモード
 - SYSTEMモードはEucalyptusを1台のマシン上で動作させることができるため、手軽にEucalyptus環境を構築することが可能
 - Elastic IPやセキュリティグループといった機能を利用することができない
 - 別途DHCPサーバを自前で構築する必要あり
 - Eucalyptusクラウドとして独立したネットワークを構成するわけではなく、既存システムと同一ネットワーク上にインスタンスが配置される
- STATICモード
 - STATICモードはMACアドレスとIPアドレスを予め設定しておきインスタンスに割り当てる方式
 - 各インスタンス毎にひとつのIPアドレスのみを持つこととなり、Elastic IPの機能を利用することはできない
 - それぞれのインスタンスはフラットなネットワークを構築し、各セキュリティグループごとに分離されたネットワークに配置することができない

ネットワークモードについて -2-

■ MANAGEDモード

- MANAGEDモードではEucalyptusが提供する機能をフルで利用できる
- 各インスタンスはセキュリティグループごとに別々のセグメントに配置され、Elastic IPの機能によってパブリックIPを割り当てることができる
- CCとNCを繋ぐスイッチにインテリジェンスなスイッチを利用している場合、適切にスイッチを設定しないと——タグVLANが通るように適切に設定しないとEucalyptusが動作しなく、ハマる原因になる
- 本講ではMANAGEDモードを設定します

■ MANAGED-NOVLANモード

- MANAGED-NOVLANモードはMANAGEDモードと違いタグVLANを使用しない
- セキュリティグループごとにセグメントは分離されません。しかしそれ以外はMANAGEDモードと同じように利用できます。

ネットワークモードについて -3-

■ 各ネットワークモードの機能比較

ネットワーク モード/機能	Elastic Ip管理	DHCP	セキュリティグ ループ管理	VLAN	メタデータ
SYSTEM	×	- *1	×	×	×
STATIC	×	×	×	×	×
MANAGED	○	○	○	○	○
MANAGED- NOVLAN	○	○	○	×	○

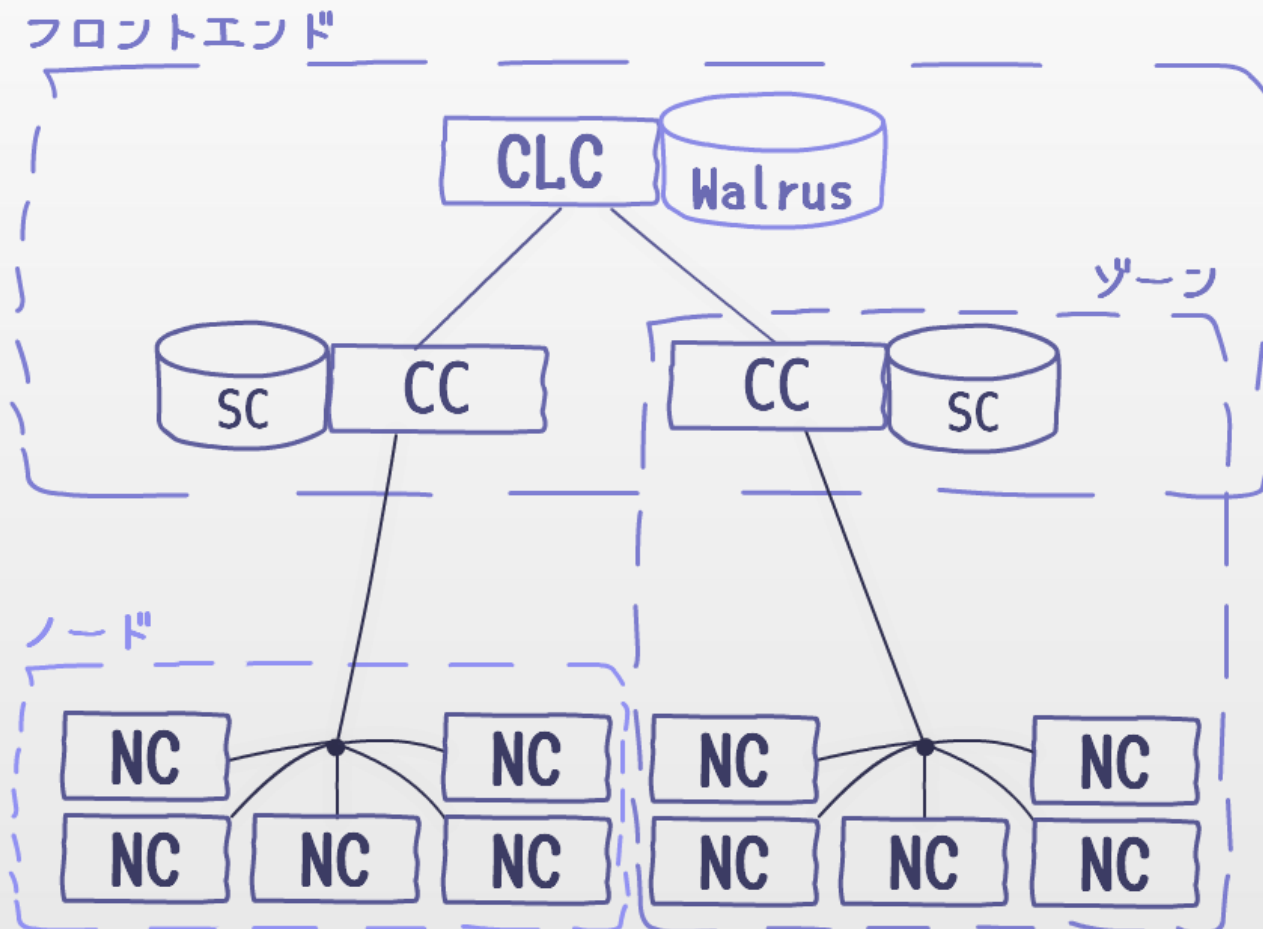
■ *1 Eucalyptusの管理外のDHCPサーバは利用できます。

- VLANは、インスタンス同士の通信網をVLANを使って論理的に分離するかどうかを表します
- メタデータは起動したインスタンスから自分自身の情報(インスタンスIDやインスタンスタイプなど)を取得する機能が利用できるかどうかを表します



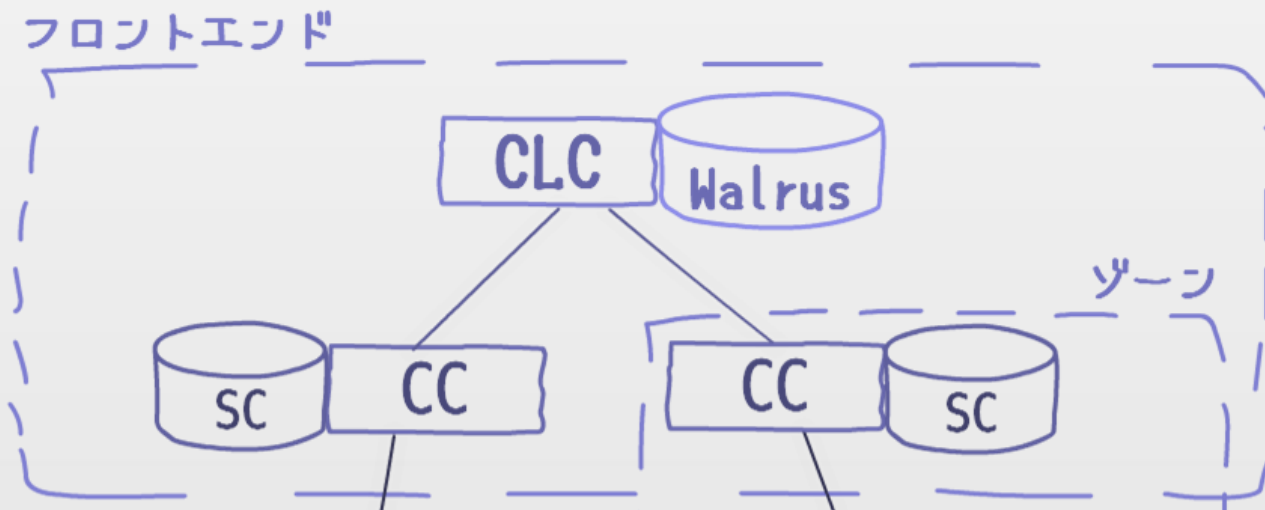
EUCALYPTUSのコンポーネント

Eucalyptusのコンポーネント全体図



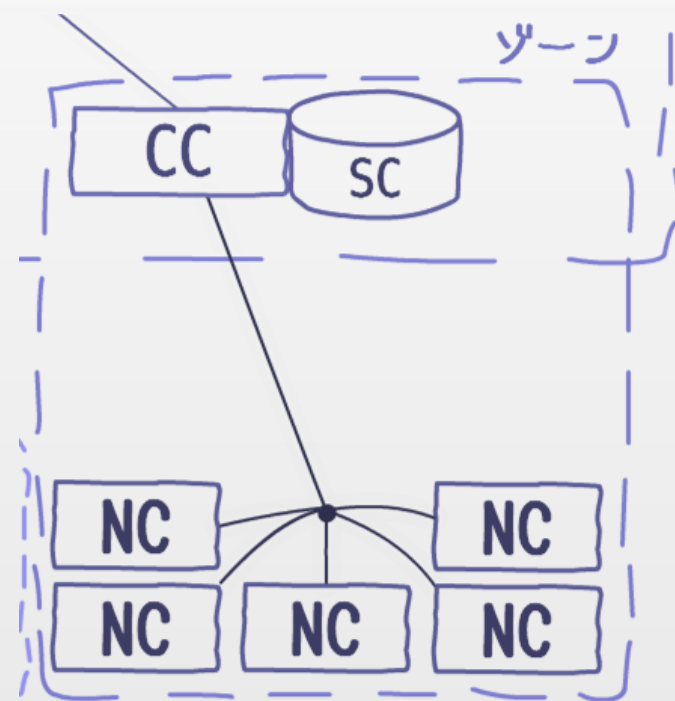
フロントエンド

- Frontendは、主にユーザからの要求を受け付ける機能を持つコンポーネントの集合
- そのため、CLC, CC, Walrusはユーザからアクセス可能なネットワークに配置されなければならない
- SCは直接はユーザと通信しないが、クラスタ毎に配置が必要なので、CCと同居させる構成がよく見られる
 - ただし負荷によってはCCと別居構成にする



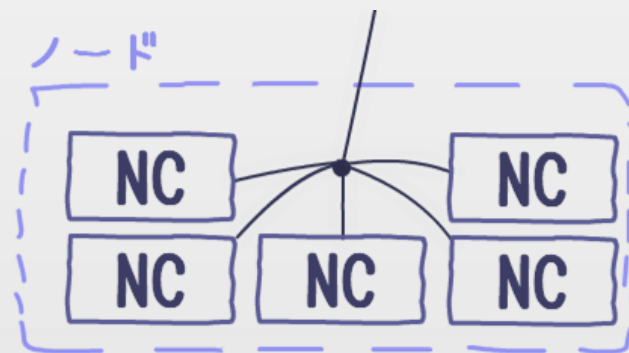
ゾーン

- Zoneは、Nodeとそれらを管理するクラスタコントローラとそのZoneに所属するストレージコントローラの集合です
- マルチクラスタ構成が可能になったv1.6系からはクラスタ毎にストレージコントローラを配置する構成が標準になりました
- v2.0系では、アーキテクチャ的な仕様により、ストレージコントローラはクラウドコントローラと通信可能なネットワークに所属していなければなりません



ノード

- Nodeはその名のとおりノードの集合体
- 1つのゾーン内のノードで利用できるハイパーバイザーは1種類のみ
 - 指定したマシンイメージがどのノード(物理マシン)上で起動するかを指定できないため
 - XenでもKVMでも起動するカーネル、RAMディスク、イメージを作れば混在してても大丈夫かもしれないが…メリットがあまり無い





CLC

- クラウドコントローラ(CLC)には主に以下の役割があります
 - ユーザアカウントの管理や認証
 - ユーザの要求を受け付ける
 - 受け付けた要求をクラスタコントローラ(CC)に渡す
 - 設定値やクラウド内の情報をDBに格納し管理
 - Web管理画面の提供
 - v3.x系からはLDAP/AD連携機能が追加



CC

- クラスタコントローラ(CC)には主に以下の役割があります
 - CLCからの要求を受けノードコントローラ(NC)に処理を要求する
 - NCで起動しているインスタンスのネットワークを制御
 - ユーザがインスタンスに接続する際のPublicIPの提供
 - NCの空きリソースを監視
 - v3.xからはマシンイメージをキャッシュする機能が追加



NC

- ノードコントローラ(NC)には主に以下の役割があります
 - CCからの要求を受け、Walrusからマシンイメージを取得
 - マシンイメージをキャッシュ
 - インスタンスの起動や停止、EBSボリュームの取り付けや取り外しなどの処理をハイパーバイザー(XenおよびKVM)に依頼



SC

- ストレージコントローラ(SC)には主に以下の役割があります
 - ボリュームの管理
 - ボリュームからのスナップショットの管理
 - インスタンスに対するボリュームの提供
 - v3.x系からは外部ストレージ(SAN/NAS)に対してボリューム操作が可能



Walus

- Walrusには主に以下の役割があります
 - S3互換の大規模ストレージ
 - マシンイメージの管理および提供

VMware Broker

- VMware BrokerはESXi上でインスタンスを起動するためのコンポーネントであり、CCからの要求をvCenterに渡す役割を担う



動作に必要な環境



CLCに要求されること

■ CPU

- CPUはそれほど高速である必要はない
 - ただし最低でも2GHz程度でコア数は2コアは必要
- ユーザからのAPIをさばける程度でOK

■ メモリ

- メモリ容量はJavaのプロセスが動くため少なくとも4GB以上必要

■ ディスク

- ディスク容量や性能はほとんど必要としない

■ NIC

- NICの枚数は1枚あれば十分
- もちろん、bonding構成であればより望ましい
- NICのスペックは1Gであれば足りる
- HA構成を構築する場合は、追加でもう1枚あったほうがいい



CCに要求されること

■ CPU

- NCへの通信を処理するためにある程度の性能は必要だが、Xeonクラスの3GHz以上であれば十分

■ メモリ

- メモリもある程度あると安心
- 8GB程度あれば十分

■ ディスク

- CLC同様にディスクはほとんど使わない

■ NIC

- NCへの通信を処理するため、1G以上、10Gあると安心
- もしくは1G数本でbonding構成
- 基本的にCCのNICは2枚構成がスタンダードおよび最小値
- HA構成を構築する場合は更に追加でNIC1枚が必要



NCに要求されること

■ CPU

- ある程度の集約性は欲しいので、コア数は8コア以上
- VT機能はあったほうがよい
 - 特にWindowsなどをゲストOSとして利用する場合

■ メモリ

- メモリもある程度あると安心
- 最低でも8GB以上は必要

■ ディスク

- VM Typesの設定値にもよるが、ある程度のディスク容量は必要
- 最低でも500GB～1TBが望ましい
- マシンイメージのコピー処理が多いため、ディスクの回転数自体も高速なものが望ましい

■ NIC

- インスタンスによってネットワークの利用率が異なるため、一概に10G必須とは言えないが、1Gは必須



SCに要求されること

- CPU
 - 特にCPUに要求はないが、I/Oの性能が低下しない程度のスペックは必要
- メモリ
 - CPU同様に要求はないが、EBSの利用が多い場合にI/Oの性能低下が発生しない程度の容量は必要
- ディスク
 - EBS機能の利用が多い場合にはディスクは市販ストレージにしたほうが望ましい
 - もちろん、容量/性能どちらも優れているほうが良い
- NIC
 - NC側——つまりバックエンド側で使用するNICは10Gのほうが望ましい
 - もちろん1G複数でbondingでも可



Walrusに要求されること

■ CPU

- 特にCPUに要求はないが、I/Oの性能が低下しない程度のスペックは必要
- マシンイメージの登録時にはCPUパワーが要求される

■ メモリ

- CPU同様に要求はないが、Walrusをマシンイメージの管理のみならずオブジェクトストアとして利用する場合にはI/Oの性能低下が発生しない程度の容量は必要

■ ディスク

- 利用度にもよるが、外部ストレージを利用するようにしたほうが、ディスク容量が少なくなった場合に対応できるので良い
- もちろん、容量/性能どちらも優れているほうが良い

■ NIC

- NC側——つまりマシンイメージを提供する側のNICは10Gのほうが望ましい
- もちろん1G複数でbondingでも可



ネットワーク機器に要求されること

- EucalyptusのネットワークモードをMANAGEDに設定する場合は、CCとNCを接続するスイッチにインテリジェントなスイッチを用意すべき
- バックエンド側のNICでもbonding設定を使用する場合には、それなりのポート数を保有するスイッチが必要



ディスク容量の最小値

■ v3.1のマニュアルにて以下の情報が記述されている

コンポーネント	ディレクトリ	最小値 (GB)
CLC	/var/lib/eucalyptus/db	20
CLC	/var/log/eucalyptus	2
Walrus	/var/lib/eucalyptus/bukkits	250
Walrus	/var/log/eucalyptus	2
SC	/var/lib/eucalyptus/volumes	250
CC	/var/lib/eucalyptus/CC	5
CC	/var/log/eucalyptus	2
NC	/var/lib/eucalyptus/instances	250
NC	/var/log/eucalyptus	2



EUCALYPTUSのディレクトリ構造



共通する構造

■ 各コンポーネントが共通で使用するディレクトリ

項番	ディレクトリパス	説明
1	/etc/bash_completion.d/	bashの補完機能使用時にeuca_confのオプションを補完する設定を格納 v3.1からは使われていない
2	/etc/eucalyptus/	Eucalyptusの設定ファイル類を格納
3	/opt/euca-axis2c/	Eucalyptusが利用するAxis2/CやRampart/Cのファイルを格納 ディストリビューションによっては使われていない
4	/usr/lib/eucalyptus/	Eucalyptusが内部処理で利用するコマンドやライブラリを格納
5	/usr/sbin/	Eucalyptusの管理用コマンドを格納
6	/usr/sbin/euca_admin/	Eucalyptusの管理用コマンドで使用する共通ライブラリを格納 v3.1からは使われていない
7	/usr/share/doc/eucalyptus-X.Y.Z/	Eucalyptusのドキュメント類を格納
8	/usr/share/eucalyptus/	主にCLC, SC, Walrusの本体やEucalyptusが内部処理で利用するスクリプトを格納
9	/var/lib/eucalyptus/keys/	WS-Security使用時に各コンポーネント間の通信を暗号化/復号化するための証明書類を格納
10	/var/log/eucalyptus	Eucalyptusのログを格納
11	/var/run/eucalyptus	Eucalyptusの各コンポーネント起動時の一時情報を格納



CLC固有の構造

■ CLCが使用するディレクトリ

項番	ディレクトリパス	説明
1	/etc/eucalyptus/cloud.d/	Web管理画面に関する表示上の設定や旧バージョンからのアップグレードスクリプトなどを格納
2	/var/lib/eucalyptus/db/	CLCが管理するDBを格納
3	/var/lib/eucalyptus/webapps/	Web管理画面のアーカイブファイルを格納
4	/var/run/eucalyptus/webapp/	Web管理画面の実行時ファイルを格納

■ CLCが出力するログファイル

項番	ファイル名	説明
1	cloud-debug.log	CLCが出力するDEBUGレベルまでのログ。ログのローテートは1～10まで。
2	cloud-error.log	CLCが出力するERRORレベルだけのログ。ログのローテートは1～10まで。
3	cloud-exhaust.log	DBやクライアントなどの接続情報を全て出力。ログのローテートは未定義。デフォルト設定では出力しない。
4	cloud-output.log	CLCが出力するINFOレベルまでのログ。ログのローテートは1～10まで。
5	cloud-cluster.log	v3.1から出力されるようになったログ。
6	cloud-extreme.log	v3.1から出力されるようになったログ。
7	db-err.log	v3.1から出力されるようになったログ。
8	db.log	v3.1から出力されるようになったログ。
9	jetty-request-YYYY_MM_DD.log	Web管理画面で使用しているJettyが出力するログ。ログのローテートは日付単位。

CC固有の構造

■ CCが使用するディレクトリ

項番	ディレクトリパス	説明
1	/opt/euca-axis2c/services/EucalyptusCC	CCの本体を格納 (v2.xまで)
2	/usr/lib64/axis2c/services/EucalyptusCC	CCの本体を格納 (v3.1から)
3	/var/lib/eucalyptus/CC/	CCのプロセスが保有するメモリ情報をキャッシュとして格納
4	/var/run/eucalyptus/net/	CCが起動するDHCPサーバの管理情報を格納
5	/dev/shm/	CCが使用するセマフォ情報を格納

■ CCが出力するログファイル

項番	ファイル名	説明
1	axis2c.log	Axis2/Cが出力するログ。1世代前のログはサフィックス「.old」が付与される。
2	cc.log	CCが出力するログ。ログのローテートは0～5まで。
3	httpd-cc_error_log	CCのプロセスの標準出力。ログローテートはなし。



NC固有の構造

■ NCが使用するディレクトリ

項番	ディレクトリパス	説明
1	/opt/euca-axis2c/services/EucalyptusNC/	NCの本体を格納 (v2. xまで)
2	/usr/lib64/axis2c/services/EucalyptusNC/	NCの本体を格納 (v3. 1から)
3	/usr/local/eucalyptus/eucalyptus/cache/	NCが使用するマシンイメージのキャッシュを格納 (v2. xまで)
4	/var/lib/eucalyptus/instances/cache/	NCが使用するマシンイメージのキャッシュを格納 (v3. 1から)
5	/usr/local/eucalyptus/ユーザ名/インスタンスID/	インスタンスが使用するデータを格納 (v2. xまで)
6	/var/lib/eucalyptus/instances/work/	インスタンスが使用するデータを格納 (v3. 1から)
7	/dev/shm/	NCが使用するセマフォ情報を格納

■ NCが出力するログファイル

項番	ファイル名	説明
1	axis2c.log	Axis2/Cが出力するログ。1世代前のログはサフィックス「.old」が付与される。
2	euca_test_nc.log	NCのプロセス起動時に実施するlibvirtやKVMやXenまわりのチェック結果が出力される。
3	httpd-nc_error.log	NCのプロセスの標準出力。ログローテートはなし。
4	nc.log	NCが出力するログ。ログのローテートは0～5まで。
5	local-net	インスタンスのネットワーク情報が出力される。(v3. 1から)

SC固有の構造

■ SCが使用するディレクトリ

項番	ディレクトリパス	説明
1	/var/lib/eucalyptus/volumes/	SCが管理するボリュームとスナップショットを格納

■ SCが出力するログファイル

項番	ファイル名	説明
1	cloud-debug.log	SCが出力するDEBUGレベルまでのログ。ログのローテートは1～10まで。
2	cloud-error.log	SCが出力するERRORレベルだけのログ。ログのローテートは1～10まで。
3	cloud-output.log	SCが出力するINFOレベルまでのログ。ログのローテートは1～10まで。
4	sc-stats.log	SCがボリュームの作成などを実施した際に出力するログ。

Walrus固有の構造

■ Walrusが使用するディレクトリ

項番	ディレクトリパス	説明
1	/var/lib/eucalyptus/bukkits/	Walrusが管理するバケットやオブジェクトを格納

■ Walrusが出力するログファイル

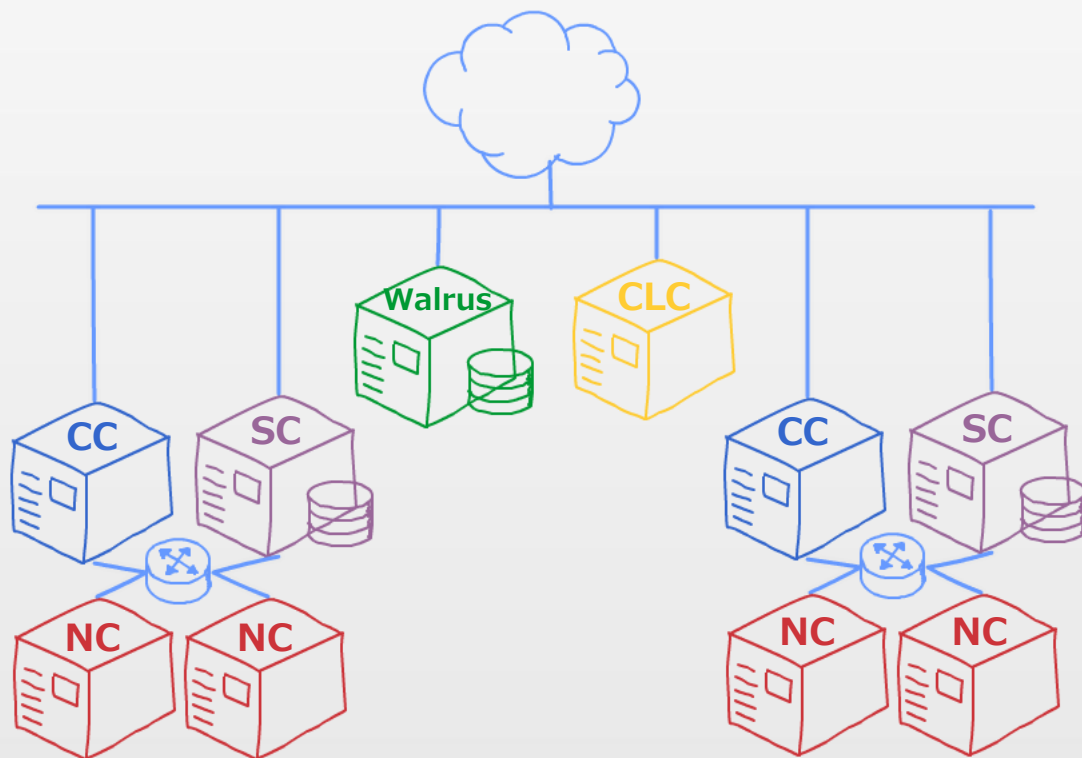
項番	ファイル名	説明
1	cloud-debug.log	Walrusが出力するDEBUGレベルまでのログ。ログのローテートは1～10まで。
2	cloud-error.log	Walrusが出力するERRORレベルだけのログ。ログのローテートは1～10まで。
3	cloud-output.log	Walrusが出力するINFOレベルまでのログ。ログのローテートは1～10まで。
4	walrus-stats.log	Walrusがオブジェクトを格納した際に出力するログ。



バージョン3.1での構成例

マルチクラスタ構成

- 下記の構成の前提
 - Walrus、SCは外部ストレージを使用せず、自身のストレージを利用する構成
- CLC、Walrus、CCはユーザから到達可能な場所へ配置すること
- SCはユーザから到達可能である必要はないが、CLCとは通信が可能な場所に配置すること
 - EBSのスナップショット機能はWalrus上にスナップショットイメージをアップロードするため、Walrusとも通信可能であることが望ましい
- CC、SCはNCと通信可能な場所へ配置すること



HA構成

- 下記の構成の前提
 - SC、Walrusのストレージはそれぞれ外部ストレージを使用する
 - CLCでのユーザ認証はLDAP/AD連携を使用する
- CLC、CC、SC、Walrusは各2台ずつ用意し、Active/Standby構成とする
 - 図中にて明記していないが、この4コンポーネントは死活監視用のNICおよびネットワークが必要
 - 同じく明記していないが、Standby側も正常系のネットワークに接続すること

- LDAP/ADはCLCと通信可能とする
- Walrus用の外部ストレージはWalrusと通信可能なこと
- SC用の外部ストレージはSCおよびNCと通信可能なこと

