8.OpenStack Image Service

岡山大学大学院自然科学研究科
電子情報システム工学専攻
谷口研究室所属 M2 千崎良太

目次

目次

Image Serviceの概要

＜用途＞

仮想計算機のImageの発見，登録，読み出し

＜構成＞

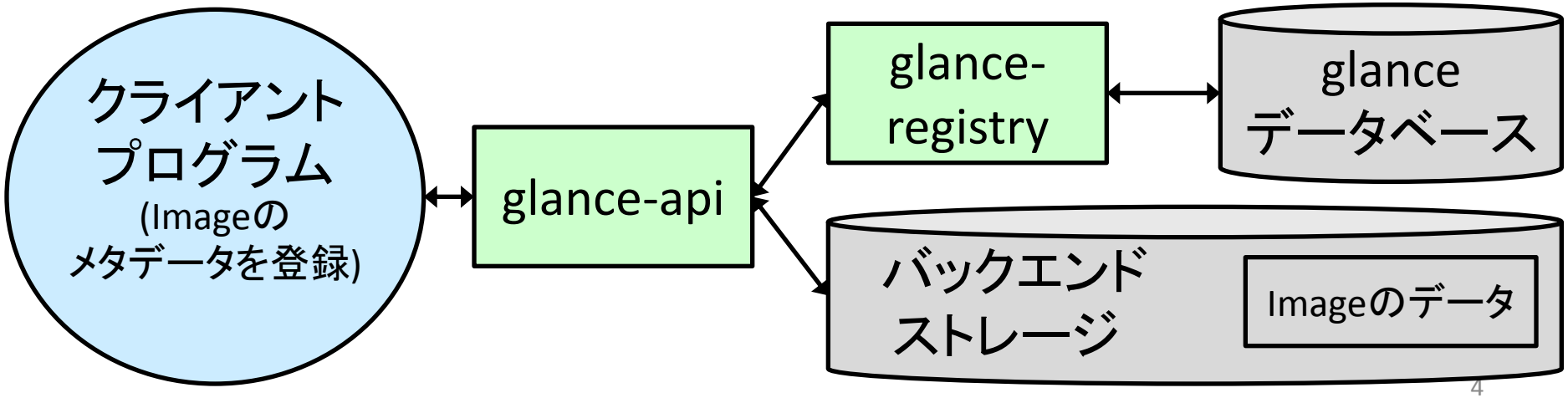
(1) API server

(2) Registry server(s)

＜特徴＞

(1) 様々なバックエンドのストレージとデータベースを提供するために柔軟に設計

(2) REST APIを提供



OpenStack Image Service API Server

(1) API server (“glance-api”プログラム)はOpenStack Image Serviceのインタフェース

(2) API serverはクライアントプログラムとストレージとの間のやりとりを担当

＜使用可能なバックエンドストレージ＞

(1) OpenStack Object Storage

高可用性なストレージ

(2) Filesystem

(A) ローカルのファイルシステムにImageファイルを格納

(B) 初期設定でImageはファイルシステムに格納

(3) S3

Amazon S3

(4) HTTP

仮想計算機ImageをHTTPを介して読み込み可能

OpenStack Image Service Registry Servers

(1) OpenStack Image Service Registry Serversは
OpenStack Image Service Registry APIに準拠するサーバ

(2) OpenStack Image Serviceには実装の参考文献が付属

目次

構築の必要要件

＜ハードウェア＞

OpenStackのコンポーネントが通常のハードウェア上で動作

＜OS＞

Ubuntu上で動作

＜ネットワーク＞

1000Mbps推奨

＜データベース＞

SQLAlchemyと互換性のあるデータベース(例：MySQL，Oracle)

SQLAlchemy：PythonのためのORM(Object-Relational Mapping)ライブラリ

ORM：データベースとオブジェクト指向プログラミング言語の間の非互換なデータを変換するプログラミング技法

＜権限＞

rootかsudoer

インストール手順

シングルノードにサービスを構築する方法の説明

同サーバにAPI serverとregistry serverを構築

(1) glanceのリポジトリをPPA(Personal Package Archive)に追加

```
$ sudo add-apt-repository ppa:glance-core/trunk
```

(2) apt-getのアップデート

```
$ sudo apt-get update
```

(3) glanceサーバのインストール

```
$ sudo apt-get install glance
```

Glance開発者の場合，Bazaarブランチからドキュメントを入手可能

起動方法

(1) glance-apiプログラムを使用する方法

＜マニュアルの例＞

```
$ sudo glance-api {CONFPATH}
```

＜実際例＞

```
$ sudo glance-api /etc/glance/glance-api.conf
```

(2) glance-controlプログラムを使用する方法

＜マニュアルの例＞

```
$ sudo glance-control {SERVER} {OPERATION} {CONFPATH}
// SERVER = all，registry，api
// OPERATION = start，stop，restart
```

＜実際例＞

```
$ sudo glance-control registry start /etc/glance/glance-registry.conf
```

nova.confの設定

nova.confに以下の項目を記述

設定項目	初期値	説明
--glance_host	None	GlanceサーバのIPアドレス
--image_service	None	nova.image.glance.GlanceImageServiceを設定

Glanceサーバの設定

設定ファイルを指定せずに，glance-apiを起動した場合，以下のいずれかのディレクトリに”glance.conf”という設定ファイルが作成

(1) .

(2) ~/.glance

(3) ~/

(4) /etc/glance

(5) /etc

Ubuntu 10.10に構築した環境では，以下の設定ファイルが作成

(1) /etc/glance/glance-api.conf

(2) /etc/glance/glance-registry.conf

glance-api.confの設定(1/2)

glance-api.confに以下の項目を記述

設定項目	初期値	説明
verbose	True	ログを詳細に出力
debug	False	ログにデバッグ情報を出力
default_store	file	バックエンドストレージの設定 他に‘swift’と’s3’が設定可能
bind_host	0.0.0.0	APIサーバのIPアドレス
bind_port	9292	APIサーバのポート番号
registry_host	0.0.0.0	registryサーバのIPアドレス
resigtry_port	9191	registryサーバのポート番号

glance-api.confの設定(2/2)

設定項目	初期値	説明
log_file	/var/log/glance/api.log	ログの出力先
filesystem_store_datadir	/var/lib/glance/images/	バックエンドストレージがファイルシステムの場合のImagesの格納場所
swift_store_auth_address	127.0.0.1:8080/v1.0/	Swiftの認証サービスのIPアドレス
swift_store_user	Jdoe	Swiftの認証サービスに対するユーザ
swift_store_key	a86850deb2742ec3cb4158e26aa2d89	Swiftの認証サービスに対する認証鍵
swift_store_container	glance	SwiftでImageを格納するためのアカウント
swift_store_create_container_on_put	False	上記のアカウントが存在しない場合に，アカウントを作成するか否か

glance-registry.confの設定

glance-registry.confに以下の項目を記述

設定項目	初期値	説明
verbose	True	ログを詳細に出力
debug	False	ログにデバッグ情報を出力
bind_host	0.0.0.0	registryサーバのIPアドレス
bind_port	9191	registryサーバのポート番号
log_file	/var/log/glance/registry.log	ログの出力先
sql_connection	sqlite:////var/lib/glance/glance.sqlite	SQLAlchemyが接続するためパス
sql_idel_timeout	3600	SQLAlchemyがデータベースへの接続を復旧するまでの時間(秒)

Logファイルの設定

glance.confに以下の項目を記述

設定項目	初期値	説明
--log-config=PATH	None	ログ設定ファイルのパス
--log-format	%(asctime)s %(levelname)8s [%(name)s] %(message)s	ログを記録するフォーマット
--log_file	None	ログを出力するファイル名
--log_dir	None	ログを出力するディレクトリのパス
--log_date_format	%Y-%m-%d %H:%M:%s	ログの日付のフォーマット

目次

OpenStack Image Service REST API

GlanceはImageのメタデータとImageのデータ自身を操作可能なREST 風のAPIを提供

＜REST APIを使用する想定環境＞

(1) Glance API Server(/bin/glance-api)が実行

(2) Glance APIC ServerがURL:http://glance.example.comで実行

＜REST APIで可能な操作＞

(1) Imageの一覧の取得

(2) Imageメタデータの詳細情報を取得

(3) 特定のImageメタデータの詳細情報を取得

(4) Imageの読み出し

(5) Imageの追加

(6) Imageの更新

Imageの一覧の取得

＜操作＞

GET http://glance.example.com/images/

＜返却(JSON形式)＞

```
{'images': [
  {'uri': 'http://glance.example.com/images/1',
   'name': 'Ubuntu 10.04 Plain',
   'type': 'kernel',
   'size': '5368709120'}
...]}
```

＜注意点＞

(1) GETで返却されるImageは全てパブリックImage

Imageメタデータの詳細情報を取得

＜操作＞

GET http://glance.example.com/images/detail

＜返却(JSON形式)＞

```
{'images': [
 {'uri': 'http://glance.example.com/images/1',
  'name': 'Ubuntu 10.04 Plain 5GB',
  'type': 'kernel‘,
  'size': '5368709120',
  'store': 'swift‘,
  'created_at': '2010-02-03 09:34:01',
  'updated_at': '2010-02-03 09:34:01‘,
  'deleted_at': '',
  'status': 'active‘,
  'is_public': True,
  'properties': {'distro': 'Ubuntu 10.04 LTS'}},
 ...]}
```

＜注意点＞

(1) タイムスタンプは全てUTC

(2) updated_atはメタデータの最終更新日時

(3) propertiesはkey/valueペアを記述

特定のImageメタデータの詳細情報を取得

＜操作＞

HEAD http://glance.example.com/images/1

＜返却(HTTPヘッダ)＞

x-image-meta-uri	http://glance.example.com/images/1
x-image-meta-name	Ubuntu 10.04 Plain 5GB
x-image-meta-type	kernel
x-image-meta-size	5368709120
x-image-meta-store	swift
x-image-meta-created_at	2010-02-03 09:34:01
x-image-meta-updated_at	2010-02-03 09:34:01
x-image-meta-deleted_at	
x-image-meta-status	available
x-image-meta-is_public	True
x-image-meta-property-distro	Ubuntu 10.04 LTS

＜注意点＞

(1) x-image-meta-updated_atはメタデータの最終更新日時

(2) x-image-meta-property-[key] [value]を設定

Imageの読み出し

＜操作＞

GET http://glance.example.com/images/1

＜返却(HTTPヘッダ)＞

x-image-meta-uri	http://glance.example.com/images/1
x-image-meta-name	Ubuntu 10.04 Plain 5GB
x-image-meta-type	kernel
x-image-meta-size	5368709120
x-image-meta-store	swift
x-image-meta-created_at	2010-02-03 09:34:01
x-image-meta-updated_at	2010-02-03 09:34:01
x-image-meta-deleted_at	
x-image-meta-status	available
x-image-meta-is_public	True
x-image-meta-property-distro	Ubuntu 10.04 LTS

＜注意点＞

(1) 返却されたContent-Lengthヘッダはx-image-meta-sizeと同一

(2) 返却されるHTTPのbodyにはImageデータ自体が格納

Imageの追加

＜操作＞

POST http://glance.example.com/images/

＜送信対象＞

(1) HTTPヘッダ

Imageのメタデータ

(2) body

MIME(Multipurpose Internet Mail Extensions)形式で
エンコードされたImageデータ自体

＜注意点＞

以下の2つの処理をGlance APIで一度に実行可能

(1) Glance内にImageデータを保存

(2) Glance内にImageのメタデータを保存

Imageの更新

＜操作＞

PUT http://glance.example.com/images/1

＜返却＞

ImageのメタデータをHTTPヘッダとして返却

＜注意点＞

(1) Imageが事前に予約されていれば(キュー状態ならば),
bodyにImageデータを追加可能

(2) 既にImageがメタデータと関連付けされているならば
(キュー状態でなければ), 409コンフリクトを返却

HTTPヘッダの形式(1/2)

(1) x-image-meta-name(必須)

(A) Imageの名前

(B) 1つのGlanceノードに対してユニークな名前でなくても可能

(2) x-image-meta-id(オプション)

(A) Imageの識別番号

(B) 既に存在する番号ならば，Glanceは409コンフリクトを返却

(3) x-image-meta-store(オプション)

(A) 使用するバックエンドストレージ

(B) file，s3，swiftを指定可能

(C) 提供されていないバックエンドストレージの場合，
Glanceは400Badリクエストを返却

HTTPヘッダの形式(2/2)

(4) x-image-meta-type(必須)
  (A) Imageの形式
  (B) kernel，machine，raw，ramdiskを指定可能
(5) x-image-meta-size(オプション)
  (A) 送信するbodyのサイズ
  (B) 送信するbodyのサイズと一致していなければ，Glanceは400Badリクエストを返却
(6) x-image-meta-is_public(オプション)
  (A) パブリックImageか否か
  (B) on，1，trueの場合パブリック
  (C) パブリックの場合，全てのユーザがGlanceからImageとメタデータを閲覧可能
(7) x-image-meta-property-*(オプション)
  (A) key/valueペアを記述
  (B) 作成数の制限はないが，ヘッダサイズの上限は8KB