

# 4.OpenStack Compute の設定

池田 騰

# 目次

(1) はじめに

(2) OpenStack Computeの設定

(A) 設定の概要

(B) 各種設定

# 目次

(1) はじめに

(2) OpenStack Computeの設定

(A) 設定の概要

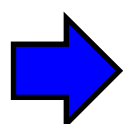
(B) 各種設定

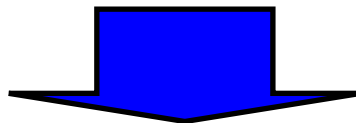
# はじめに

OpenStackは、以下の3つの重要な機能から構成

- (1) OpenStack Compute
- (2) OpenStack Object storage
- (3) OpenStack Image Store

これらの機能は独立に構築されていても、協調動作することが可能

 独立に構築した後、各機能を独立させることも可能であり、  
接続要素とすることも設定可能



OpenStack Computeの設定について説明

# 目次

(1) はじめに

(2) OpenStack Computeの設定

(A) 設定の概要

(B) 各種設定

# 設定の概要

”nova.conf flag file”を用いてOpenStack Computeを設定

<nova.conf flag file>

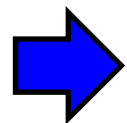
(1) 大部分の構成情報を格納

(2) 初期設定で `/etc/nova/nova.conf` に格納

nova serviceを使用する際 `--flagfile` オプションによって  
設定ファイルを使用可能

```
$ nova-<servicename> --flagfile=[nova.conf PATH]
```

novaの各サービスは、与えられたfile名からフラグを決定



デバッグやパフォーマンスチューニングに便利

# 各種設定

以下の10点の設定について説明

- (1) 4.1.一般的な用途の設定
- (2) 4.1.全ての service に関する設定
- (3) 4.2.ログに関する設定
- (4) 4.3.Hypervisors に関する設定
- (5) 4.4.IPv6 を使用するための設定
- (6) 4.5.Image service と Storage for Compute に関する設定
- (7) 4.6.LiveMigration に関する設定
- (8) 4.7.データベースに関する設定
- (9) 4.8.メッセージシステムに関する設定
- (10) 4.9.認証と許可の設定

## 4.1.一般的な用途の設定

flag file と node についての情報を得るために使用

以下のフラグを nova.conf に記述

フラグ	デフォルト	説明
--my_ip	無し	IPアドレス;ホストIPアドレス
--host	無し	文字列;クラウドコントローラをホストしているノード名
-,--[no]help	無し	ヘルプを表示
--[no]helpshort	無し	そのモジュールの仕様のみを表示
--[no]helpxml	無し	ヘルプをテキストの代わりにXMLで出力

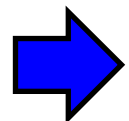


## 4.1.全てのserviceに関する設定

以下のフラグを nova.conf に記述

フラグ	デフォルト	説明
--state_path	'/Users/username/nova/../../'	ディレクトリパス; nova の state を持つ トップレベルディレクトリ
--periodic_interval	'60'	整数値; 周期的なタスクのインターバル 時間(秒)
--report_interval	'10'	整数値; 各ノードがデータストアに状態 を記録するインターバル時間(秒)

--state\_path フラグで Coumpute の状態を保存した Image を  
格納しているトップレベルディレクトリを指定



各サービスの状態を維持可能

## 4.2.ログに関する設定(1/3)

OpenStack Computeは, ログに関して, 以下の3つを設定可能

- (1) ログを記録する場所
- (2) 記録するレベル
- (3) 記録する書式

以下のフラグを nova.conf に記述

### <ログを記録する場所>

フラグ	デフォルト	説明
--logdir	'/var/logs/nova'	ディレクトリパス; サービスごとのログファイルを指定したディレクトリに出力
--logfile	"	ファイル名; 指定したファイルに出力
--[no]use_syslog	'false'	syslog に syslog の命名規則を使用して出力

## 4.2.ログに関する設定(2/3)

### <記録するレベル>

フラグ	デフォルト	説明
--deralt_log_levels	'amqplib=WARN,sqlalchemy=WARN,eventlet.wsgi.server=WARN'	名付けられた1組の logger と記録されるメッセージのレベル: logger=LEVEL の組のリスト
--verbose	'false'	1またはtrueをセット:デバッグの詳細表示 - 使用は任意だが, 初期セットアップの手助けになる

## 4.2.ログに関する設定(3/3)

### <記録する書式>

フラグ	デフォルト	説明
--logging_context _format_string	'%(asctime)s %(levelname)s %(name)s [% (request_id)s %(user)s %(project)s] %(message)s'	ログメッセージに使用する フォーマットを追加した context に変更
--logging_debug _format_suffix	'from %(processName)s (pid=%(process)d) %(funcName)s %(pathname)s: %(lineno)d'	レベルがデバッグの時にロ グフォーマットにデータを与 える
--logging_default _format_string	'%(asctime)s %(levelname)s %(name)s [-] %(message)s'	ログメッセージに context を 追加していないフォーマット を使用
--logging_ exception_prefix	'(%(name)s):TRACE'	文字列;各例外ラインの前に このフォーマットを置く

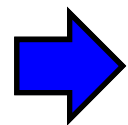
## 4.3.Hypervisors の設定

### <Hypervisors>

- (1) 仮想計算機を実現するための制御プログラム
- (2) OpenStack Compute に必要

OpenStack Computeは、いくつかの Hypervisorsと仮想化標準に対応している

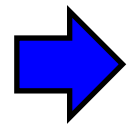
特定のHypervisorを使用する構成



いくつかのインストールと構成処理が必要

## 4.4.IPv6 アドレスを使用するための 設定(1/2)

IPv4とIPv6のアドレスの両方をコミュニケーションに使用



Compute を IPv4 / IPv6 デュアルスタック・モード で使用する  
設定が必要

### <IPv4 / IPv6 デュアルスタック・モード>

- (1) IPv6のグローバルユニキャストアドレスをステートレスなアドレスの自動構成メカニズム[ RFC 4862/2462 ]で得ることが可能
- (2) VlanManager と FlatDHCPManager の両ネットワーク・モードで動作
- (3) Bexar のリリース において フローティング IPs はサポートされていない

## 4.4.IPv6 アドレスを使用するための 設定(2/2)

### <VlanManager>

異なる 64 ビットグローバル・ルーティング・プレフィクスアドレスを, プロジェクト毎に設定

### <FlatDHCPManager>

1つの 64 ビットのグローバル・ルーティング・プレフィクスアドレスを, すべてのインスタンスに設定

- Cactus のリリース は, 要求されるデータベースマイグレーションにおいて FlatManager ネットワーク・モードのサポートを含む

nova-service を実行する各ノードで以下の2つをインストール

- (1) python-netaddr
- (2) radvd

# 設定例(1/4)

(1) 各 nova-network ノードで, python-netaddr をインストール

```
# sudo apt-get install -y python-netaddr
```

(2) IPv6ネットワークを構成

各 nova-network で, ノードに radvd をインストール

```
# sudo apt-get install -y radvd  
# sudo bash -c "echo 1 > /proc/sys/net/ipv6/conf/all/forwarding"  
# sudo bash -c "echo 0 > /proc/sys/net/ipv6/conf/all/accept_ra"
```



## 設定例(2/4)

- (3) 作成する nova.conf file の各ノードで --use\_ipv6 flag を true にセット
- (4) 各 nova-services をリスタート
- (5) nova-manage network create コマンドを使用  
IPv6アドレスに一定の範囲を加えることが可能

```
nova-manage network create fixed_range num_networks network_size  
[vlan_start] [vpn_start] [fixed_range_v6]
```

# 設定例(3/4)

## <注釈>

- (1) fixed\_range\_v6パラメータを用いて IPv6 global routing prefix をセット可能
  - デフォルト値は : fd00:/48
- (2) FlatDHCPManagerを使う際, コマンドは fixed\_range\_v6 の初期値を使用
- (3) VlanManagerを使う際, コマンドはサブネット id を増加させることでサブネットの prefix を作成
- (4) ゲスト VMs はこの prefix を使用し, IPv6 global unicast address を生成

# 設定例(4/4)

## (6) VlanManager を設定

```
nova-manage network create 10.0.1.0/24 3 32 100 1000 fd00:1::/48
```

## (7) FlatDHCPManager を設定

```
nova-manage network create 10.0.2.0/24 3 32 0 0 fd00:1::/48
```

### <注釈>

[vlan\_start]と[vpn\_start]パラメータは FlatDHCPManager  
により用いられない

# IPv6のアドレス使用の設定フラグ

以下のフラグを nova.conf に記述

フラグ	デフォルト	説明
--use_ipv6	'false'	1をセットするか, Trueにする; IPv6ネットワーク・アドレスを使用するべきかどうか決定
--flat_injected	'false'	Cactusのみ: このオプションはCompute (nova) がゲストの /etc/network/interfaces を ipv6 構成に書き換えるかどうかを示す 現在のDebian-baseのシステム上でのみ働く

## 4.5.Image Service と Storage for Computeの設定(1/7)

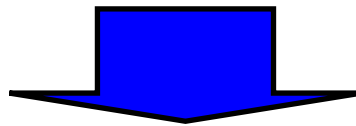
Image の格納/読み出しをするためには、以下の2つの方法がある

- (1) local image storage system の使用
- (2) Glance の使用

<Glance を使用するための nova-compute 構成の変更>

--image\_service flag を

nova.image.glance.GlanceImageService に変更



OpenStack Compute 用のイメージの格納・読み出しに  
Glance を使用可能

## 4.5.Image Service と Storage for Computeの設定(2/7)

### <Glance Image Service と Storage の設定>

以下のフラグをnova.confに記述

フラグ	デフォルト	説明
--glance_host=GLANCE_SERVER_IP	'\$my_ip'	IPアドレス; Glanceサーバのインストール先を識別
--glance_host	'\$my_ip'	IPアドレス; OpenStack Compute イメージ・サービス・サーバ(プロジェクトの名前は Glance)のIPアドレス
--glance_port	'9292'	ポートの値; OpenStack Compute イメージ・サービス・サーバ(プロジェクトの名前は Glance)のために, 開かれたポート
--s3_dmz	'\$my_ip'	IPアドレス; instance 内部のIP (DMZはDeMilitarized Zoneの短縮形)

## 4.5.Image Service と Storage for Computeの設定(3/7)

フラグ	デフォルト	説明
--image_service	'nova.image.local. LocalImageService'	<p>イメージを格納・取り出しのためのサービス. イメージは euca2ools を使って別途登録する必要がある. オプション:</p> <ul style="list-style-type: none"><li>• nova.image.s3.S3ImageService Image Service のためのS3バックエンド.</li><li>• nova.image.local.LocalImageService ローカル・ディスクにイメージを保存するImage Service. image_idsが整数であると仮定. イメージ・マネージャがここで定められないならば, デフォルト・セッティングとなる.</li><li>• nova.image.glance.GlanceImageService イメージを格納・読み出しするためのバックエンド;</li></ul> <p>詳細は<a href="http://glance.openstack.org">http://glance.openstack.org</a></p>

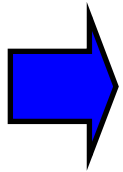
## 4.5.Image Service と Storage for Computeの設定(4/7)

フラグ	デフォルト	説明
--s3_host	'\$my_ip'	IPアドレス; 基盤となるS3ホストのIPアドレス. OpenStack Computeがobjectstoreサービス(バーチャル・マシン・イメージと bucket を含む)がホストされている場所
--s3_port	'3333'	整数値; S3ホストが動作しているポート
--use_s3	'ture'	1をセットするか, Tureにする; s3からイメージを得るべきであるか, ローカル・コピーを使うべきかどうか決定



## 4.5.Image Service と Storage for Computeの設定(5/7)

Image Service のために Glance を使用しない場合



object store を使用する

<object store>

特定の場所でイメージを保持

サーバー・ローカル上の nova.conf ファイルへの state path

イメージを保存するために, S3 bucket のセットを使うことも可能

## 4.5.Image Service と Storage for Computeの設定(6/7)

### <Local image storage の設定>

以下のフラグを nova.conf に記述

フラグ	デフォルト	説明
--state_path	'/Users/username/p/nova/nova/../'	ディレクトリパス;nova's state を保持するファイルのトップレベルディレクトリ
--buckets_path	'\$state_path/buckets'	ディレクトリパス;S3スタイルの bucket のために作成されるディレクトリ
--image_path	'\$state_path/images'	ディレクトリパス; object store を使うとき, イメージを保存するディレクトリ

## 4.5.Image Service と Storage for Computeの設定(7/7)

フラグ	デフォルト	説明
--image_service	default: 'nova.image.local.LocalImageService'	<p>イメージを取り出したり, 検索するためのサービス. イメージは euca2oolsを使って別途に登録しておく必要がある.</p> <p>オプション;</p> <ul style="list-style-type: none"><li>・nova.image.s3.S3ImageService</li></ul> <p>Image ServiceのためのS3バックエンド; Cactusでは, S3 image service はEC2 API の使用のために他の Image Service を覆い隠す. フラグをセットせずとも, EC2 APIは常にデフォルトで S3 imege service を利用</p> <ul style="list-style-type: none"><li>・nova.image.local.LocalImageService</li></ul> <p>ローカル・ディスク に イメージ を保存している Image Service. image_ids が整数であると仮定</p> <ul style="list-style-type: none"><li>・nova.image.glance.GlanceImageService</li></ul> <p>イメージを格納・読み出しするための Glance バックエンド; 詳細は <a href="http://glance.openstack.org">http://glance.openstack.org</a></p>

## 4.6.Live Migration の設定

### <Live Migration機能>

マシンの動作を維持しながら，以下の2つを実行する場合に役立つ

- (1) hyper visors/BIOS をアップグレード
- (2) hyper visors/BIOS にパッチをインストール

- 多くのVMインスタンスが特定の物理的マシンで走っているとき high load を再配分可能
- VM instances が散在しているとき，より論理的に配置するためにVM instances を物理的なマシンに移動可能

### <具体例>

- (1) HDD volumes RAID の 1つ，または NIC の1つが故障した場合
- (2) 定期的なメンテナンスのため VM instances を migrate する際

# 動作環境(1/2)

## <OS>

Uboutu 10.04/10.10 :instance と host で使用

## <共有保管>

NOVA-INST-DIR/instances/

共有記憶装置(NFSを使ってテストされる)で開始される

## <instances>

ISCSI/AoE base の volumes で移動可能

## <Hypervisor>

libvirt による KVM

# 動作環境(2/2)

## <メモ1>

- (1) vmイメージは, 「nova-INST-DIR/instance」に置かれると想定される
- (2) デフォルト値は nova.compute.managerの「flags.instances\_path」にある

## <メモ2>

- (1) nova-manage が必要なため, この機能は管理者権限のみで使用となる

# Nova インストールの前準備

- (1) 最低でも3つのホストサーバー(ここではHostA, HostB, HostC)を準備
  - (A) nova-api/nova-network/nova-volume/nova-objectstore/nova-scheduler(また, その他のデーモン) はHostA上で動作
  - (B) nova-computeはHostBとHostC上で動作
- (2) HostAはNOVA-INST-DIR/instanceをエクスポートし, HostB と HostCはこれをマウント
  - (A) 混乱を避けるため, 「NOVA-INST-DIR」はHostA/HostB/HostCと同じである(「NOVA-INST-DIR」はインストールディレクトリのトップを示す)

# 初期構成の手順(1/8)

## (1) /etc/hosts を構成する

- (A) 3つのホストが互いの名前で解決できるか確認  
それぞれ互いに ping するのは良い方法である

```
# ping HostA  
# ping HostB  
# ping HostC
```



# 初期構成の手順(2/8)

## (2) HostAのNFSを構成

(A) /etc/exports に下記を追加

```
NOVA-INST-DIR/instances HostA/255.255.0.0(rw, sync, fsid=0, no_root_squash
```

(B) "255.255.0.0"を適当なネットマスクに変更  
これは HostB/HostC を含まなければならない

```
# /etc/init.d/nfs-kernel-server restart
```

(C) NFSサーバーを再起動

```
# /etc/init.d/idmapd restart
```

# 初期構成の手順(3/8)

## (3) HostBのNFSを構成

(A) /etc/fstabに下記を追加

```
HostA:/ DIR nfs4 defaults 0 0
```

(B) マウント

エクスポートされたディレクトリがマウント可能か確認

```
# mount -a -v
```

失敗するならば, 全てのホストで以下のコマンドをためす

```
# iptables -F
```

# 初期構成の手順(4/8)

(C) file/daemon の権限を確認する

全てのnovaデーモンはrootとして動作していると想定される

```
# ps -ef | grep nova
root 5948 5904 9 11:29 pts/4 00:00:00 python /opt/nova-2010.4//bin/nova-api
root 5952 5908 6 11:29 pts/5 00:00:00 python /opt/nova-2010.4//bin/nova-
objectstore
... (snip)
```

# 初期構成の手順(5/8)

(D) "NOVA-INST-DIR/instances/"ディレクトリの権限を確認  
HostA から見るか確認する

```
# ls -ld NOVA-INST-DIR/instances/  
drwxr-xr-x 2 root root 4096 2010-12-07 14:34 nova-install-dir/instances/
```

# 初期構成の手順(6/8)

(E) HostB と HostC も同様に確認する

```
# ls -ld NOVA-INST-DIR/instances/
drwxr-xr-x 2 root root 4096 2010-12-07 14:34 nova-install-dir/instances/

# df -k
Filesystem            1K-blocks    Used Available Use% Mounted on
/dev/sda1             921514972  4180880 870523828   1% /
none                  16498340    1228 16497112   1% /dev
none                  16502856         0 16502856   0% /dev/shm
none                  16502856    368 16502488   1% /var/run
none                  16502856         0 16502856   0% /var/lock
none                  16502856         0 16502856   0% /lib/init/rw
HostA: 921515008 101921792 772783104 12% /opt ( <--- this line is
important.)
```

# 初期構成の手順(7/8)

## (4) Libvirt を構成する

(A) /etc/libvirt/libvirt.conf を以下のように修正する

before : #listen\_tls = 0

after : listen\_tls = 0

before : #listen\_tcp = 1

after : listen\_tcp = 1

add: auth\_tcp = "none"

# 初期構成の手順(8/8)

(B) /etc/init/libvirt-bin.conf を修正

```
before : exec /usr/sbin/libvirtd -d  
after  : exec /usr/sbin/libvirtd -d -l
```

(C) /etc/default/libvirt-bin を修正

```
before :libvirtd_opts="-d"  
after  :libvirtd_opts="-d -l"
```

(D) libvirtを再起動. libvirt が再起動されることを確認

```
# stop libvirt-bin && start libvirt-bin  
# ps -ef | grep libvirt  
root 1145 1 0 Nov27 ? 00:00:03 /usr/sbin/libvirtd -d
```

# Live Migration の設定変更(1/2)

通常以下のフラグを使用する必要はない  
カスタマイズする場合のみ, 以下のフラグをnova.conf に記述

フラグ	デフォルト	説明
--live_migration_retry_count	30	live_migration のリトライ回数. リトライには1秒のインターバルがある
--live_migration_uri	'qemu+tcp://%s/system'	live_migration機能により使用されるプロトコルを定める. qemu+sshを使用する場合, <a href="http://libvirt.org/">http://libvirt.org/</a> で解説されるように変更



# Live Migration の設定変更(2/2)

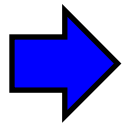
フラグ	デフォルト	説明
--live_migration_ bandwidth	0	live migration によって使われるネットワーク帯域を定める.
--live_migration_flag	'VIR_MIGRATE_UNDEFINE_ SOURCE, VIR_MIGRATE_PEER2PEER'	live migration のための libvirt フラグを定義する.

## 4.7.データベースコネクションの構成

### <データベースコネクション>

データベース名は『nova』で、このエントリの大部分はnova-scheduler サービスによって書かれる

OpenStack Compute をどのような SQLAlchemy 互換のデータベースでも使えるように構成可能



すべてのサービスがデータベースのエントリを更新可能である必要

novaデータベースのためのコネクションを構成するために、以下のセッティングを使用

# nova データベースのための コネクションストリング構成

以下のフラグを nova.conf に記述

フラグ	デフォルト	説明
--sql_connection	'sqlite:///state_path/ nova.sqlite'	IPアドレス; OpenStack Compute SQLデータベースのロケーション
--sql_idle_timeout	'3600'	
--sql_max_retries	'12'	整数値; SQL接続を使用するアク セスの数
--sql_retry_interval	'10'	整数値; SQL接続のリトライイン ターバル
--db_backend	'sqlalchemy'	バックエンドは、データベース接続 のために選ばれる
--db_driver	'nova.db.api'	データベース・アクセスの用途への ドライバ

## 4.8.Compute メッセージシステム の設定(1/5)

OpenStack Compute はメッセージシステムに AMQP を使用

### <AMQP>

オープンでスタンダードなメッセージ発信用ミドルウェア

RabbitMQ が AMQP を可能にすることでnova-service が互いに  
通信可能になる

## 4.8.Compute メッセージシステム の設定(2/5)

メッセージコミュニケーションを以下の2つのために設定可能

- (1) 異なるインストールシナリオ
- (2) RabbitMQ のリトライ回数とRPC スレッドプールのサイズの調整

メッセージシステムに関して、以下の3つを設定可能

- (1) Remote Procedure Calls と RabbitMQ Messaging
- (2) RabbitMQ Messaging の調整
- (3) Exchange と Topic Names の変更

これらの設定のため以下のフラグを nova.conf に記述

## 4.8.Compute メッセージシステム の設定(3/5)

### <Remote Procedure Calls と RabbitMQ Messaging の設定>

フラグ	デフォルト	説明
--rabbit_host	'localhost'	IPアドレス; RabbitMQのインストール先
--rabbit_password	'guest'	文字列; RabbitMQサーバーのパスワード
--rabbit_port	'5672'	整数値;RabbitMQ serverが 実行/入力待ちしているポート
--rabbit_userid	'guest'	文字列; ラビット接続のために使われるユーザーID
--rabbit_virtual_host	'/'	仮想RabbitMQインストールの位置

## 4.8.Compute メッセージシステム の設定(4/5)

### <RabbitMQ Messaging の調整設定 >

フラグ	デフォルト	説明
--rabbit_max_retries	'12'	整数値; RabbitMQ connection 最大接続回数
--rabbit-retry-interval	'10'	整数値; RabbitMQ connection のリトライインターバルの時間
--rpc_thread_pool_size	'1024'	整数値; Remote Procedure Callsのスレッドプールのサイズ

## 4.8.Compute メッセージシステム の設定(5/5)

### <Exchange と Topic Names の変更の設定>

フラグ	デフォルト	説明
--control_exchange	'nova'	文字列; 接続するメイン exchange の名前
--ajax_console_proxy_topic	'ajax_proxy'	文字列; ajaxプロキシノードが監視するトピック
--console_topic	'console'	文字列; プロキシノードが監視するコンソールのトピック
--network_topic	'network'	文字列; ノードが監視するネットワークのトピック
--scheduler_topic	'scheduler'	文字列; ノードが監視するスケジューラのトピック
--volume_topic	'volume'	文字列; ボリュームノードが監視するトピックの名前



## 4.9.認証と許可の設定(1/8)

OpenStack Computeは、以下のどちらかを実装する

- (1) Active Directory や LDAP のようなユーザ管理システムをバックエンドに持つ認証システム
- (2) SAMLポリシーコントローラをグループにマッピングできるようなシステム

### <注釈>

- (1) プロジェクトのために role をカスタマイズすることが可能
- (2) API呼び出しのための証明書は、プロジェクトzipファイルに保存される
- (3) CA (Certificate Authority) は、nova.conf でカスタマイズされる

## 4.9.認証と許可の設定(2/8)

認証と許可に関して、以下の4つを設定可能

- (1) 認証
- (2) role
- (3) credentials
- (4) CA

これらの設定のため以下のフラグを nova.conf に記述

## 4.9.認証と許可の設定(3/8)

### <認証の設定>

フラグ	デフォルト	説明
--auth_driver	default:'nova.auth.dbdriver.DbDriver'	<p>文字列; 認証のためのドライバーの名称</p> <ul style="list-style-type: none"><li>・nova.auth.dbdriver.DbDriver</li><li>- デフォルト設定</li><li>・nova.auth.Idapdriver.FakeLdapDriver</li><li>-別のクラス名でこのドライバと同じメソッドを持つドライバを外から見れるように置くことで、他のバックエンドをサポートするこのドライバの置き換えを作成可能</li></ul>

## 4.9. 認証と許可の設定(4/8)

### <role の設定>

フラグ	デフォルト	説明
--allowed_roles	'cloudadmin,itsec, sysadmin,netadmin,developer'	コンマ区切りのリスト; プロジェクトに許された role
--global_roles	'cloudadmin,itsec'	コンマ区切りのリスト; すべてのプロジェクトにあてはまる role
--superuser_roles	'cloudadmin'	コンマ区切りのリスト; チェックしている認可を完全に無視する role

## 4.9.認証と許可の設定(5/8)

### <credentials の設定>

フラグ	デフォルト	説明
--credentials_template	'/Users/termie/p/ nova/nova/auth/novarc. template'	ディレクトリ; ユーザのRCファイルの 作成のテンプレート
--credential_rc_file	'%src'	ファイル名; zip証明書のrcのファイ ル名
--credential_cert_file	'cert.pem'	ファイル名; zip証明書で証明される File名

## 4.9. 認証と許可の設定(6/8)

フラグ	デフォルト	説明
--credential_ key_file	'pk.pem'	ファイル名; zip証明書のrcのファイル名
--vpn_client_ template	'nova/cloudpipe/ client/ovpn.template'	ディレクトリ;ユーザーvpnファイルの作成のテンプレートファイルのある場所を示す
--credential_ vpn_file	'nova-vpn.conf'	ファイル名; zip証明書で証明されるファイル名

## 4.9. 認証と許可の設定(7/8)

### <CAの設定>

フラグ	デフォルト	説明
--keys_path	'\$state_path/keys'	ディレクトリ; novaがキーを保存する場所
--ca_file	'cacert.pem'	ファイル名; ルートCAのファイル名
--crl_file	'crl.pem'	ファイル名; 証明書失効リストのファイル名
--key_file	'private/cakey.pem'	ファイル名; プライベートキーのファイル名
--use_project_ca	'false'	True or false; 各プロジェクトのためにCAを使うべきかどうか示す; false だった場合のCAは各プロジェクトに使用されない

## 4.9. 認証と許可の設定(8/8)

フラグ	デフォルト	説明
--project_cert _subject	'/C=US/ST=California/L= MountainView/O=AnsoLabs/OU= NovaDev/CN=project-ca-%s-%s'	文字列; プロジェクトの証明書, プロジェクトの%s, タイムスタンプのためのサブジェクト
--user_cert _subject	'/C=US/ST=California/L= MountainView/O=AnsoLabs/OU= NovaDev/CN=%s-%s-%s'	文字列; ユーザーの証明書, プロジェクトの%s, ユーザー, タイムスタンプのためのサブジェクト
--vpn_cert _subject	'/C=US/ST=California/L= MountainView/O=AnsoLabs/OU= NovaDev/CN=project-vpn-%s-%s'	文字列; vpnsの証明書, プロジェクトの%s, タイムスタンプのためのサブジェクト

※ %s の置き換えられ方が, プロジェクト, ユーザ, タイムスタンプである