

7.System Administration

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

(1) OpenStack Computeの役割

(2) Computeの管理

(A) Instanceの起動

(B) Instanceの終了

(C) カスタムImageの作成

(D) Cloudの管理

(E) Compute Userの管理

(F) Volumeの管理

目次

(1) OpenStack Computeの役割

(2) Computeの管理

(A) Instanceの起動

(B) Instanceの終了

(C) カスタムImageの作成

(D) Cloudの管理

(E) Compute Userの管理

(F) Volumeの管理

Computeの役割(1/2)

(1) API Server

- (A) ハイパバイザとストレージ, ネットワークを制御するコマンドを提供
- (B) エンドポイントはhttpウェブサービスとして提供

(2) Message Queue

- (A) Compute内の各ノードの通信を行う方法
- (B) 通常, APIからのリクエストがMessage送信を開始
- (C) 受け取ったMessageを認証し, Userがコマンドを実行可能か確認
- (D) 各Workerのrole毎にキューイング

Computeの役割(2/2)

(3) Compute Worker

- (A) Instanceの起動
- (B) Instanceの終了
- (C) Instanceの再起動
- (D) Volumeの取り付け
- (E) Volumeの取り外し
- (F) コンソール出力の入手

(4) Network Controller

- (A) IPアドレスの割り当て
- (B) プロジェクトのためのVLANの設定
- (C) Computeノードのためのネットワークの設定

(5) Volume Workers

- (A) Volumeの作成
- (B) Volumeの削除
- (C) Volumeの確立

目次

(1) OpenStack Computeの役割

(2) Computeの管理

(A) Instanceの起動

(B) Instanceの終了

(C) カスタムImageの作成

(D) Cloudの管理

(E) Compute Userの管理

(F) Volumeの管理

Instanceの起動(1/2)

(1) Imageのダウンロードと登録

```
image="ubuntu1010-UEC-localuser-image.tar.gz"  
$ wget http://c0179148.cdn1.cloudfiles.rackspacecloud.com/ubuntu1010-UEC-localuser-image.tar.gz  
$ uec-publish-tarball $image [bucket-name] [hardware-arch]
```

<実際の例>

```
$ uec-publish-tarball ubuntu1010-UEC-localuser-image.tar.gz  
dub-bucket x86_64
```

Instanceの起動(2/2)

(2) 登録されているImageの表示

```
$ euca-describe-images
```

(3) key-pairの作成

```
$ euca-add-keypair test > test.pem
```

```
$ chmod 600 test.pem
```

(4) Instanceの起動

```
$ euca-run-instances -k test -t m1.tiny ami-zqkyh9th
```

(5) Instanceに接続

```
$ ssh ubuntu@$ipaddress
```

(6) sudo権限の確認

```
$ sudo -i
```


Instanceの終了

(1) Instanceの終了

```
$ euca-terminate-instances $instanceid
```

カスタムImageの作成

下記のURLから作成ツールを入手可能

<https://code.launchpad.net/~smoser/+junk/ttylinux-uec>

下記のURLからドキュメントを入手可能

<https://help.ubuntu.com/community/UEC/CreateYourImage>

Cloudの管理

(1) 以下のコマンドで管理可能

```
$ nova-manage category command [args]
```

<[args]に与える引数>

(1) user

(2) project

(3) role

(4) shell

(5) vpn

(6) floating

Compute Usersの管理(1/2)

(1) Userの追加

```
$ nova-manage user create john my-access-key a-super-secret-key
```

<Imageを登録する必要がない場合>

(2-A) Projectの追加と設定ファイルの適用

```
$ nova-manage project environment john_project john  
$ source novarc
```

<Imageを登録する必要がある場合>

(2-B) Projectの追加と設定ファイルの適用

```
$ nova-manage project zipfile john_project john  
$ unzip nova.zip  
$ source novarc
```

Compute Usersの管理(2/2)

(3) アクセス権限の設定

```
$ nova-manage role add john netadmin  
$ nova-manage role add john netadmin john_project
```

Volumesの管理(1/3)

(1) 必要なパッケージをインストール

```
$ apt-get install lvm2 nova-volume
```

(2) パーティションを操作するための確認

```
$ partprobe
```

(3) パーティションを状態を確認

```
$ fdisk -l
```

Volumesの管理(1/3)

(1) 必要なパッケージをインストール

```
$ apt-get install lvm2 nova-volume
```

(2) パーティションを操作するための確認

```
$ partprobe
```

(3) パーティションを状態を確認

```
$ fdisk -l
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	12158	97280	83	Linux
/dev/sda2		12158	24316	97655808	83	Linux
/dev/sda3		24316	24328	97654784	83	Linux
/dev/sda4		24328	42443	145507329	5	Extended
/dev/sda5		24328	32352	64452608	8e	Linux LVM
/dev/sda6		32352	40497	65428480	8e	Linux LVM
/dev/sda7		40498	42443	15624192	82	Linux swap / Solaris

Volumesの管理(2/3)

(4) physical volumeとvolume groupの作成

```
$ pvcreate /dev/sda5  
$ vgcreate nova-volumes /dev/sda5
```

(5) iscsiターゲットの設定と起動

```
$ sed -i 's/false/true/g' /etc/default/iscsitarget  
$ service iscsitarget start
```

(6) nova.confの設定

–iscsi_ip_prefix=フラグを設定

(7) nova-volumeの起動

```
$ service nova-volume start
```


Volumesの管理(3/3)

(8) 管理しているvolumeを表示

```
$ nova-manage service list
```

(9) volumeの作成

```
$ euca-create-volume -s 7 -z nova
```

```
// -sオプションはサイズ(GB)
```

```
// -zオプションは初期設定の領域(通常はnovaを指定)
```

Volumesの管理(3/3)

(8) 管理しているvolumeを表示

```
$ nova-manage service list
```

(9) volumeの作成

```
$ euca-create-volume -s 7 -z nova  
// -sオプションはサイズ(GB)  
// -zオプションは初期設定の領域(通常はnovaを指定)  
VOLUME vol-0000000b 7 creating (wayne, None, None,  
None) 2011-02-11 06:58:46.941818
```

Volumesの管理(3/3)

(8) 管理しているvolumeを表示

```
$ nova-manage service list
```

(9) volumeの作成

```
$ euca-create-volume -s 7 -z nova  
// -sオプションはサイズ(GB)  
// -zオプションは初期設定の領域(通常はnovaを指定)  
VOLUME vol-0000000b 7 creating (wayne, None, None,  
None) 2011-02-11 06:58:46.941818
```

(10) volumeの取り付け

```
$ euca-attach-volume vol-00000009 -i i-00000008 -d /dev/vdb
```

Live Migrationの使用

<Live Migration>

走行中のインスタンスをあるOpenStack Computeサーバから別のOpenStack Computeサーバへ移動させる事

<Live Migrationの使用手順>

- (1) 特定のサーバでインスタンスが走行しているか確認
- (2) インスタンスをMigration可能なサーバを確認
- (3) Migration先のHostにMigrationのための十分な資源が存在するか否か確認
- (4) Live Migrationの実行

<想定環境>

- (1) HostA, HostB, HostCでOpenStackが走行中
- (2) HostBからHostCにマイグレーションを実行

Live Migrationの使用手順(1/4)

(1) HostBでインスタンスが走行しているか確認

```
# euca-describe-instance
```

Live Migrationの使用手順(1/4)

(1) HostBでインスタンスが走行しているか確認

```
# euca-describe-instance
Reservation:r-2raqmabo
RESERVATION r-2raqmabo admin default
INSTANCE    i-00000003 ami-ubuntu-lucid a.b.c.d e.f.g.h
running    testkey (admin, HostB) 0      m1.small  2011-02-
15 07:28:32 nova
```

Live Migrationの使用手順(1/4)

(1) HostBでインスタンスが走行しているか確認

```
# euca-describe-instance
Reservation:r-2raqmabo
RESERVATION r-2raqmabo admin default
INSTANCE i-000000003 ami-ubuntu-lucid a.b.c.d e.f.g.h
running testkey (admin, HostB) 0 m1.small 2011-02-
15 07:28:32 nova
```



i-000000003インスタンスが走行中

Live Migrationの使用手順(2/4)

(2) インスタンスをMigration可能なサーバを確認

```
# nova-manage service list
```


Live Migrationの使用手順(2/4)

(2) インスタンスをMigration可能なサーバを確認

```
# nova-manage service list  
HostA nova-scheduler enabled :-) None  
HostA nova-volume enabled :-) None  
HostA nova-network enabled :-) None  
HostB nova-compute enabled :-) None  
HostC nova-compute enabled :-) None
```

Live Migrationの使用手順(2/4)

(2) インスタンスをMigration可能なサーバを確認

```
# nova-manage service list  
HostA nova-scheduler enabled :-) None  
HostA nova-volume enabled :-) None  
HostA nova-network enabled :-) None  
HostB nova-compute enabled :-) None  
HostC nova-compute enabled :-) None
```



nova-computeが起動しているため, HostCにマイグレーションが可能

Live Migrationの使用手順(3/4)

- (3) HostCでLive Migrationのための十分な資源が存在するか確認
 - (A) update_resourceコマンドを使用して, HostCで使用している資源(used)の更新を実行

```
# nova-manage service update_resource HostC
```

Live Migrationの使用手順(3/4)

- (3) HostCでLive Migrationのための十分な資源が存在するか確認
(B) HostCで使用している資源の状態を表示

```
# nova-manage service update_resource HostC  
# nova-manage service describe_resource HostC
```

Live Migrationの使用手順(3/4)

- (3) HostCでLive Migrationのための十分な資源が存在するか確認
(B) HostCで使用している資源の状態を表示

```
# nova-manage service update_resource HostC
# nova-manage service describe_resource HostC
HOST          PROJECT    cpu  mem(mb) disk(gb)
HostC(total)           16   32232   878
HostC(used)            13   21284   442
HostC             p1    5   10240   150
HostC             p2    5   10240   150
.....
```

Live Migrationの使用手順(3/4)

(3) HostCでLive Migrationのための十分な資源が存在するか確認

(B) HostCで使用している資源の状態を表示

```
# nova-manage service update_resource HostC
# nova-manage service describe_resource HostC
HOST          PROJECT  cpu  mem(mb) disk(gb)
HostC(total)           16  32232    878
HostC(used)            13  21284    442
HostC             p1    5  10240    150
HostC             p2    5  10240    150
.....
```

<表示の説明>

cpu: cpuの数

Live Migrationの使用手順(3/4)

- (3) HostCでLive Migrationのための十分な資源が存在するか確認
(B) HostCで使用している資源の状態を表示

```
# nova-manage service update_resource HostC
# nova-manage service describe_resource HostC
HOST          PROJECT  cpu  mem(mb) disk(gb)
HostC(total)           16   32232    878
HostC(used)           13   21284    442
HostC           p1      5   10240    150
HostC           p2      5   10240    150
.....
```

<表示の説明>

mem(mb): メモリの総量(MB)

Live Migrationの使用手順(3/4)

- (3) HostCでLive Migrationのための十分な資源が存在するか確認
(B) HostCで使用している資源の状態を表示

```
# nova-manage service update_resource HostC
# nova-manage service describe_resource HostC
HOST          PROJECT  cpu  mem(mb) disk(gb)
HostC(total)           16  32232   878
HostC(used)           13  21284   442
HostC           p1      5  10240   150
HostC           p2      5  10240   150
.....
```

<表示の説明>

disk(gb): NOVA-INST-DIR/instances(GB)の総量

Live Migrationの使用手順(3/4)

(3) HostCでLive Migrationのための十分な資源が存在するか確認

(B) HostCで使用している資源の状態を表示

```
# nova-manage service update_resource HostC
# nova-manage service describe_resource HostC
HOST          PROJECT  cpu  mem(mb) disk(gb)
HostC(total)      16   32232   878
HostC(used)       13   21284   442
HostC            p1    5   10240   150
HostC            p2    5   10240   150
.....
```

<表示の説明>

1行目: 物理サーバの資源総量

Live Migrationの使用手順(3/4)

(3) HostCでLive Migrationのための十分な資源が存在するか確認

(B) HostCで使用している資源の状態を表示

```
# nova-manage service update_resource HostC
# nova-manage service describe_resource HostC
HOST          PROJECT  cpu  mem(mb) disk(gb)
HostC(total)           16  32232   878
HostC(used)           13  21284   442
HostC           p1      5  10240   150
HostC           p2      5  10240   150
.....
```

<表示の説明>

2行目: 現在使用している資源

Live Migrationの使用手順(3/4)

(3) HostCでLive Migrationのための十分な資源が存在するか確認

(B) HostCで使用している資源の状態を表示

```
# nova-manage service update_resource HostC
# nova-manage service describe_resource HostC
HOST          PROJECT  cpu  mem(mb) disk(gb)
HostC(total)           16  32232   878
HostC(used)           13  21284   442
HostC           p1      5  10240   150
HostC           p2      5  10240   150
.....
```

<表示の説明>

3行目 : project毎の使用資源

Live Migrationの使用手順(4/4)

(4) Live migrationの実行

```
# nova-manage vm live_migration i-000000003 HostC
```

Live Migrationの使用手順(4/4)

(4) Live migrationの実行

```
# nova-manage vm live_migration i-000000003 HostC  
Migration of i-000000001 initiated. Check its progress using euca-  
describe-instances.
```

<注意>

- (1) Migrationが成功したか否かは, euca-describe-instance
コマンドを実行して確認可能
- (2) 依然としてHostBでインスタンスが走行中であれば,
ログファイル(src/dest nova-computeとnova-scheduler)を確認

nova.confのフラグ(1/18)

利用可能なフラグの一覧は、以下のコマンドを実行して取得可能

```
$ /bin/nova-<servicename> --help
```

＜nova-api, nova-computeに関するフラグ＞

設定項目	初期値	説明
--ajax_console_proxy_port	8000	ajaxコンソールプロキシサーバのポート番号.
--ajax_console_proxy_topic	ajax_proxy	ajaxプロキシノードのテーマ.
--ajax_console_proxy_url	http://127.0.0.1:8000	ajaxコンソールプロキシのIPアドレスとポート番号.
--auth_token_ttl	3600	認証トークンの残存する時間(秒).
--aws_access_key_id	admin	awsに接続するID(ユーザ名).
--aws_secret_access_key	admin	awsに接続するIDとペアになっている秘密鍵.

nova.confのフラグ(2/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--compute_manager	nova.compute.manager.ComputeManager	インスタンスを作成するためのRPCをハンドルするComputeのためのマネージャ.
--compute_topic	compute	LISTENするcomputeノードのテーマ名.
--connection_type	libvirt	インスタンスを生成するための仮想化ドライバ. libvirt, xenapi, fakeを指定.
--console_manager	nova.console.manager.ConsoleProxyManager	コンソールプロキシのためのマネージャ.
--console_topic	console	プロキシノードがLISTENするコンソールのテーマ.
--control_exchange	nova	接続するためのmain exchangeの名前.

nova.confのフラグ(3/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--default_image	ami-11111	使用するImageの初期設定の名前. テスト目的のみ使用.
--db_backend	sqlalchemy	データベースに接続するためのバックエンドの選択.
--db_driver	nova.db.api	データベースに接続するためのドライバ.
--default_instance_type	m1.small	Instanceタイプの初期設定の名前. テスト目的のみ使用.
--default_log_levels	amqpplib=WARN,sqlalchemy=WARN,eventlet.wsgi.server=WARN	ロガーの名前とログレベルのペア. "ロガーの名前=レベル"の組みで記述.
--default_project	openstack	openstackのための初期設定のプロジェクト名.

nova.confのフラグ(4/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--ec2_dmz_host	\$my_ip	APIサーバの内部のIPアドレス. DMZ(DeMilitarized Zone).
--ec2_host	\$my_ip	外部から見えるAPIサーバのIPアドレス. (External-facing IP)
--ec2_path	/services/Cloud	nova-apiが存在するEC2スタイルURLの接尾辞.
--ec2_port	8773	(nova-apiが存在する)Cloud Controllerのポート番号.
--ec2_scheme	http	nova-apiが存在するEC2スタイルURLの接頭辞.
--ec2_url	none	nova-apiが存在するURL. 例: http://184.106.239.134:8773 /services/Cloud

nova.confのフラグ(5/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--flat_injected	false	NovaがゲストにIPv6ネットワーク設定の情報を割り込みするか否か. Debianのみ可能.
--fixed_ip_disassociate_timeout	600	IP割り当てが解除された後に、切断するまでの時間(秒).
--fixed_range	10.0.0.0/8	iptablesの規則によって作られるIPブロックの設定.
--fixed_range_v6	fd00::/48	IPv6ブロックの設定.
--[no]flat_injected	true	ネットワークの設定をゲストに割り込むか否か. Debianのみ可能.
--flat_interface	br100	設定されたインタフェースにFlatDhcpがブリッジ接続する.

nova.confのフラグ(6/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--flat_network_bridge	br100	簡易なネットワークのためのブリッジ.
--flat_network_dhcp_start	10.0.0.2	flatDhcpを使用した場合に, DHCPサーバに最初に割り当てるIPアドレス.
--flat_network_dns	8.8.4.4	簡易なネットワークのためのDNS.
--floating_range	4.4.4.0/24	フローティングIPアドレスブロック.
--[no]fake_network	false	Novaが擬似ネットワークデバイスとアドレスを使用するか否か.
--[no]enable_new_services	true	nova-manageを使用してサービスを作成した時に, 利用可能なプールにサービスを追加する.

nova.confのフラグ(7/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--[no]fake_rabbit	false	Novaが擬似rabbitサーバを使用するか否か.
--glance_host	\$my_ip	GlanceのIPアドレス.
--glance_port	9292	Glanceの開放ポート.
-,--[no]help		このHELPを表示する.
--[no]helpshort		このモジュールだけのHELPを表示する.
--[no]helpxml		textの代わりにxmlでHELPを出力する.
--host		Cloud Controllerを提供しているノードの名前.

nova.confのフラグ(8/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--image_service	nova.image.s3.S3ImageService	<p>このサービスは, Imageの読み出しと検索を行う. Imageの登録はeuca2oolsを使用する.</p> <p>オプション:</p> <ul style="list-style-type: none">(1) nova.image.s3.S3ImageService S3バックグラウンド(2) nova.image.local.LocalImageService ローカルディスクに保存する. 何も設定されていないならば, この設定になる.(3) nova.image.glance.GlanceImageService Glanceバックグラウンド

nova.confのフラグ(9/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--instance_name_template	instance-%08x	一般的なインスタンスに使用する名前のテンプレート.
--libvirt_type	kvm	libvirtを通してハイパバイザに接続するための名前. kvm,qemu,uml,xenが使用可能.
--lock_path	none	lockファイルを保存するための書き込み可能なパス.
--logfile	none	ログファイルの名前.
--logging_context_format_string	%(asctime)s %(levelname)s %(name)s [%(request_id)s %(user)s %(project)s] %(message)s	ログメッセージに出力する文字列のフォーマット.

nova.confのフラグ(10/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--logging_debug_format_suffix	%(processName)s (pid=%(process)d) %(funcName)s %(pathname)s:%(lineno)d	ログレベルがDEBUGの時に ログメッセージに加えるデータ.
--logging_defalut_format_string	%(asctime)s %(levelname)s %(name)s [-] %(message)s	文脈なしに出力するログメッセージのフォーマット.
--logging_exception_prefix	(%(name)s): TRACE	フォーマットから接頭辞を除いたものを出力.
--my_ip		Cloud controllerのIPアドレス.

nova.confのフラグ(11/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--network_manager	nova.network.manager.VlanManager	Cloud controllerがOpenStackの各Nodeと仮想計算機に接続する方法. オプション (1) nova.network.manager.FlatManager 簡易なnon-VLANネットワーク (2) nova.network.manager.FlatDHCPManager DHCPを用いたFlatネットワーク (3) nova.network.manager.VlanManager DHCPを用いたVLANネットワーク. 何も設定されていないければ, この設定になる.

nova.confのフラグ(12/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--network_driver	nova.network.linux_net	ネットワークを作成するためのドライバ.
--network_host	preciousroy.hsd1.ca.comcast.net	FlatモデルにおいてIPアドレスの割り当てに用いるネットワークホスト.
--network_size	256	各プライベートsubnetで使用するアドレスの数.
--num_networks	1000	提供するネットワークの数.
--network_topic	network	ネットワークノードがLISTENするノードのテーマ.
--node_availability_zone	nova	このノードが利用可能な領域.
--null_kernel	nokernel	カーネルイメージをrawディスクイメージとして使用する.

nova.confのフラグ(13/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--osapi_host	\$my_ip	APIサーバのIPアドレス
--osapi_path	/v1.0/	
--osapi_port	8774	APIサーバポート番号.
--osapi_scheme	http	OpenStack APIのURLの接頭辞.
--periodic_interval	60	タスクを走行させる間隔(秒).
--pidfile		このサービスのためのpidファイルの名前.
--rabbit_host	localhost	Rabbitmqをインストールした場所のIPアドレス.
--rabbit_max_retries	12	Rabbitの接続数.
--rabbit_password	guest	Rabbitmqサーバのパスワード.

nova.confのフラグ(14/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--rabbit_port	5672	Rabbitmqサーバが走行/LISTENしているポート番号.
--rabbit-retry-interval	10	Rabbitが再接続する間隔.
--rabbit_userid	guest	Rabbitの接続に用いるユーザID.
--region_list		コンマで区切ったFQDNの組み.
--report_interval	10	ノードがdata storeに状態を通知する時間(秒).
--routing_source_ip	10	ネットワークホストのパブリックIP.
--s3_dmz	\$my_ip	インスタンスの内部IPアドレス.

nova.confのフラグ(15/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--s3_host	\$my_ip	S3ホストのIPアドレス. objectstoreサービスを提供しているOpenStack Computeの場所.
--s3_port	3333	S3ホストが走行しているポート番号.
--scheduler_manager	nova.scheduler.manager.SchedulerManager	Schedulerのためのマネージャ.
--scheduler_topic		SchedulerがLISTENするノード.
--sql_connection	sqlite:///state_path/nova.sqlite	OpenStack Compute SQLデータベースの場所.
--sql_idle_timeout	3600	
--sql_max_retries	12	SQLが接続を試行する数.

nova.confのフラグ(16/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--sql_retry_interval	10	SQL接続のための試行間隔.
--state_path	/usr/lib/pymodules/python2.6/nova/..	Novaの状態を維持するためのトップレベルディレクトリ.
--use_ipv6	false	IPv6ネットワークアドレスを使用するか否か. 1かtrueで有効化.
--use_s3	true	s3からImageを取得するか否か. 1かtrueで有効化.
--verbose	false	初期設定中に詳細な情報を出力するか否か 1かtrueで有効化.
--vlan_interface	eth0	VlanManagerとvlanがブリッジ接続するインタフェース.

nova.confのフラグ(17/18)

<nova-api, nova-computeに関するフラグ>

設定項目	初期値	説明
--vlan_start	100	プライベートネットワークのための最初のVLAN.
--vpn_image_id	ami-cloudpipe	cloudpipe VPNサーバのためのAMI(Amazon Machine Image)
--vpn_key_suffix	-vpn	VlanManager

nova.confのフラグ(18/18)

<nova-volumeに関するフラグ>

設定項目	初期値	説明
--iscsi_ip_prefix		簡単な文字列マッチングに使用されるIPアドレス.
--colume_manager	nova.volume.manager.Volum eManager	nova-volumeが使用するマネージャ.
--volume_name_template	volume-%08x	volumeの名前を生成するために使用するテンプレート.
--volume_topic	volume	volumeノードがLISTENするテーマの名前.