MIRACLE ZBX + Hatohol による OpenStack 監視環境構築手順書

ミラクル・リナックス株式会社

更新日: 2015-02-09

バージョン: 1.0.1

更新履歴

バージョン	日付	更新内容
1.0.0	2015-02-05	初版
1.0.1	2015-02-09	項番 5.4.3 7(6) マクロ設定を追加

目次

1 前提

日本仮想化技術株式会社が作成、公開している「OpenStack 構築手順書 Icehouse 版」にしたがって構築された OpenStack 環境が存在していることを前提とします。また、OpenStack 環境に生成されるインスタンスがインターネットと通信を行えることが必要です。

本書では、既に構築されているテナント demo 上に CentOS 6.6 のインスタンスを 2 つ起動し、一方を MIRACLE ZBX Server として、他方を MIRACLE ZBX Agent ホストとして自動構築します。また、構築した MIRACLE ZBX Server に OpenStack を構成する各ノードを監視する機能を追加します。

なお、OpenStack を構成する各ノードの監視を行う MIRACLE ZBX Server は多数のホストを監視する特性上、また通常のインスタンスに永続性がない特性上から OpenStack インスタンス外に構築することを推奨しますが、その際の参考としてください。

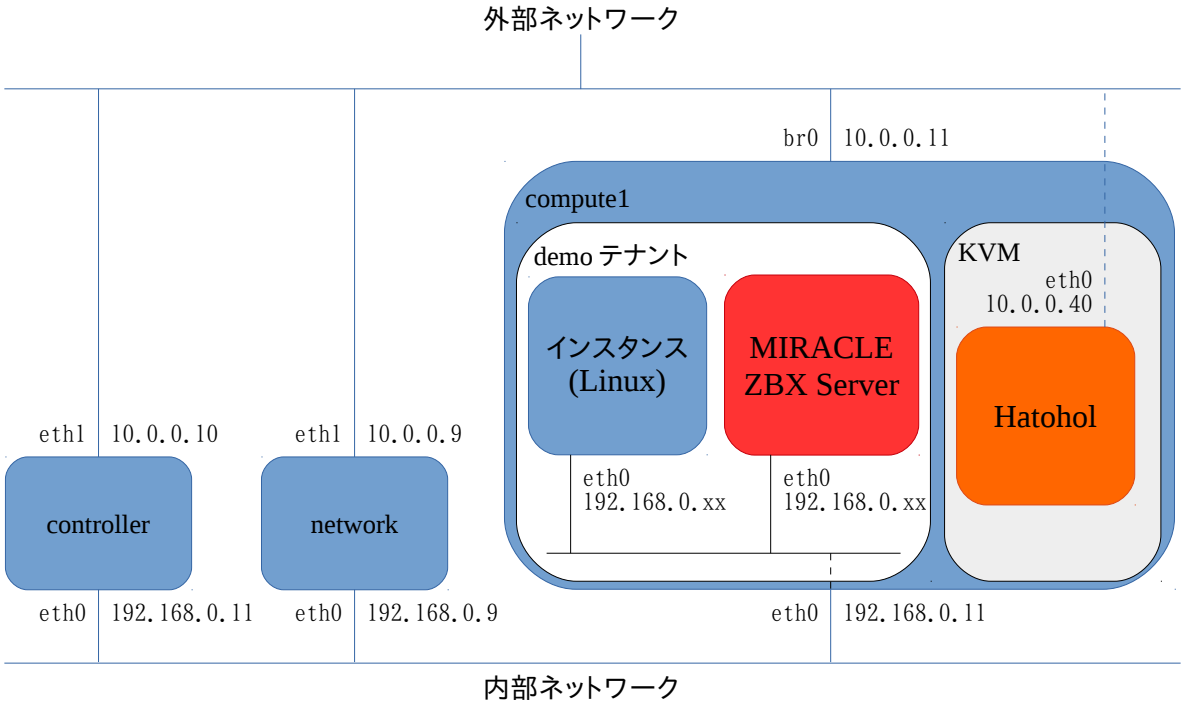
加えて、ノード compute1 上に(OpenStack インスタンスではなく)KVM ゲスト環境(CentOS 6.6)を起動し、Hatohol サーバーとして自動構築します。Hatohol による MIRACLE ZBX Server, Zabbix Server の集約監視を構築する際の参考としてください。

2 構築後の論理構成

本書にしたがって操作を行うと、下図の構成で監視環境が構築されます。

なお、demo テナントと内部ネットワークのネットワークアドレスが同一ですが、それぞれ独立したネットワークとなるため通信を行うことはできません。インスタンスとの通信は Floating IP を介して行うことになります。

Hatohol サーバーは KVM 環境となるため、ブリッジを追加すれば OpenStack を構成する各ノードとの通信は内部ネットワーク、外部ネットワークのいずれでも行うことができます。本書では外部ネットワークを使用して通信することを想定して説明します。



3 OpenStack 環境の設定変更

初期状態では外部ネットワークとの通信に支障があるため、インスタンスの MTU を縮小します。次のファイルを新規作成します。実行例では MTU を 1400 としていますが、値は環境に合わせて変更してください。

network ノード: /etc/neutron/dnsmasq.conf

```
dhcp-option-force=26,1400
```

新規作成したファイルを参照するよう、ファイルの編集を行います。セクション DEFAULT に次のパラメータを追加してください。なお、ファイル中にコメントアウトされたスケルトンが用意されています。

network ノード: /etc/neutron/dhcp_agent.ini

```
[DEFAULT]
dnsmasq_config_file = /etc/neutron/dnsmasq.conf
```

また、インスタンスが名前解決に使用する DNS サーバーのフォワード先 DNS サーバーの IP アドレスを指定します。この DNS サーバーでインターネット上の一般的な名前解決ができる必要があります。

network ノード: /etc/neutron/dhcp_agent.ini

```
[DEFAULT]
dnsmasq_dns_servers = DNS サーバーの IP アドレス
```

設定変更後、OpenStack 環境を再起動します。

OpenStack 環境の再起動を行わない場合は、次の手順が必要となります。id は動的に生成されるため、設定対象の環境に合わせて変更する必要があります。

```
ubuntu@controller:$ source admin-openrc
ubuntu@controller:$ neutron agent-list          ←DHCP agent の id を特定するために実行
+--------------------------------------+--------------------+----------+-------+----------------+
| id                                   | agent_type         | host     | alive | admin_state_up |
+--------------------------------------+--------------------+----------+-------+----------------+
| 700dbe25-d6a9-4aea-ae6e-56c049cf8e3e | Open vSwitch agent | network  | :-)   | True           |
| 8b00636c-fdb2-4bd6-b8b8-ef6523f0c192 | Metadata agent     | network  | :-)   | True           |
| a8a50a4e-744b-481b-8892-0aa5a26bd974 | Open vSwitch agent | compute1 | :-)   | True           |
| cd06f9f1-2ada-415f-a37f-520618adce07 | DHCP agent         | network  | :-)   | True           | ←該当行
| f65e9719-28a5-4d88-a31d-cc6db3904af1 | L3 agent           | network  | :-)   | True           |
+--------------------------------------+--------------------+----------+-------+----------------+
ubuntu@controller:$ neutron agent-update cd06f9f1-2ada-415f-a37f-520618adce07 --admin_state_up=False
Updated agent: cd06f9f1-2ada-415f-a37f-520618adce07

※network ノード上のプロセス dnsmasq が停止したことを確認したのち、次のコマンドを実行

ubuntu@controller:$ neutron agent-update cd06f9f1-2ada-415f-a37f-520618adce07 --admin_state_up=True
Updated agent: cd06f9f1-2ada-415f-a37f-520618adce07
```

4 イメージの登録

本項では MIRACLE ZBX Server, MIRACLE ZBX Agent ホスト, Hatohol Server の稼働環境として使用する CentOS 6.6 のクラウドイメージを入手し、OpenStack 環境に登録する手順を示します。

4.1 CentOS クラウドイメージのダウンロード

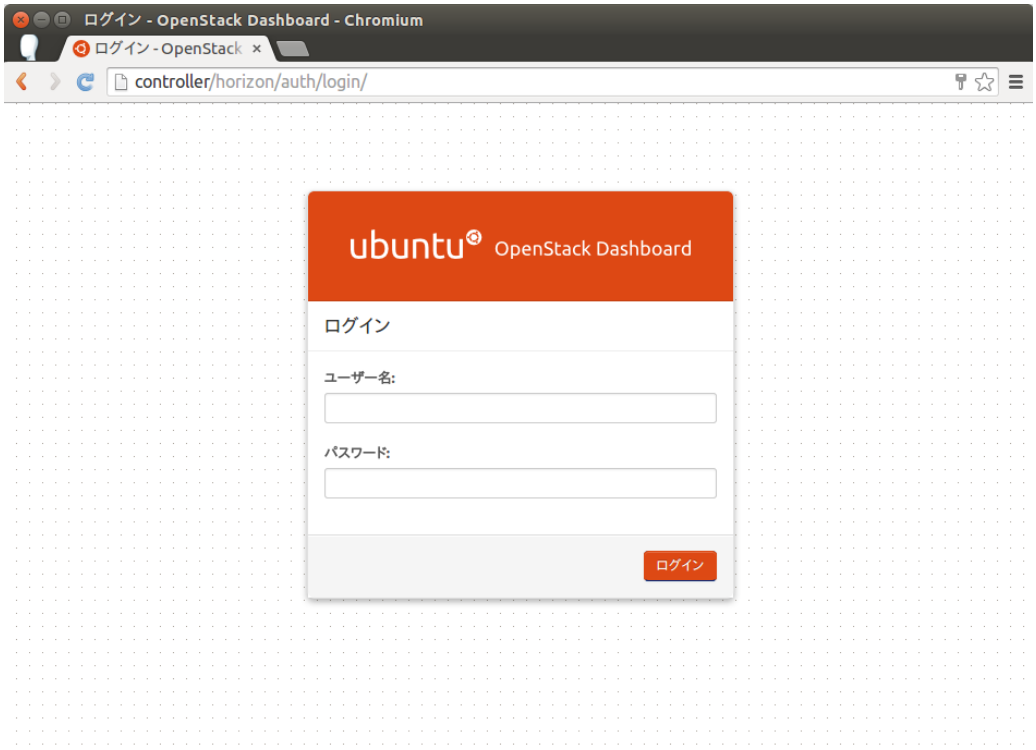
次の URL から、CentOS 6.x のクラウドイメージを事前にダウンロードしてください。執筆時点では 6.6 が最新のため、これを使用します。CentOS 公式 Web サイトのページ「GET CENTOS」の右下に表示されているリンク「Check out our Clouds」をクリックし、ファイル「CentOS-6-x86_64-GenericCloud.qcow2」をダウンロードしてください。

4.2 ダウンロードしたクラウドイメージの登録

4.2.1 OpenStack フロントエンドを使用する場合

1. **ログイン**

   Web ブラウザで OpenStack フロントエンドへアクセスし、ユーザ demo でログインしてください。
   「OpenStack 構築手順書」にしたがって構築した環境では、パスワードが password となっています。



2. **イメージ画面の表示**

   パネル「コンピュート」を開き、カテゴリー「イメージ」をクリックします。下図のように、登録済みイメージの一覧が表示されます。ボタン「+イメージの作成」をクリックします。

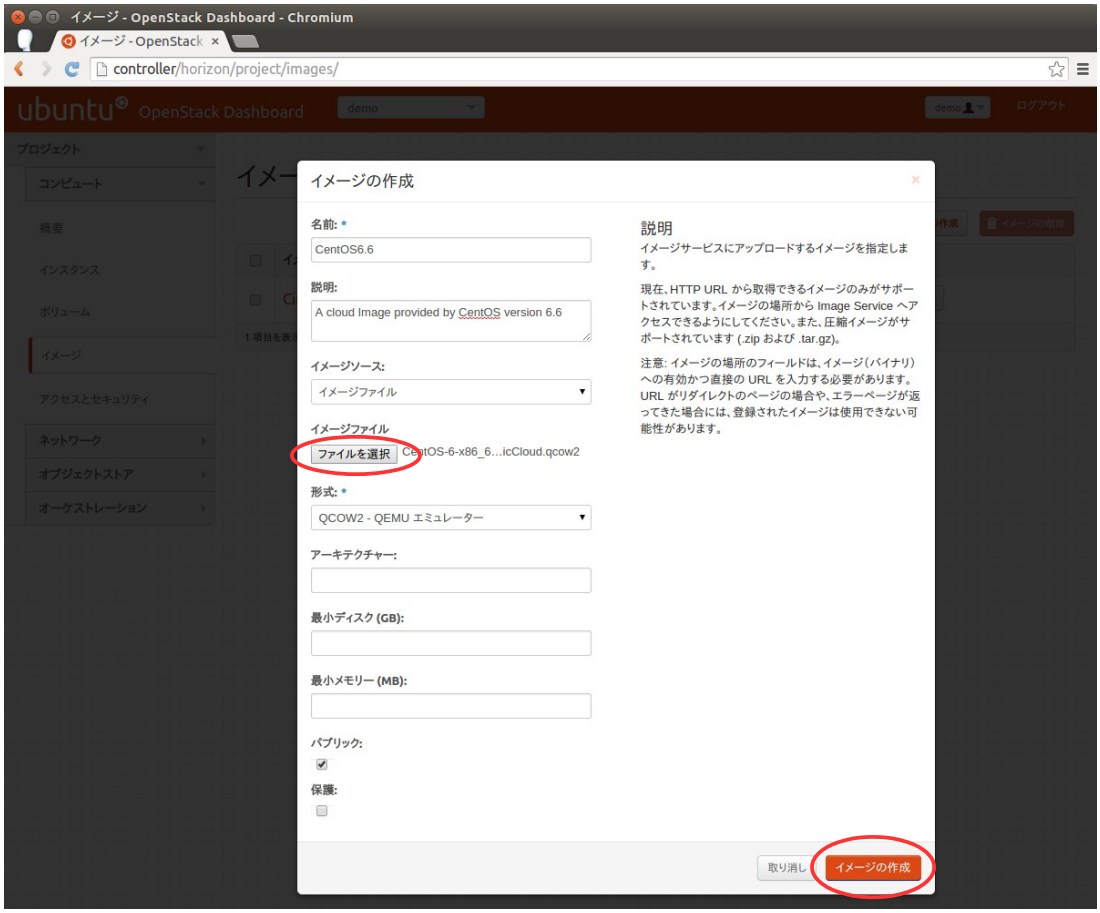
3. イメージ登録情報の入力

登録に際して必要となる情報を入力し、イメージファイルを指定します。次の表にしたがって入力してください。

イメージファイルの項目ではボタン「ファイルを選択」をクリックし、ポップアップウィンドウでダウンロード済み CentOS 6.6 クラウドイメージを選択します。

値が(任意)となっている項目は、設定したい状態に合わせて変更してください。

項目	値
名前	CentOS6.6
説明	(任意) 実行例: A cloud image provided by CentOS version 6.6
イメージソース	イメージファイル
イメージファイル	(ダウンロードした CentOS 6.6 クラウドイメージ)
形式	QCOW2 QEMU エミュレーター
アーキテクチャ	(空欄)
最小ディスク (GB)	(空欄)
最小メモリー (MB)	(空欄)
パブリック	(任意) 全てのユーザに公開する場合にチェック
保護	(任意) 登録したユーザのみ削除可能とする場合にチェック



入力終了後、画面右下のボタン「イメージの作成」をクリックします。

4. **イメージ登録状態の確認**

正常に登録されると、登録済みイメージの一覧に 1 行増えます。状態が「Active」と表示されていることを確認してください。

4.2.2 コマンドラインを使用する場合

クラウドイメージファイルを controller ノード上に転送すると、コマンドラインで OpenStack 環境で使用可能なイメージとして登録することが可能となります。

1. **テナント demo 用環境変数の設定**

「OpenStack 構築手順書 Icehouse 版」で作成した、テナント demo 操作に必要となる環境変数を設定するためのファイル demo-openrc を読み込みます。

```
controller:$ source demo-openrc
```

2. **イメージファイルの Glance への登録**

次のコマンドを実行し、クラウドイメージファイル CentOS-6-x86_64-GenericCloud.qcow2 を、CentOS6.6 という名称で登録します。

なお、次の実行例ではクラウドイメージファイルはカレントディレクトリに配置していることを前提とします。また、全てのユーザに公開するようにオプションを指定しています。

```
controller:$ glance image-create --name="CentOS6.6" --disk-format=qcow2 ¥
 --container-format=bare --is-public=true < CentOS-6-x86_64-GenericCloud.qcow2
```

5 インスタンスの起動

5.1 キーペアの生成

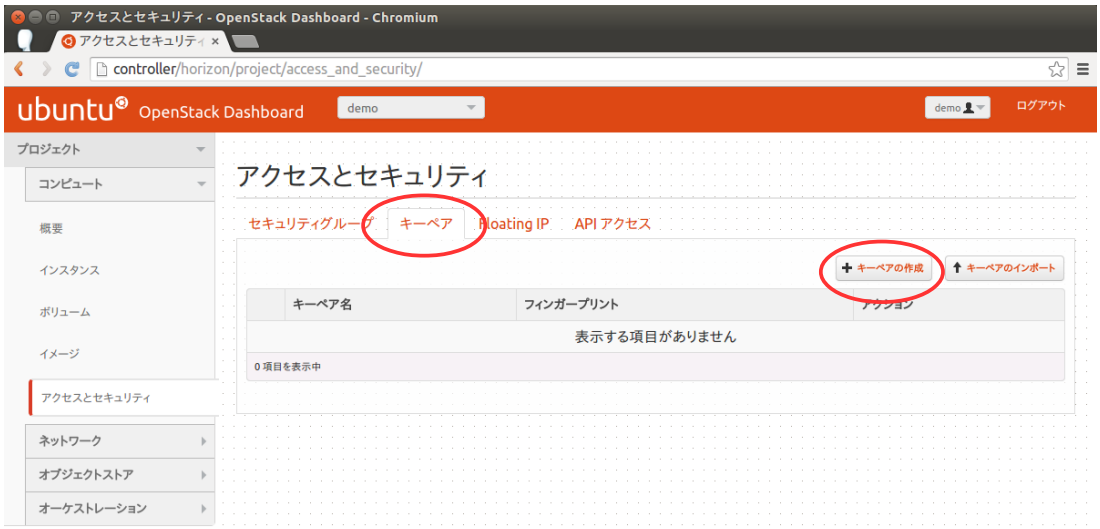
生成後のインスタンスに対し SSH で接続するには、公開鍵認証が必要となります。その際に使用する秘密鍵を、事前に接続元にダウンロードします。操作はいずれもユーザ demo でログインした状態で実施します。

1. **キーペア画面の表示**

   パネル「コンピュート」を開き、カテゴリー「アクセスとセキュリティ」をクリックします。

   続いて、タブ「キーペア」をクリックします。下図のように、キーペアの一覧が表示されます(この時点では存在しないため、「表示する項目がありません」と出力されています)。

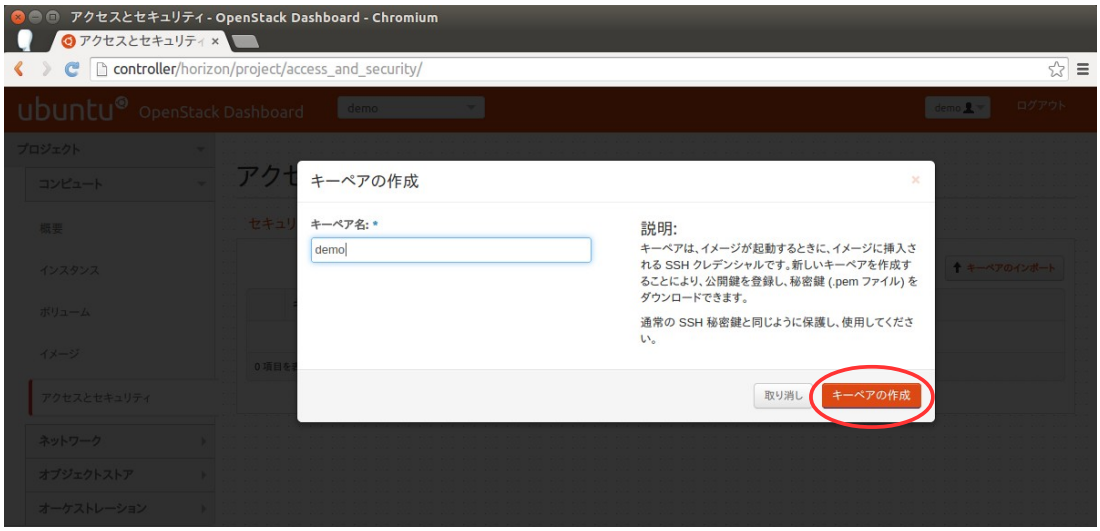
   ボタン「+イメージの作成」をクリックします。



2. **キーペア名の指定**

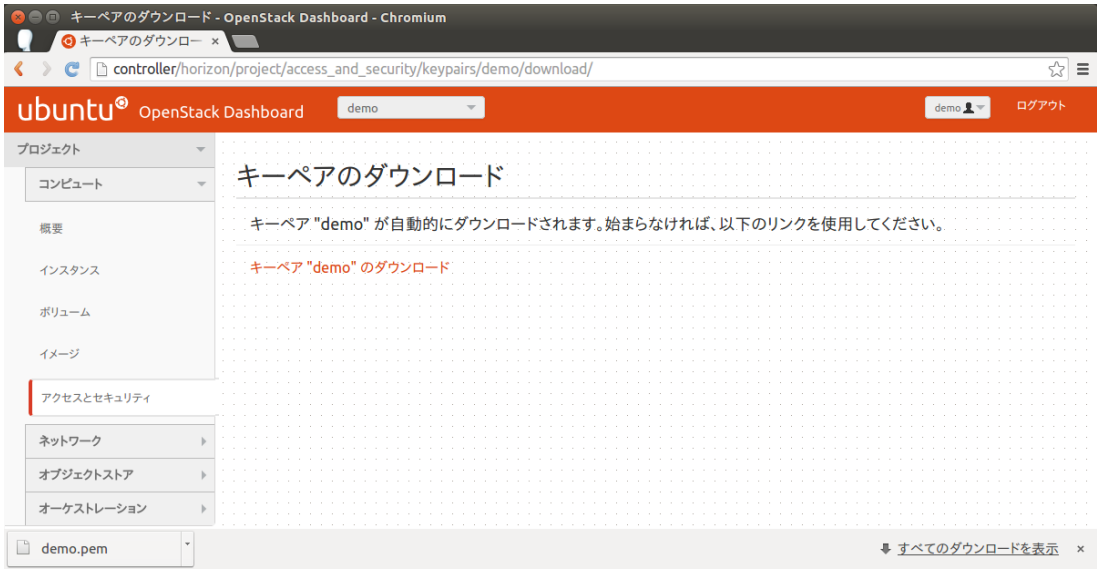
   キーペア名を入力します。以降の実行例では「demo」と指定したと仮定して説明します。

   入力後、ボタン「キーペアの作成」をクリックします。



3. **秘密鍵のダウンロード**

即座に秘密鍵のダウンロードが開始されます。SSH で接続する際に使用できるよう、適切なディレクトリに配置してください。



5.2 インスタンスの生成

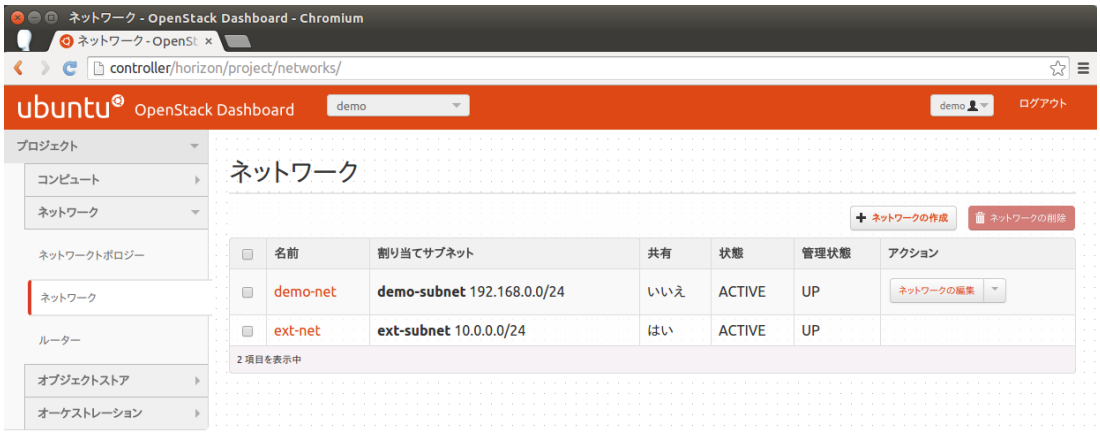
ここでは、MIRACLE ZBX Server と MIRACLE ZBX Agent ホスト(監視対象ホスト)等を定義したテンプレートファイル stack_zabbix.yaml を利用し、スタックとして一度に生成する方法を説明します。stack_zabbix.yaml は本書とともに入手可能です。

なお、MIRACLE ZBX Server の各種パラメータは試験的に動作させるために必要な値が設定されます。実運用システムに使用する場合は stack_zabbix.yaml を編集するか、生成後のシステムのパラメータを変更する必要があります。

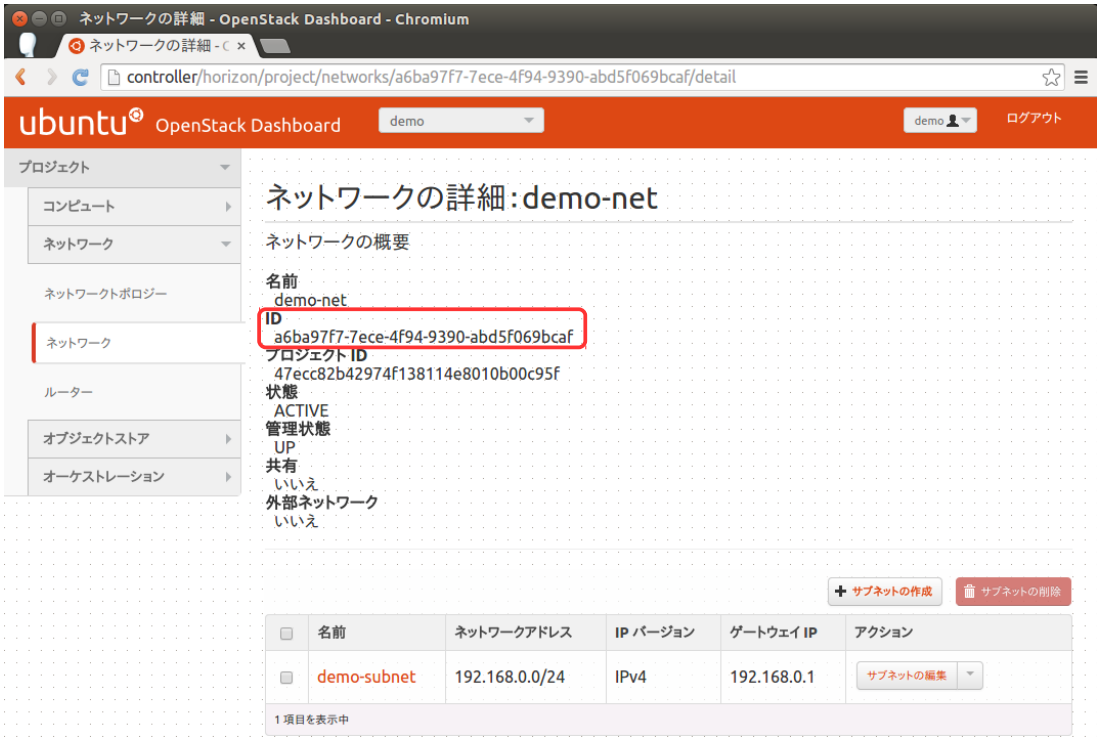
5.2.1 OpenStack フロントエンドを使用する場合

1. **ネットワーク、サブネットの ID の取得**

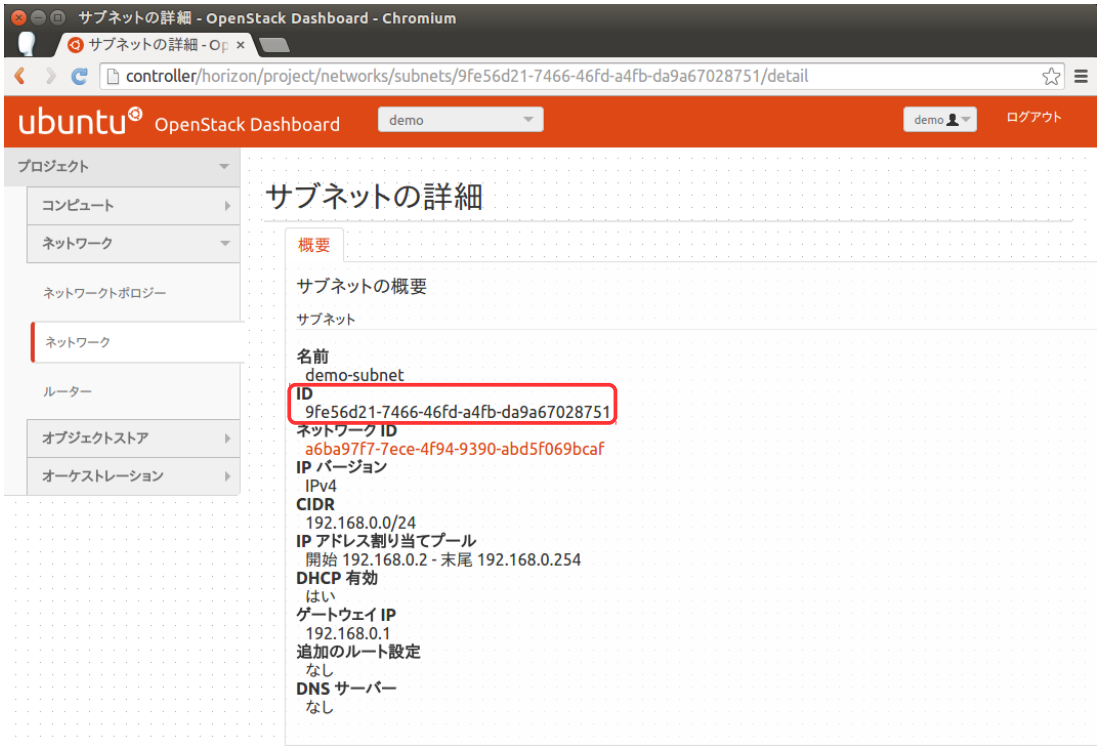
   パネル「ネットワーク」を開き、カテゴリー「ネットワーク」をクリックします。この先のリンクから、ネットワークおよびサブネットの ID を取得します。



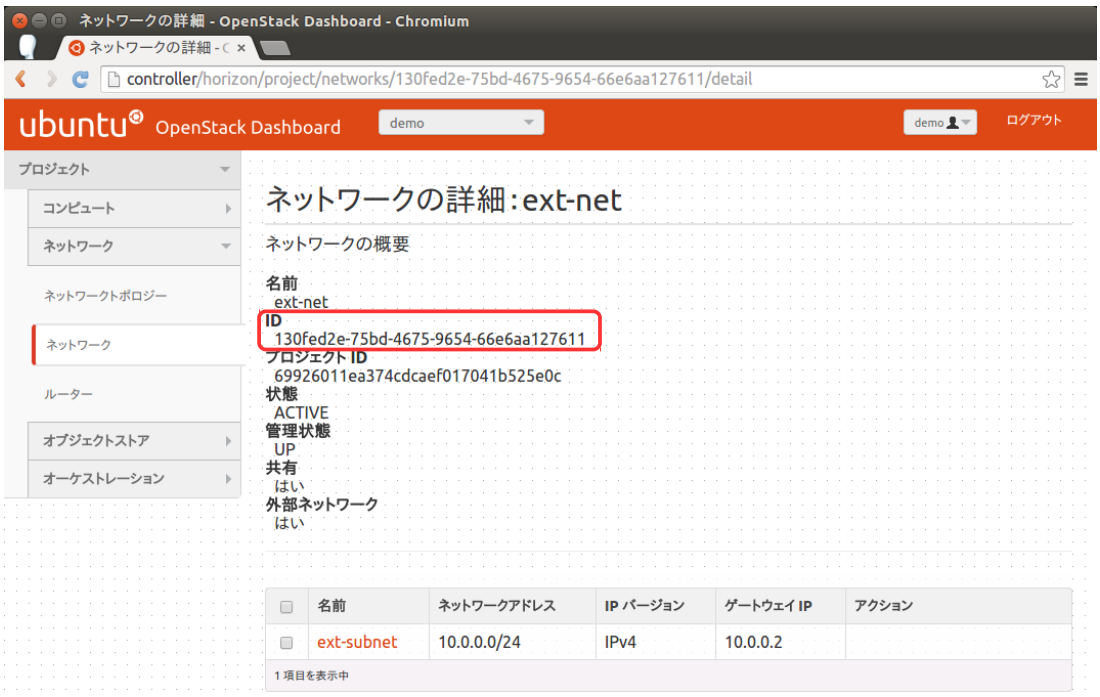
   リンク「demo-net」をクリックすると、次の画面が表示されます。赤枠箇所がネットワーク「demo-net」の ID です。これを控えておきます。



続けて、画面下方のリンク「demo-subnet」をクリックすると、次の画面が表示されます。赤枠箇所がサブネット「demo-subnet」の ID です。これを控えておきます。



カテゴリー「ネットワーク」に戻り、リンク「ext-net」をクリックします。次の画面が表示されます。赤枠箇所がネットワーク「ext-net」の ID です。これを控えておきます。

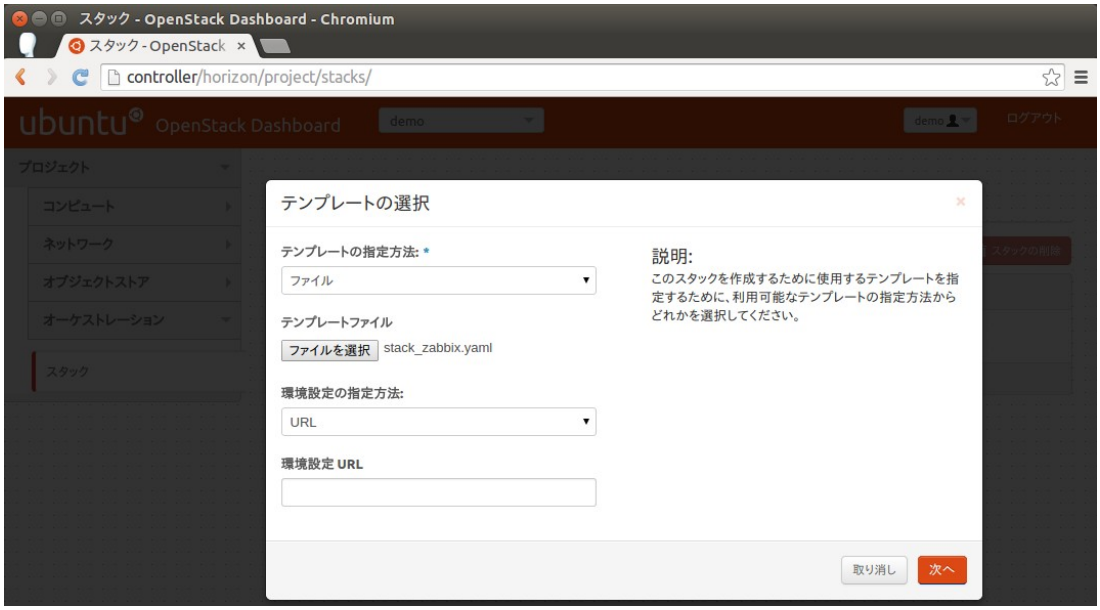


2. スタックの作成

パネル「オーケストレーション」を開き、カテゴリー「スタック」をクリックします。画面右側のボタン「+スタックの起動」をクリックします。

ダイヤログボックス「テンプレートの選択」が表示されます。ドロップダウン「テンプレートの指定方法」でファイルを選択し、テンプレートファイルとして「stack_zabbix.yaml」を選択します。

これらを選択したのち、ボタン「次へ」をクリックします。

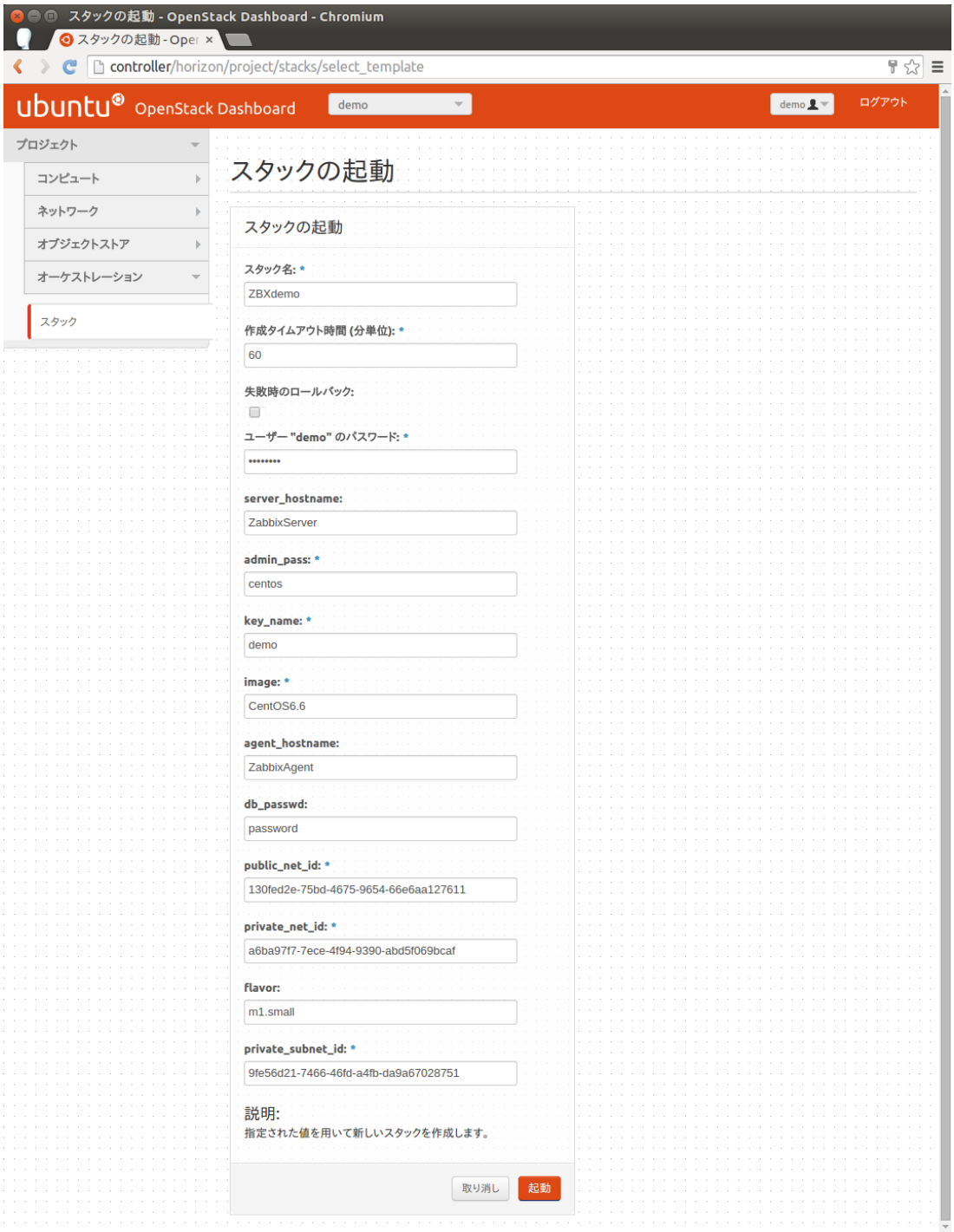


次にダイヤログボックス「スタックの起動」が表示されます。以下の情報を入力します。

項目	値
スタック名	任意（実行例では ZBXdemo）
ユーザー demo のパスワード	password
server_hostname	任意（初期値: ZabbixServer）
admin_pass	任意（ログインユーザ centos のパスワードとして設定される）
key_name	demo（キーペア名を入力）
image	CentOS6.6
agent_hostname	任意（初期値: ZabbixAgent）
db_pass	任意（初期値: password。Zabbix 用 DB「zabbix」のパスワード）
public_net_id	ネットワーク「ext-net」の ID
private_net_id	ネットワーク「demo-net」の ID
flavor	任意（初期値: m1.small。m1.small 以上のリソースが必須）

private_subnet_id	サブネット「demo-subnet」の ID

実行例の環境では、次の図のように入力することとなります。入力終了後、ボタン「起動」をクリックします。



画面が自動的に切り替わり、スタックの一覧が表示されます。列「状態」が「In Progress」から「Complete」に切り替わるまで、しばらく待ちます。

【状態: In Progress】

【状態: Complete】

5.2.2 コマンドラインを使用する場合

あらかじめファイル stack_zabbix.yaml を controller ノードに転送してください。以下の実行例ではカレントディレクトリに配置されていることを前提としています。

1. **テナント demo 操作用環境変数の設定**

   ファイル demo-openrc に記述された環境変数を読み込みます。

```
controller:$ source demo-openrc
```

2. **ネットワーク、サブネットの ID の取得**

   コマンド neutron を使用し、ext-net と demo-net、demo-subnet の ID を取得します。以下の実行例で赤文字で記載されている箇所が該当します。

```
controller:$ neutron net-list
+--------------------------------------+----------+------------------------------------------------------+
| id                                   | name     | subnets                                              |
+--------------------------------------+----------+------------------------------------------------------+
| 130fed2e-75bd-4675-9654-66e6aa127611 | ext-net  | 9a32a953-92fe-4c64-91c5-f5f476cc31e9 10.0.0.0/24     |
| a6ba97f7-7ece-4f94-9390-abd5f069bcaf | demo-net | 9fe56d21-7466-46fd-a4fb-da9a67028751 192.168.0.0/24  |
+--------------------------------------+----------+------------------------------------------------------+
```

```
controller:$ neutron subnet-list
+--------------------------------------+-------------+---------------+-----------------------------------------------------+
| id                                   | name        | cidr          | allocation_pools                                    |
+--------------------------------------+-------------+---------------+-----------------------------------------------------+
| 9a32a953-92fe-4c64-91c5-f5f476cc31e9 | ext-subnet  | 10.0.0.0/24   | {"start": "10.0.0.200", "end": "10.0.0.250"}        |
| 9fe56d21-7466-46fd-a4fb-da9a67028751 | demo-subnet | 192.168.0.0/24| {"start": "192.168.0.2", "end": "192.168.0.254"}    |
+--------------------------------------+-------------+---------------+-----------------------------------------------------+
```

3. スタックの作成

コマンド heat を使用してスタックを作成します。与えるパラメータは次の形式となります。なお、""の内側にスペースを入れないように注意してください。

```
controller:$ heat stack-create スタック名 --template-file ファイル名 --parameters ¥
"key_name=キーペア名;¥
image=使用するイメージ名;¥
admin_pass=インスタンスの OS ユーザのパスワード;¥
public_net_id=ext-net の ID;¥
private_net_id=demo-net の ID;¥
private_subnet_id=demo-subnet の ID"
```

実行例の環境では、次のコマンド列となります。

```
controller:$ heat stack-create ZBXdemo --template-file stack_zabbix.yaml --parameters ¥
> "key_name=demo;¥
> image=CentOS6.6;¥
> admin_pass=centos;¥
> public_net_id=130fed2e-75bd-4675-9654-66e6aa127611;¥
> private_net_id=a6ba97f7-7ece-4f94-9390-abd5f069bcaf;¥
> private_subnet_id=9fe56d21-7466-46fd-a4fb-da9a67028751"
+--------------------------------------+------------+--------------------+----------------------+
| id                                   | stack_name | stack_status       | creation_time        |
+--------------------------------------+------------+--------------------+----------------------+
| d2b015e5-77bf-4ee4-a13a-1d18bf009cf4 | testStack  | CREATE_COMPLETE    | 2015-01-24T17:27:03Z |
| 5648012c-1cfb-4e30-8fb8-c68a855491c0 | ZBXdemo    | CREATE_IN_PROGRESS | 2015-01-28T02:14:24Z |
+--------------------------------------+------------+--------------------+----------------------+
```

4. スタックの作成状態の確認

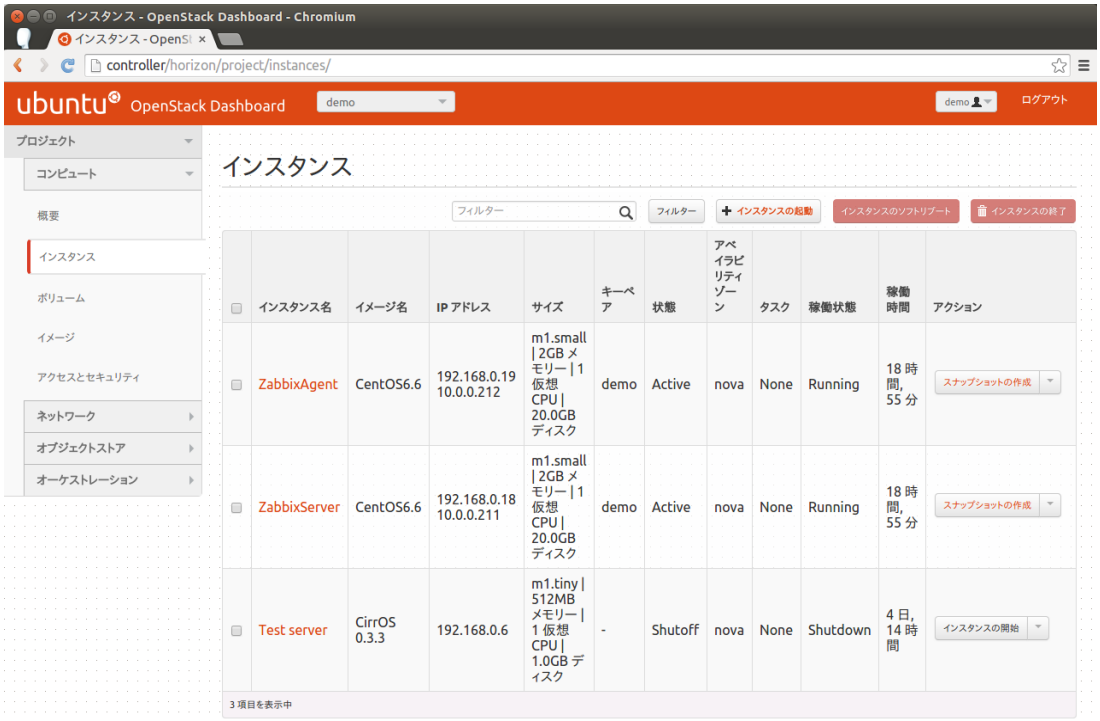
コマンド heat を使用して、スタックの作成状態を確認します。stack_name 列が ZBXdemo と出力されている行があることを確認してください。

```
controller:~$ heat stack-list
+--------------------------------------+------------+-----------------+----------------------+
| id                                   | stack_name | stack_status    | creation_time        |
+--------------------------------------+------------+-----------------+----------------------+
| d2b015e5-77bf-4ee4-a13a-1d18bf009cf4 | testStack  | CREATE_COMPLETE | 2015-01-24T17:27:03Z |
| 5648012c-1cfb-4e30-8fb8-c68a855491c0 | ZBXdemo    | CREATE_COMPLETE | 2015-01-28T02:14:24Z |
+--------------------------------------+------------+-----------------+----------------------+
```

5.3 SSH によるインスタンスへの接続確認

1. IP アドレスの確認

パネル「コンピュート」を開き、カテゴリー「インスタンス」をクリックします。存在するインスタンスのリストが表示されるので、外部からの接続に使用する Floating IP を控えておきます。



コマンドラインでは、次の手順で表示させることが可能です。

```
controller:$ source demo-openrc
controller:$ nova list
+----------------+--------------+---------+------------+-------------+----------------------------------+
| ID             | Name         | Status  | Task State | Power State | Networks                         |
+----------------+--------------+---------+------------+-------------+----------------------------------+
| ef3e3902-(省略) | Test server  | SHUTOFF | -          | Shutdown    | demo-net=192.168.0.6             |
| c3f5dc17-(省略) | ZabbixAgent  | ACTIVE  | -          | Running     | demo-net=192.168.0.19, 10.0.0.212 |
| da8cb852-(省略) | ZabbixServer | ACTIVE  | -          | Running     | demo-net=192.168.0.18, 10.0.0.211 |
+----------------+--------------+---------+------------+-------------+----------------------------------+
```

2. ssh でのログイン

他の PC から ssh で接続する場合は、項番 5.1 で生成したキーペアの秘密鍵を使用します。次の実行例では demo.pem が相当するファイルです。

```
user@localhost:~$ ssh -i Downloads/demo.pem -l centos 10.0.0.211
[centos@zabbixserver ~]$
```

ユーザ root で操作を行う必要がある場合は sudo と組み合わせて実行します。

```
[centos@zabbixserver ~]$ sudo コマンド
```

または

```
[centos@zabbixserver ~]$ sudo -s
[root@zabbixserver centos]#
```

5.4 MIRACLE ZBX の設定

5.4.1 フロントエンドの設定

1. **フロントエンドへのアクセス**

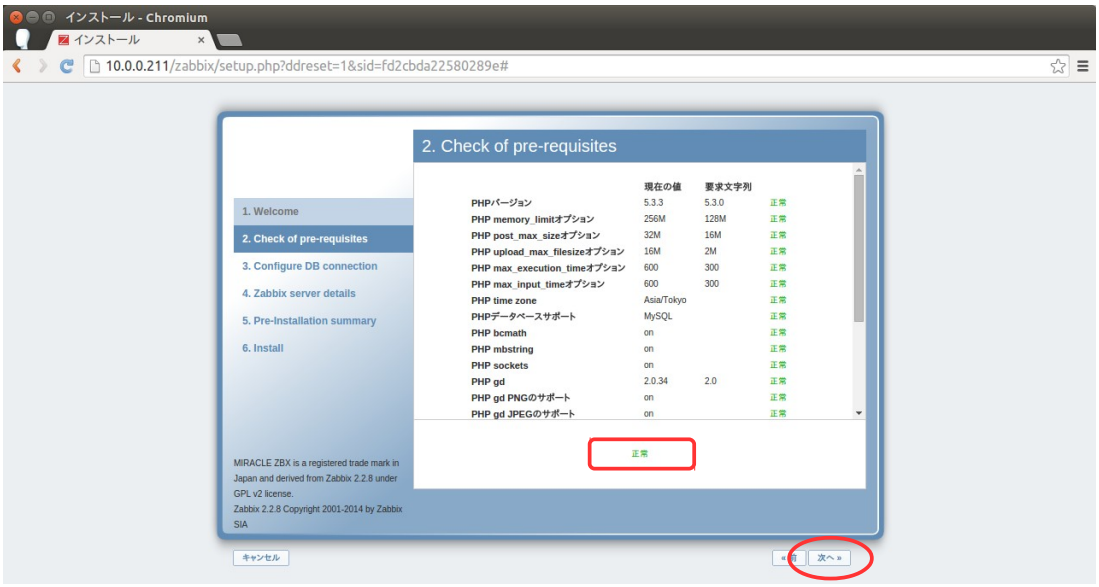
項番 5.3 で得た、インスタンス ZabbixServer の Floating IP を使用します。「外部ネットワーク」にアクセスできる環境で、ブラウザを使用して次の URL を表示してください。

```
http://<ZabbixServer の Floating IP>/zabbix/
```

下図が表示されたら、ボタン「次へ》」をクリックします。

2. **各パラメータの確認**

赤枠箇所が「正常」と表示されていることを確認し、ボタン「次へ》」をクリックします。

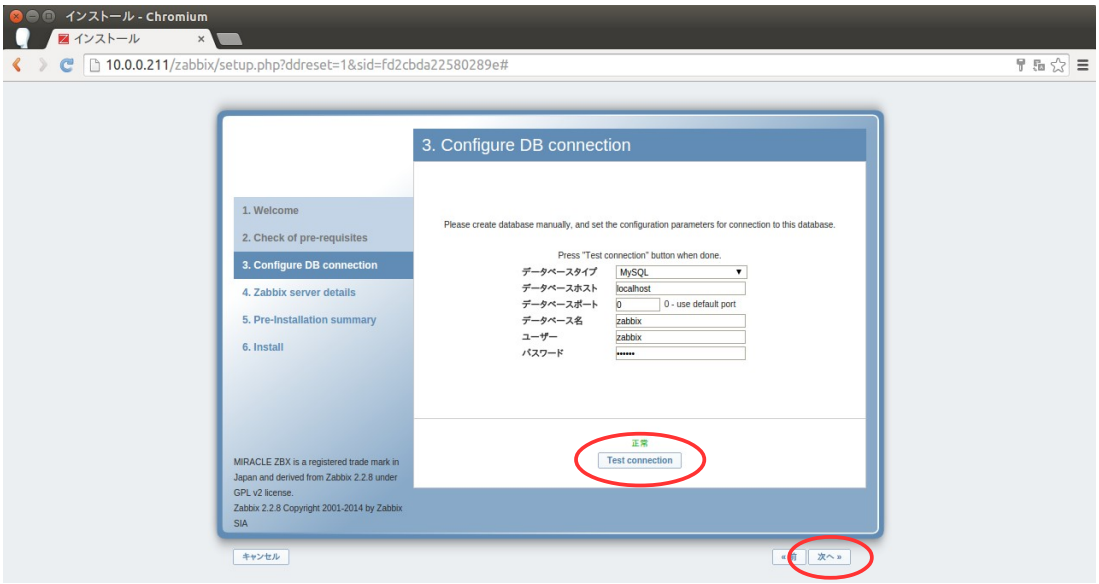


3. **DB の接続設定**

次の情報を入力します。

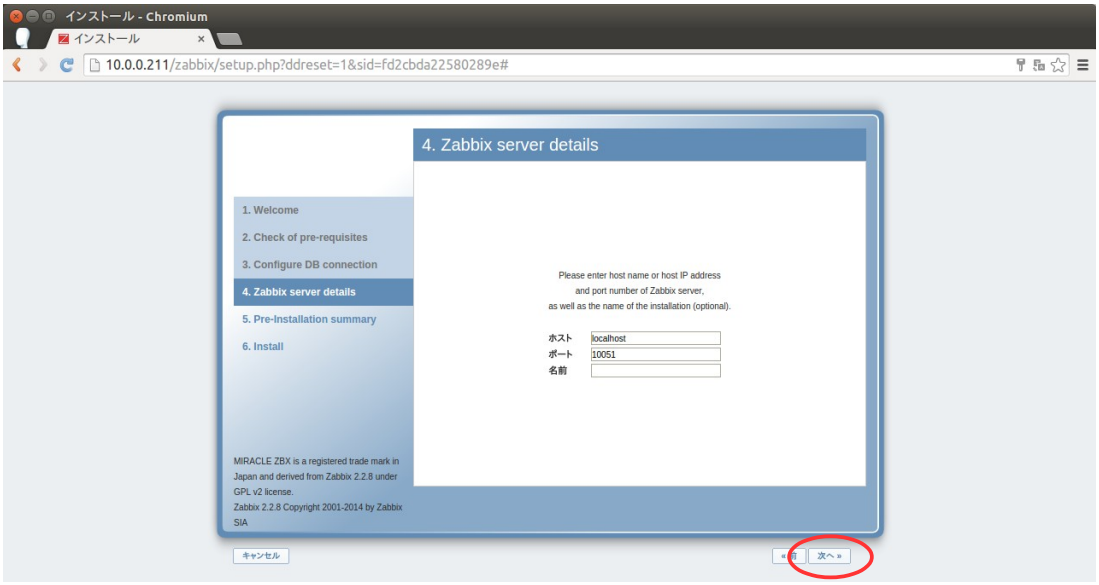
項目	値
データベースタイプ	MySQL（初期値）
データベースホスト	localhost（初期値）
データベースポート	0（初期値）
データベース名	zabbix（初期値）
ユーザー	zabbix（初期値：root）
パスワード	項番 5.2.1-2 で指定した文字列（実行例：password）

入力後、画面中段のボタン「Test connection」をクリックし、ボタンの直上に「正常」と表示されることを確認してボタン「次へ》」をクリックします。



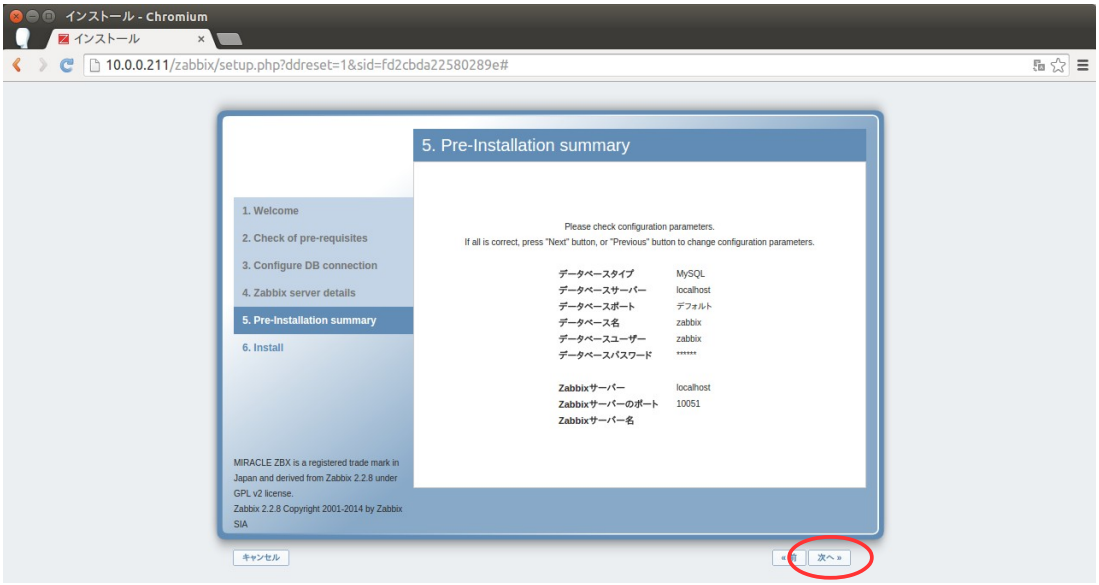
4. サーバーの詳細情報入力

特に入力は必要ありません。ボタン「次へ》」をクリックします。



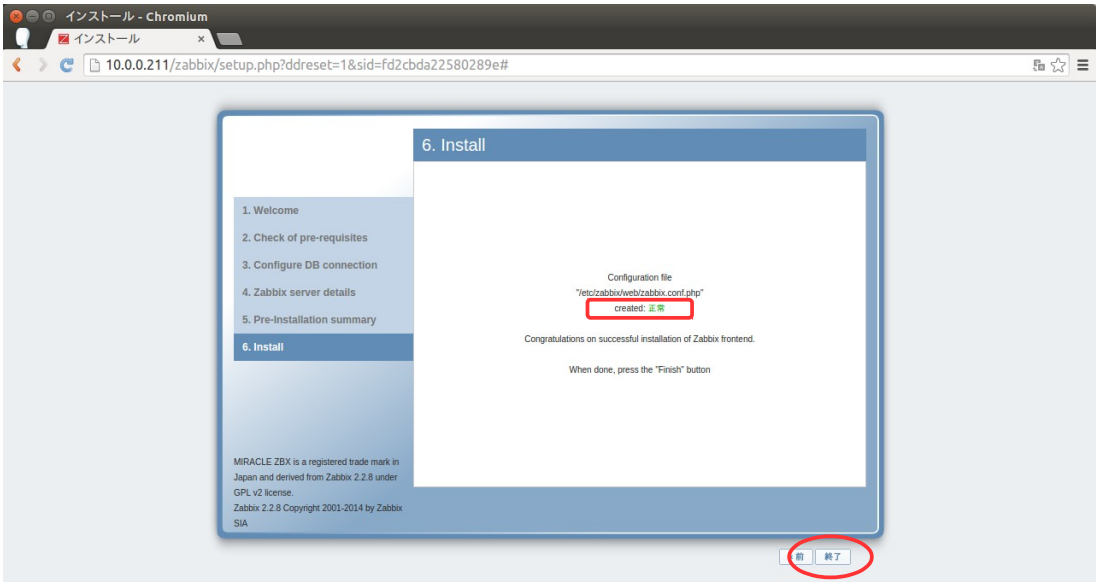
5. **設定情報の確認**

これまでの入力情報が一覧表示されます。正しいことを確認し、ボタン「次へ》」をクリックします。



6. **設定の完了**

赤枠箇所が「正常」と表示されることを確認し、ボタン「終了」をクリックします。



5.4.2 MIRACLE ZBX Agent ホストの自動登録設定

本項では、ホスト「ZabbixServer」にアクセスする MIRALCE ZBX エージェント稼働ホストが自動的に監視対象となるよう設定する方法を説明します。

なお、実行例では「ZabbixServer にアクセスする MIRACLE ZBX エージェント稼働ホストを全て同一の設定で登録する」ことを前提としています。ホストの条件によって異なる設定で自動登録するよう設定したい場合には、「アクションの実行条件」(後述)の制限を加えるなどの変更が必要です。

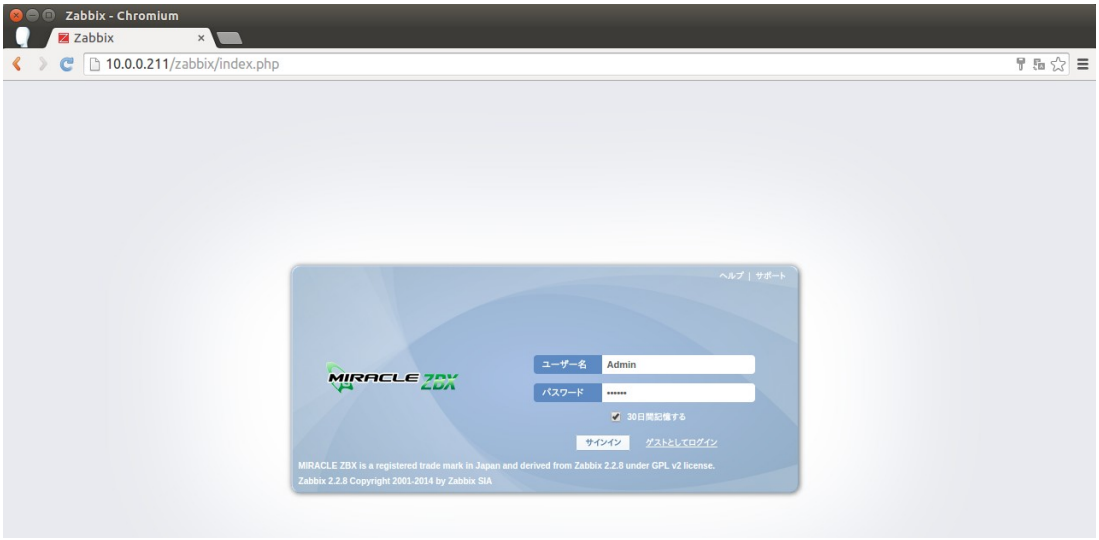
1. **フロントエンドへのログイン**

   項番 5.3 で得た、インスタンス ZabbixServer の Floating IP を使用します。「外部ネットワーク」にアクセスできる環境で、ブラウザを使用して次の URL を表示してください。

   ```
   http://<ZabbixServer の Floating IP>/zabbix/
   ```

   次の値でログインすることができます。

項目	値
ユーザー名	Admin
パスワード	zabbix



2. **MIRACLE ZBX サーバーの監視有効化**

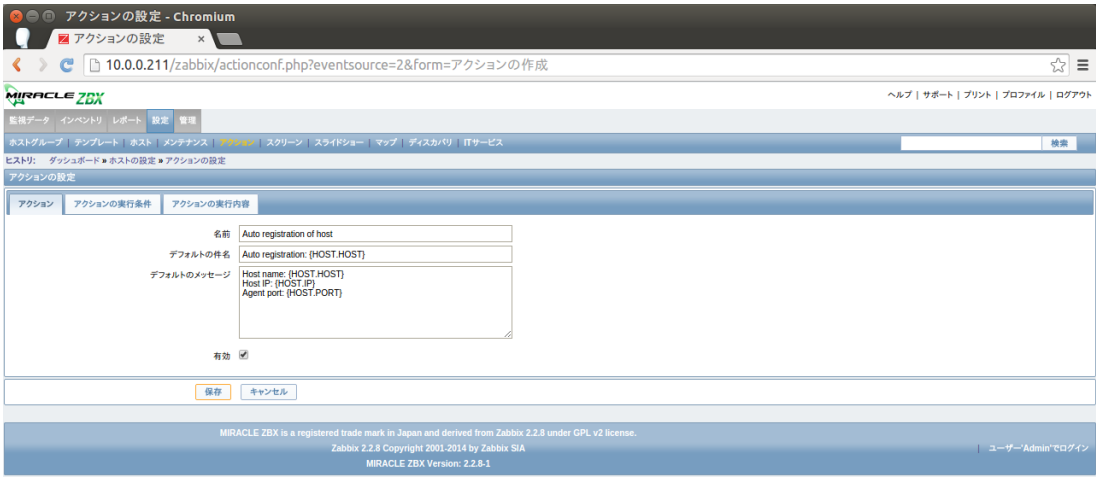
   メニュー「設定→ホスト」を選択します。下図が表示されるので、ステータス列のリンク「無効」をクリックし、ホストを「有効」にしてください。

3. アクションの追加

(1) メニュー「設定→アクション」を選択します。続いてドロップダウン「イベントソース」にて「自動登録」を選択します。

(2) ボタン「アクションの作成」をクリックします。

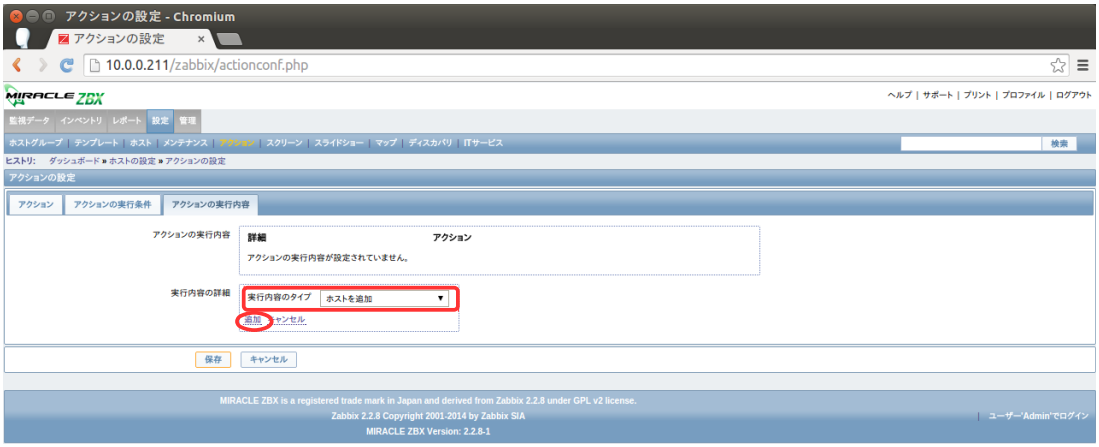
(3) タブ「アクション」の項目「名前」に、アクションの名前を入力します。任意の文字列で構いません。その他の項目は初期状態のまま変更する必要はありません。



(4) タブ「アクションの実行条件」に切り替えます。このタブではホスト名の命名規則等でアクションの実行有無を設定することができます。今回は特に設定を加えません。

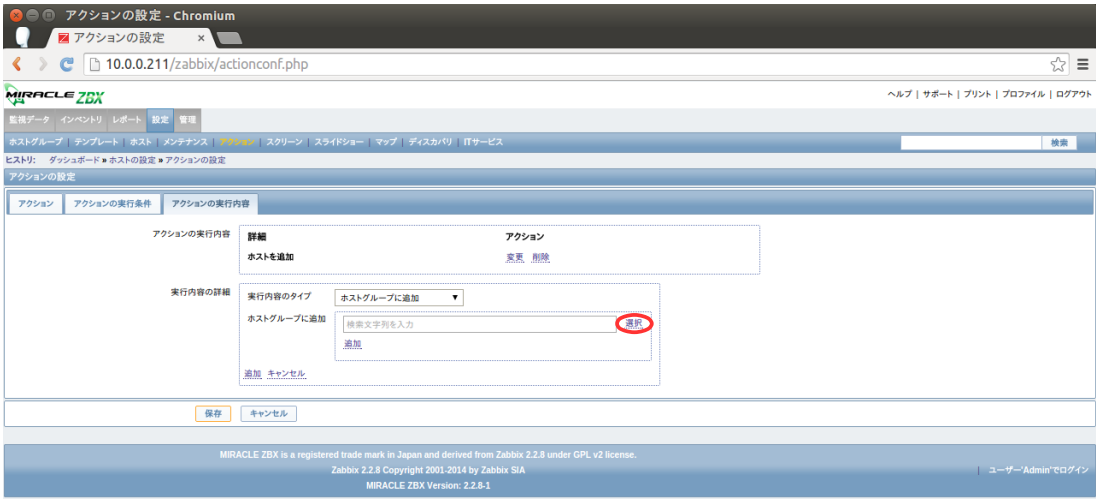
(5) タブ「アクションの実行内容」に切り替えます。次にフレーム「アクションの実行内容」内のリンク「新規」をクリックします。

(6) 赤枠箇所のドロップダウン「実行内容のタイプ」にて「ホストを追加」を選択します。続いて赤楕円箇所のリンク「追加」をクリックします。



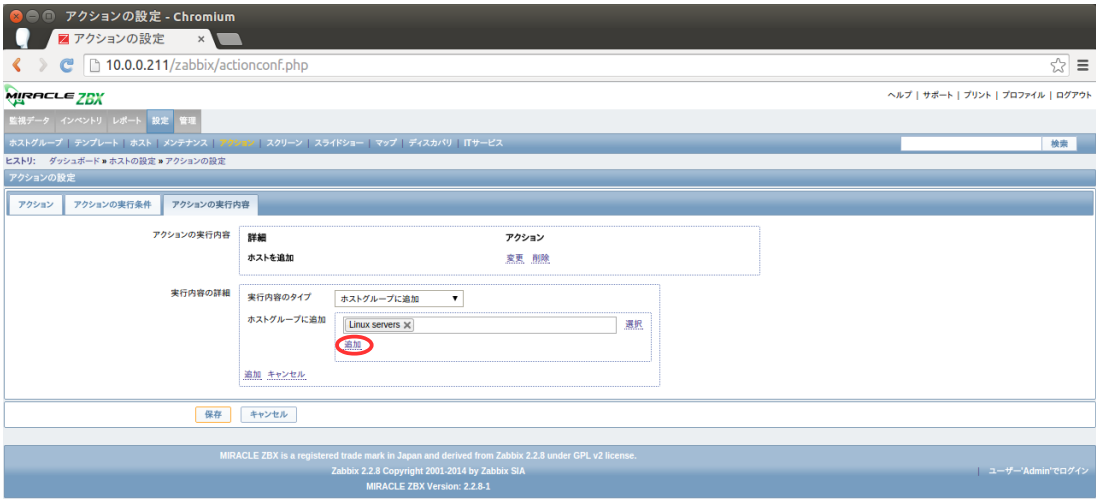
(7) フレーム「アクションの実行内容」のリンク「新規」をクリックします。

(8) ドロップダウン「実行内容のタイプ」にて「ホストグループに追加」を選択します。続いて赤楕円箇所のリンク「選択」をクリックします。

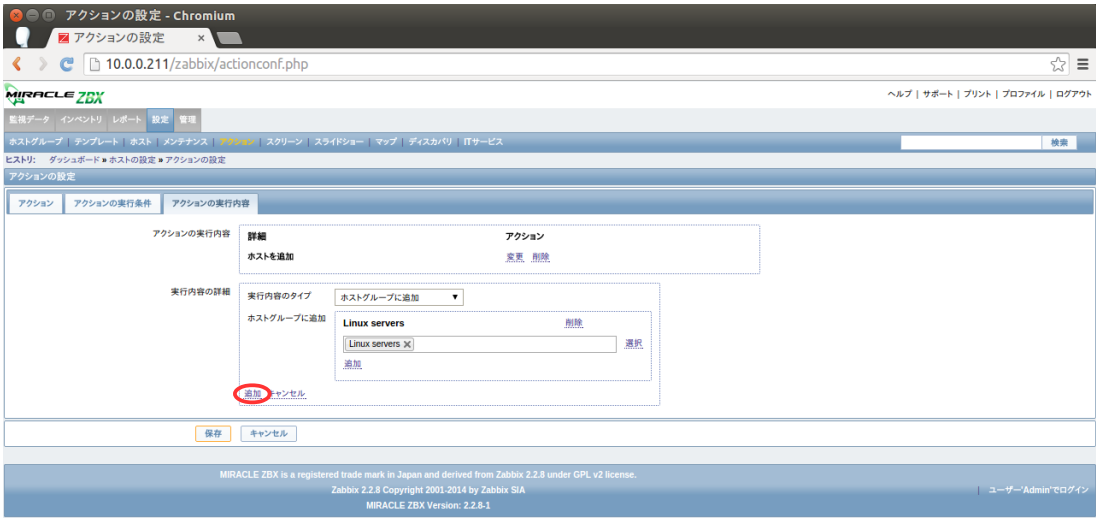


(9) ポップアップウィンドウ「ホストグループ」が表示されます。「Linux servers」列にチェックを入れ、ボタン「選択」をクリックします。

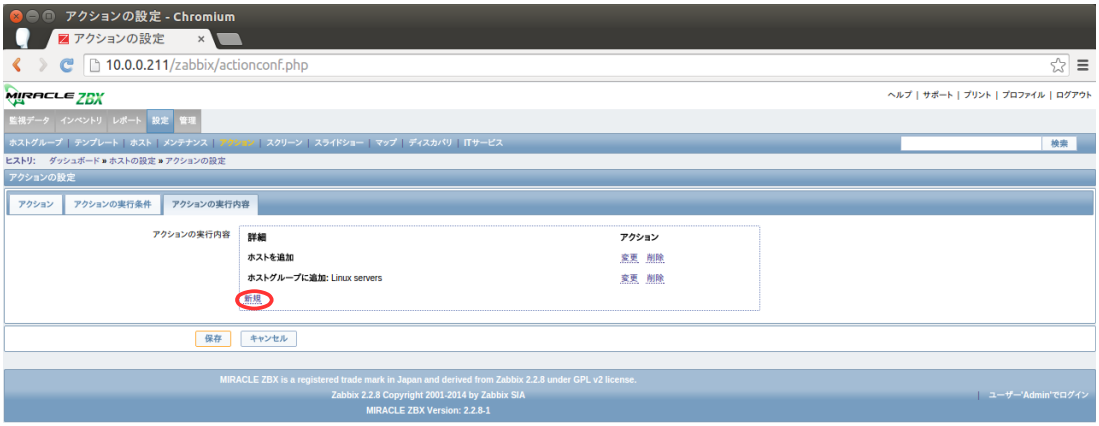
(10) フレーム「ホストグループに追加」内のリンク「追加」をクリックします。



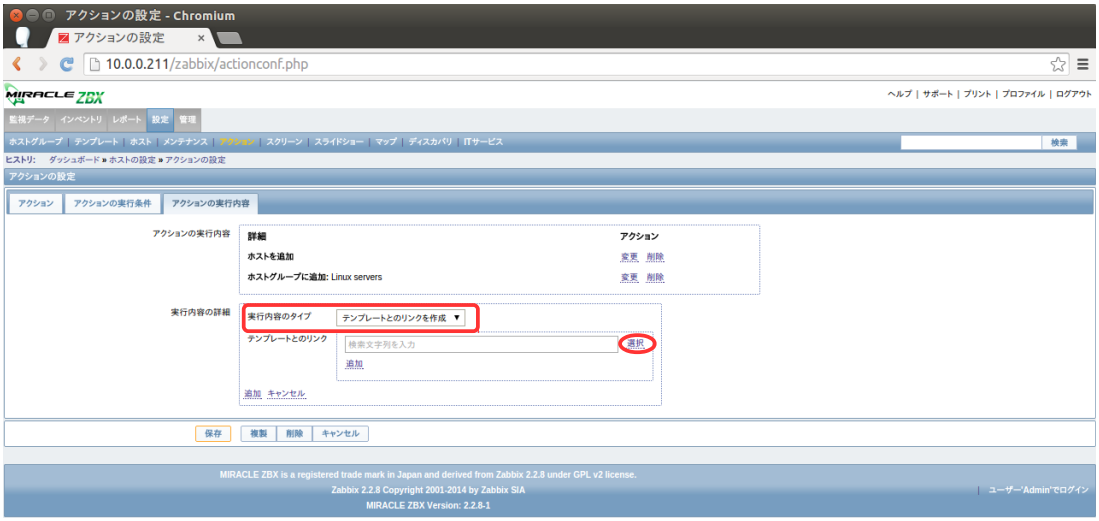
(11) フレーム「実行内容の詳細」内のリンク「追加」をクリックします。



(12) フレーム「アクションの実行内容」のリンク「新規」をクリックします。

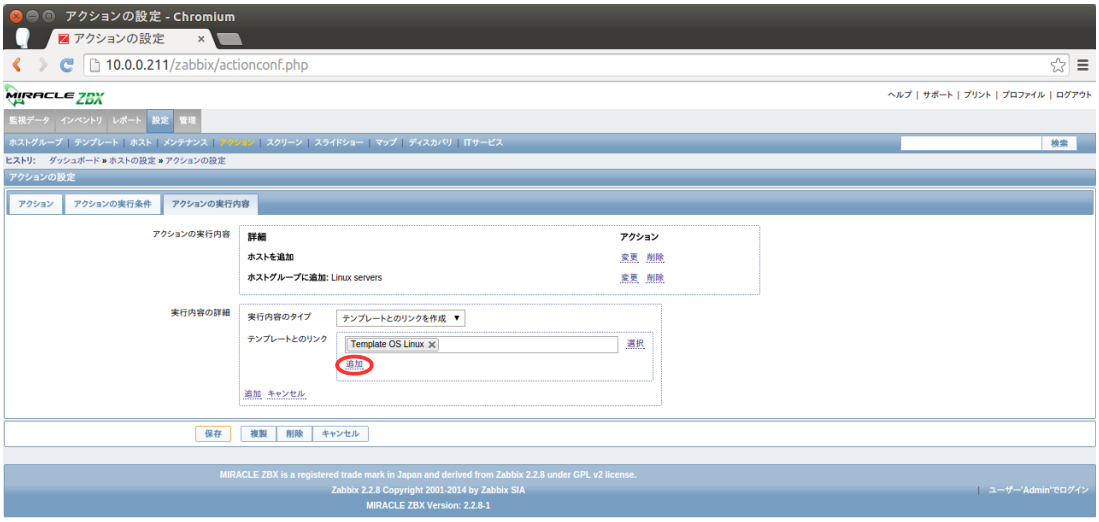


(13) ドロップダウン「実行内容のタイプ」にて「テンプレートとのリンクを作成」を選択します。次にフレーム「テンプレートとのリンク」内のリンク「選択」をクリックします。

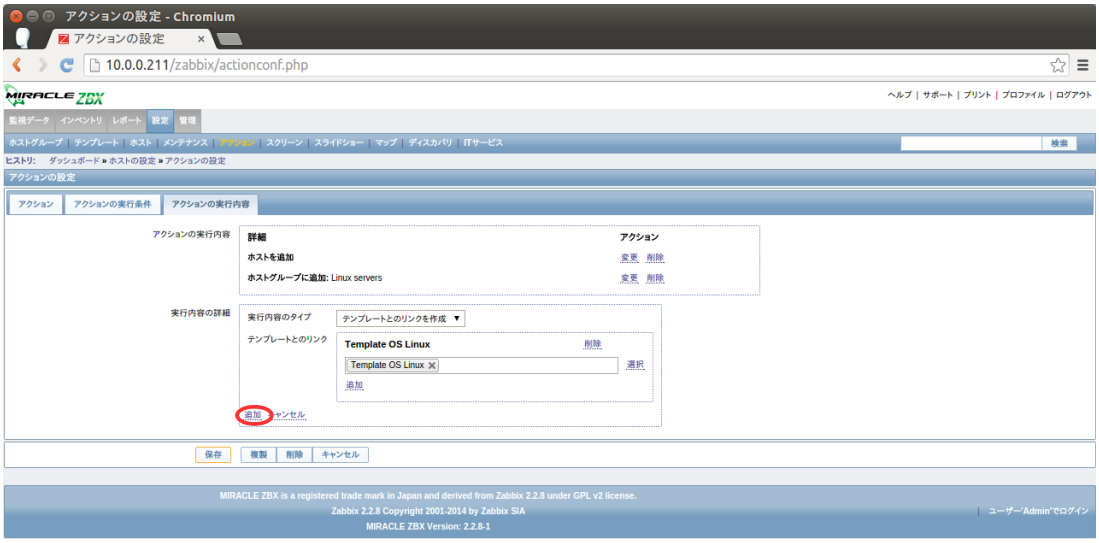


(14) ポップアップウィンドウ「テンプレート」が表示されます。「Template OS Linux」行にチェックを入れ、ウィンドウ最下部のボタン「選択」をクリックします。

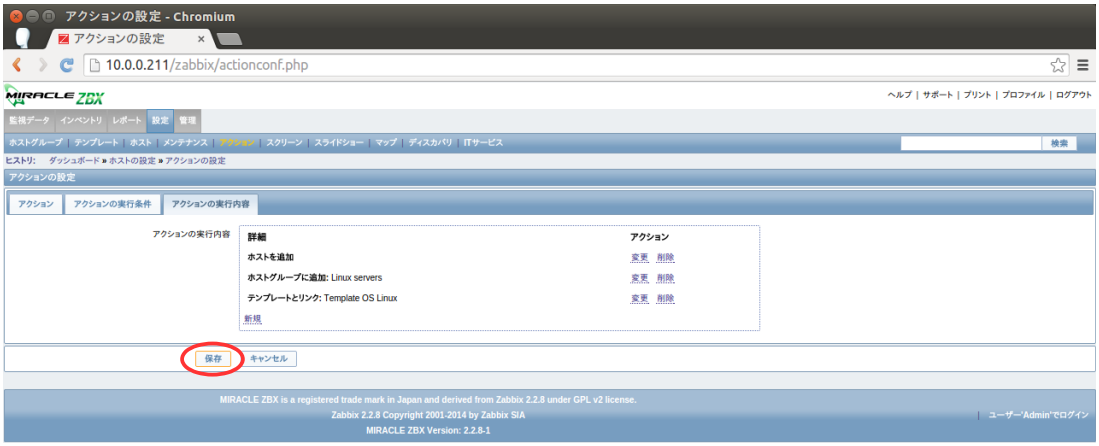
(15) フレーム「テンプレートとのリンク」内のリンク「追加」をクリックします。



(16) フレーム「実行内容の詳細」内のリンク「追加」をクリックします。



(17) 画面下方のボタン「保存」をクリックします。



(18) イベントソース「自動登録」のアクション一覧に、自動的に画面が移動します。背景が緑色の行「アクションを追加しました」が表示されていることを確認してください。

(19) 数分経過したのちメニュー「設定→ホスト」を表示すると、下図のようにホスト「zabbixagent」が追加されていることを確認できます。

5.4.3 OpenStack 環境の監視項目追加・変更

本項では、既に稼働している MIRACLE ZBX (または Zabbix)の監視対象に OpenStack の各ノードを追加する方法を説明します。

1. **OpenStack 全ノードへのパッケージ zabbix-agent の追加**

   全ノード共通で必要となる手順です。次のコマンドを実行し、パッケージ「zabbix-agent」を追加インストールします。

```
$ sudo apt-get install zabbix-agent
```

2. **OpenStack 全ノード上のファイル/etc/zabbix/zabbix_agentd.conf の編集**

   全ノード共通で必要となる手順です。パラメータ Hostname, Server, ServerActive をコメントアウトします。

```
# Hostname=Zabbix server
# Server=127.0.0.1
# ServerActive=127.0.0.1
```

3. **OpenStack 全ノードのファイル/etc/zabbix/zabbix_agentd.conf.d/zbx-server の作成**

   全ノード共通で必要となる手順です。次の 2 ファイルを作成します。

   /etc/zabbix/zabbix_agentd.conf.d/zbx-server

```
Server=ZBX Server の Floating IP
ServerActive=ZBX Server の Floating IP
```

   /etc/zabbix/zabbix_agentd.conf.d/hostname

```
Hostname=ホスト名
```

   ※2～3 については、/etc/zabbix/zabbix_agentd.conf の該当パラメータを直接変更しても構いません。
   また、以下の実行例では、パラメータ Hostname の値には、controller, network, compute1 のうち該当する名前を設定することを仮定しています。

4. **controller ノードへのアーカイブ展開**

   controller ノード上でのみ必要となる手順です。ファイル rabbitmq.queue.num-<version>.tar.gz を controller ノードへ転送した上、次のコマンドを実行します。

```
controller:$ sudo tar xzf rabbitmq.queue.num-<version>.tar.gz -C /
```

   上記コマンドで、次のファイルが生成されます。

   - /etc/zabbix/rabbitmq.queue.num
   - /etc/zabbix/zabbix_agentd.conf.d/rabbitmq
   - /etc/sudoers.d/zbx_rabbitmqctl

5. **全ノードでのサービス zabbix-agent の再起動**

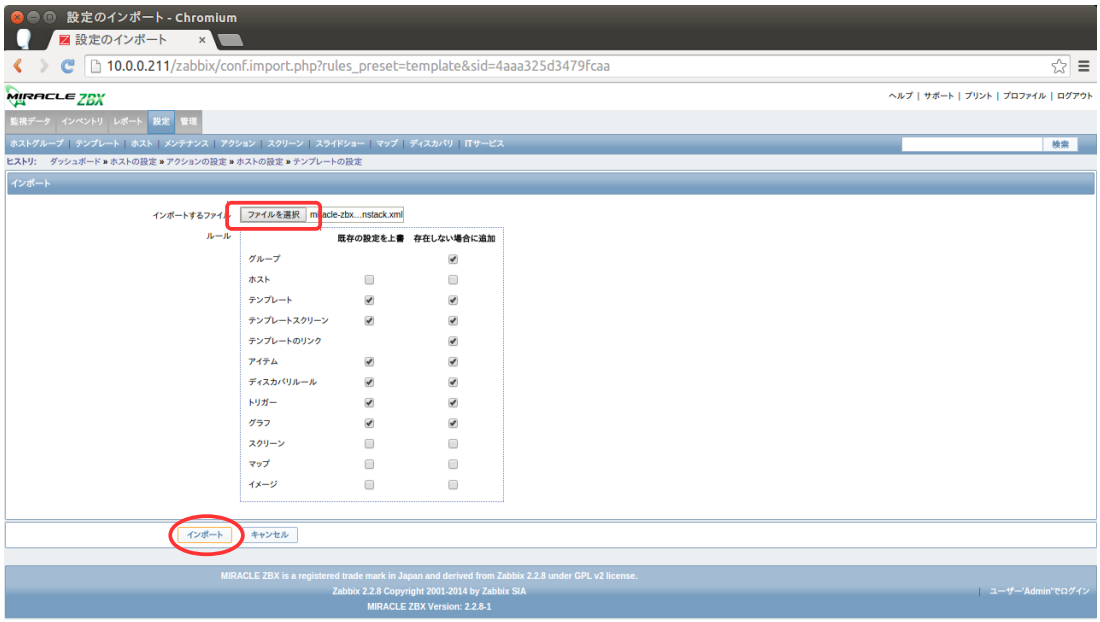
   全ノード共通で必要となる手順です。次のコマンドを実行し、サービス「zabbix-agent」を再起動します。

```
$ sudo service zabbix-agent restart
```

6. MIRACLE ZBX へのテンプレートのインポート

フロントエンドへログインし、メニュー「設定→テンプレート」を選択します。次に、画面右上のボタン「インポート」をクリックします。

ボタン「ファイルを選択」をクリックして「miracle-zbx-templates-openstack.xml」を選択し、ボタン「インポート」をクリックします。チェック項目に変更を加える必要はありません。



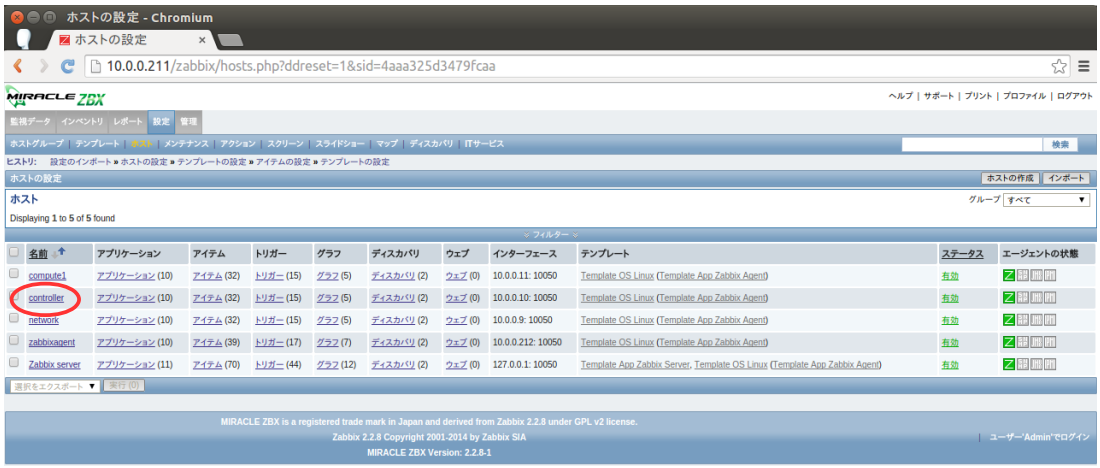
以上の操作で、次のテンプレートが生成されます。

- Template_OpenStack_Cinder
- Template_OpenStack_Common
- Template_OpenStack_Compute
- Template_OpenStack_Dashboard
- Template_OpenStack_Glance
- Template_OpenStack_Keystone
- Template_OpenStack_MySQL
- Template_OpenStack_Neutron_Controller
- Template_OpenStack_Neutron_Network
- Template_OpenStack_Nova_Controller
- Template_OpenStack_RabbitMQ
- Template_OpenStack_Swift-Proxy
- Template_OpenStack_Swift-Storage

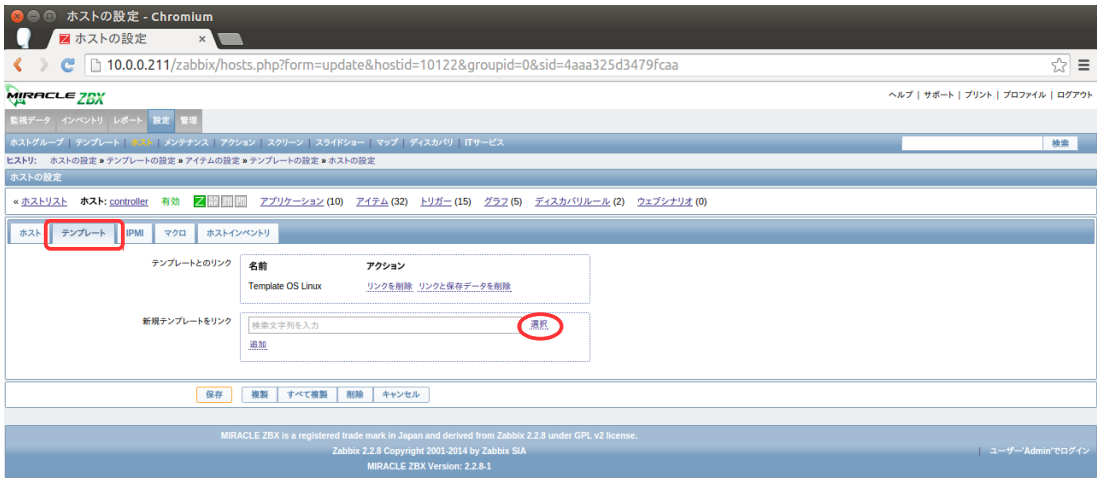
7. OpenStack 全ノードの登録およびテンプレート適用

OpenStack の各ノードは既に自動登録されています。各ノードに必要となる監視項目を含むテンプレートを適用します。

(1) 名前列のリンク「controller」をクリックします。

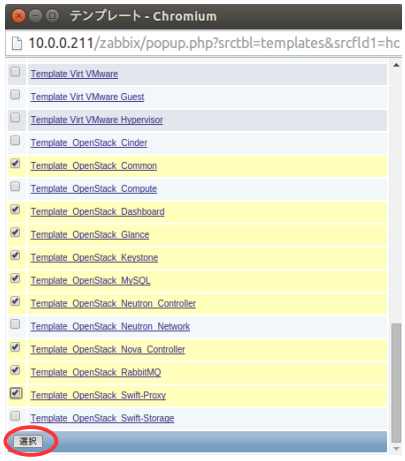


(2) タブ「テンプレート」に表示を切り替え、フレーム「新規テンプレートをリンク」内のリンク「選択」をクリックします。

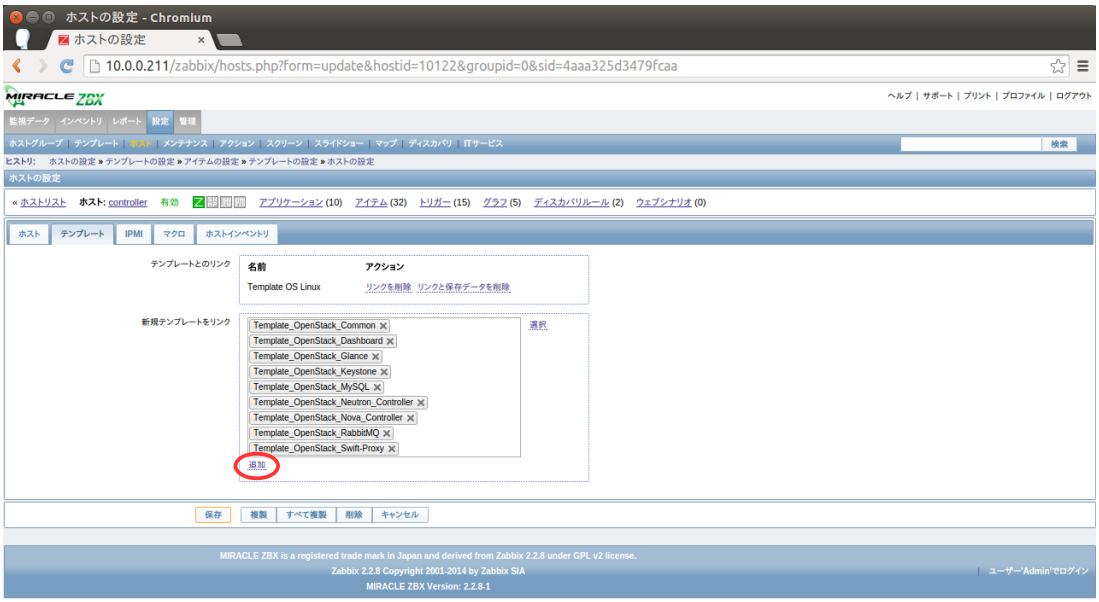


(3) ポップアップウィンドウ「テンプレート」が表示されます。次のテンプレート名の列にチェックを入れ、ボタン「選択」をクリックします。

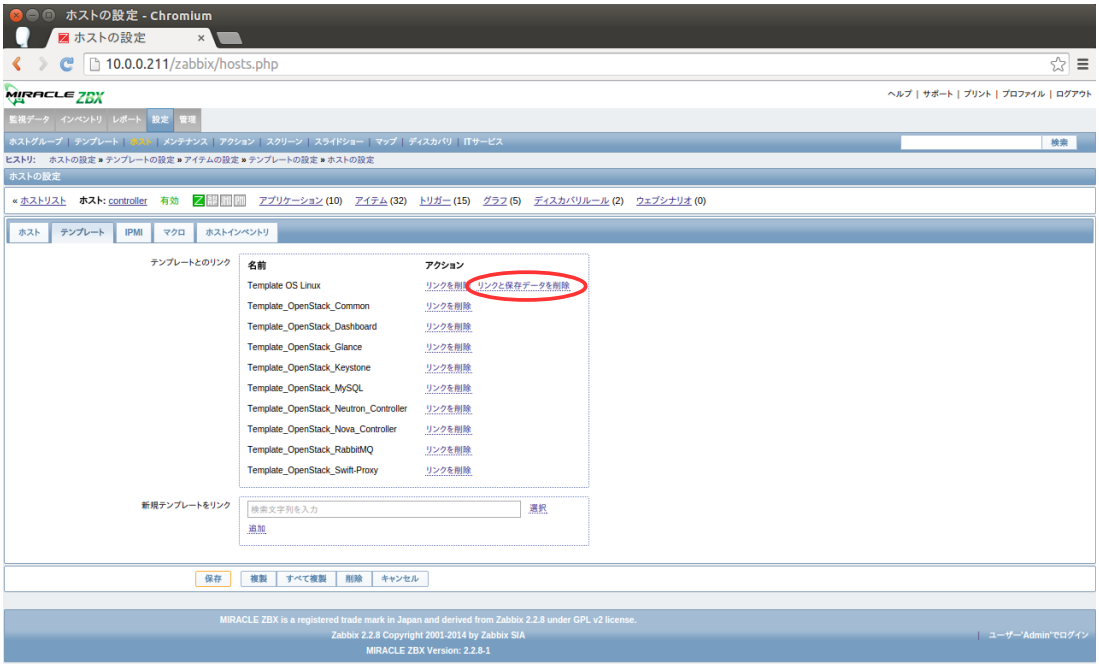
- Template_OpenStack_Cinder
- Template_OpenStack_Common
- Template_OpenStack_Dashboard
- Template_OpenStack_Glance
- Template_OpenStack_Keystone
- Template_OpenStack_MySQL
- Template_OpenStack_Neutron_Controller
- Template_OpenStack_Nova_Controller
- Template_OpenStack_RabbitMQ
- Template_OpenStack_Swift-Proxy



(4) フレーム「新規テンプレートをリンク」内のリンク「追加」をクリックします。

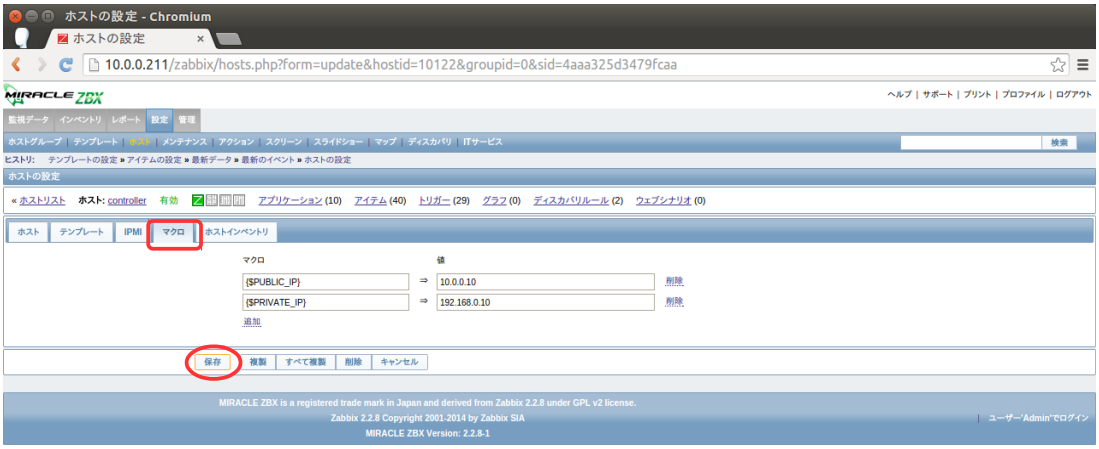


(5) フレーム「テンプレートとのリンク」内のテンプレート「Template OS Linux」行のリンク「リンクと保存データを削除」をクリックします。



(6) タブ「マクロ」をクリックします。次のマクロと値を入力し、ボタン「保存」をクリックします。

マクロ	値
{$PUBLIC_IP}	ext-net 側の IP アドレス（実行例: 10.0.0.10）
{$PRIVATE_IP}	demo-net 側の IP アドレス（実行例: 192.168.0.10）



以上の操作で controller ノードに必要となる監視項目が設定されます。その他のノードにも同様に対象となるテンプレートをリンクします。

なお、全ノードとリンク対象テンプレートとの関係は下表のとおりです。

ホスト (ノード)	テンプレート
controller	Template_OpenStack_Cinder Template_OpenStack_Common Template_OpenStack_Dashboard Template_OpenStack_Glance Template_OpenStack_Keystone Template_OpenStack_MySQL Template_OpenStack_Neutron_Controller Template_OpenStack_Nova_Controller Template_OpenStack_RabbitMQ Template_OpenStack_Swift-Proxy
network	Template_OpenStack_Common Template_OpenStack_Neutron_Network
computel	Template_OpenStack_Common Template_OpenStack_Compute Template_OpenStack_Swift-Storage

また、全ノードとマクロとの関係は下表のとおりです。

ホスト (ノード)	マクロ	値
controller	{$PUBLIC_IP}	ext-net 側の IP アドレス（実行例: 10.0.0.10）
	{$PRIVATE_IP}	demo-net 側の IP アドレス（実行例: 192.168.0.10）
network	{$PUBLIC_IP}	ext-net 側の IP アドレス（実行例: 10.0.0.9）
	{$PRIVATE_IP}	demo-net 側の IP アドレス（実行例: 192.168.0.9）
computel	{$PUBLIC_IP}	ext-net 側の IP アドレス（実行例: 10.0.0.11）
	{$PRIVATE_IP}	demo-net 側の IP アドレス（実行例: 192.168.0.11）

6 KVM ゲストの生成

compute1 ノード内に KVM の仮想ゲストとして Hatohol サーバーを構築する方法を説明します。

6.1　compute1 の eth1 設定変更

/etc/network/interfaces を編集します。

【変更前】

```
auto ethl
iface ethl inet static
        address 10.0.0.11
        netmask 255.255.255.0
        gateway 10.0.0.1
        dns-nameservers 10.0.0.1
```

【変更後】

```
auto ethl
iface ethl inet manual

auto br0
iface br0 inet static
        bridge_ports ethl
        bridge_maxwait 0
        bridge_df 0
        bridge_stp off
        address 10.0.0.11
        netmask 255.255.255.0
        gateway 10.0.0.1
        dns-nameservers 10.0.0.1
```

編集後、compute1 ノードで稼働中のインスタンスにログインし、OS のコマンドでシャットダウンした後に compute1 ノードを再起動します。

6.2　KVM 用パッケージの追加

KVM を動作させるために必要となるパッケージを追加します。

```
compute1:# apt-get install -y qemu-kvm libvirt-bin virtinst bridge-utils
```

6.3 Hatohol サーバーの構築

次に、キックスタートの機能を使用して Hatohol サーバーを構築します。hatohol-centos6.ks を compute1 ノード上にコピーし、次のコマンドを実行します。実行例はカレントディレクトリに hatohol-centos6.ks が配置されており、CPU 1 コア、メモリ 1GB を割り当てるものです。

なお、Hatohol サーバーの IP アドレス、ゲートウェイ、ネームサーバーは hatohol-centos6.ks 内に記述されています。変更する必要がある場合は、あらかじめ当該ファイルを修正した上で次のコマンドを実行する必要があります。

Hatohol サーバー自体を監視するために、MIRALCE ZBX サーバーを Hatohol サーバーと同一の仮想ゲスト上に起動する場合は、hatohol-centos6.ks の代わりに hatohol-zbx-centos6.ks を指定してください。

```
computel:$ sudo virt-install --name hatohol ¥
 --vcpus 1 --ram 1024 ¥
 --disk path=/var/lib/libvirt/images/Hatohol.img,size=8,sparse=false ¥
 --network bridge=br0  --graphics vnc --os-variant rhel6 ¥
 --location=http://ftp.iij.ad.jp/pub/linux/centos/6/os/x86_64 ¥
 --initrd-inject=hatohol-centos6.ks ¥
 --extra-args="ks=file:/hatohol-centos6.ks"
 --noautoconsole
```

インストール画面を表示させるには、virt-manager または virt-viewer からアタッチしてください。

インストール終了時にはシャットダウン状態となるので、次のコマンドで起動させます。virt-manager を使用している場合は、GUI の画面から起動させることが可能です。

```
computel:$ virsh start hatohol
```

Hatohol サーバー上のユーザ root のパスワードは centos と設定されています。

6.4　Hatohol の設定

MIRACLE ZBX または Zabbix サーバー、Ceilometer を Hatohol に登録する方法を説明します。

6.4.1　MIRACLE ZBX / Zabbix サーバーの追加

1.　Hatohol フロントエンドでのログイン

Web ブラウザで Hatohol サーバーの IP アドレス(初期値: 10.0.0.40)にアクセスします。

```
http://10.0.0.40/
```

次の値でログインすることができます。

項目	値
ユーザー名	admin
パスワード	hatohol

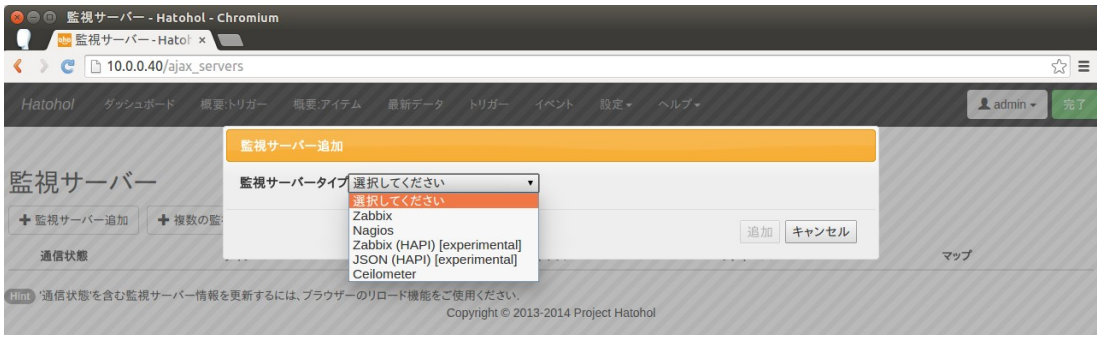
2.　監視サーバー画面への移動

メニュー「設定→監視サーバー」を選択します。

3. 監視サーバーの追加

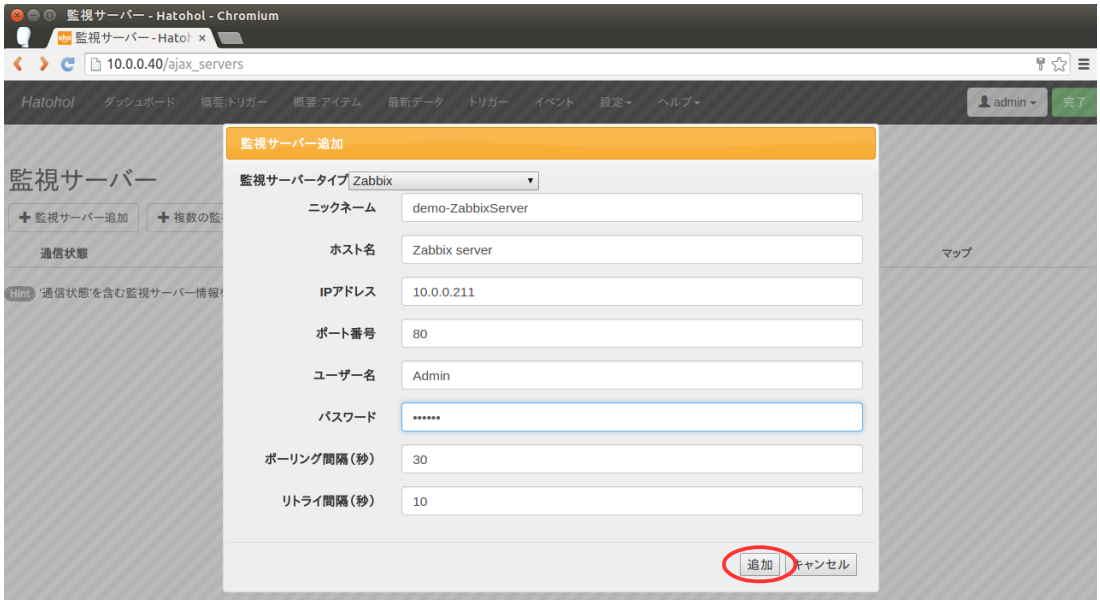
(1) ボタン「+監視サーバー追加」をクリックします。

(2) ダイヤログボックス「監視サーバー」が表示されます。ドロップダウン「監視サーバータイプ」にて「Zabbix」を選択します。



(3) Zabbix サーバーの登録に必要となる項目が表示されます。次の情報を入力します。入力後、ダイヤログボックス下方のボタン「追加」をクリックします。

項目	値
ニックネーム	任意（Hatohol での表示名）
ホスト名	Zabbix server（登録する MIRACLE ZBX, Zabbix サーバ上のホスト名）
IP アドレス	10.0.0.211（ZBX Server の Floating IP アドレス）
ポート番号	80（初期値）
ユーザー名	Admin
パスワード	zabbix
ポーリング間隔	30（初期値）
リトライ間隔	10（初期値）



以上で MIRACLE ZBX, Zabbix の登録は終了です。

なお、項番 6.3 にて hatohol-zbx-centos6.ks を指定して実行した場合は、Hatohol サーバーと同一仮想ゲスト上の MIRACLE ZBX を追加することが可能です。

6.4.2 Ceilometer の追加

(1) ボタン「+監視サーバー追加」をクリックします。

(2) ダイヤログボックス「監視サーバー追加」が表示されます。ドロップダウン「監視サーバータイプ」を選択し、登録に必要となる情報を入力します。以下の実行例では、テナント「admin」を対象としています。入力後、ダイヤログボックス下方のボタン「追加」をクリックします。

項目	値
ニックネーム	任意（Hatohol での表示名）
Keystone URL	http://controller:5000/v2.0
テナント名	admin
ユーザー名	admin
パスワード	password
ポーリング間隔	30（初期値）
リトライ間隔	10（初期値）
パッシブモード	チェックなし（初期値）
ブローカー URL	（初期値：空欄）
静的キューアドレス	（初期値：空欄）

以上で Ceilometer の登録は完了です。Hatohol は、Ceilometer に設定された alarm にしたがって監視します。

なお、同様の手順でテナント「demo」を追加することも可能です。

以下余白