Red Hat OpenStack Platform 10 DNS-as-a-Service ガイド

DNS 管理と Red Hat OpenStack Platform の統合

OpenStack Team

Red Hat OpenStack Platform 10 DNS-as-a-Service ガイド

DNS 管理と Red Hat OpenStack Platform の統合

OpenStack Team
rhos-docs@redhat.com

法律上の通知

Copyright © 2017 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

DNS と Red Hat OpenStack Platform の統合ガイド

目次

第1章 DNSAAS の概要 3
1.1. 本ガイドの構成 3
1.2. DNSAAS の要件 3
1.3. DNSAAS サービス 3
1.4. DNSAAS と COMPUTE および OPENSTACK NETWORKING の統合 4

第2章 DNSAAS の手動インストール 5

第3章 BIND9 のインストールおよび設定 10
3.1. BIND の基本インストール 10
3.2. BIND の設定 10
3.3. BIND の DNSAAS プールターゲットの設定 11
3.4. BIND のテスト 11
3.5. DNSAAS と BIND9 の統合のテスト 12
3.6. DNS レコードの自動生成の設定 (NOVA FIXED および NEUTRON FLOATING) 12
3.7. OPENSTACK NETWORKING の FLOATING IP レコード作成のテスト 13
3.8. OPENSTACK NETWORKING および COMPUTE DNS エントリーのクリーンアップ 13

第1章 DNSAAS の概要

Red Hat OpenStack Platform 10 には、Designate としても知られる DNS-as-a-Service (DNSaaS) のテクノロジープレビューが含まれています。DNSaaS にはドメインとレコードの管理のための REST API が含まれ、マルチテナントに対応しており、OpenStack Identity サービス (**keystone**) と統合して認証を行います。また、DNSaaS には Compute (**nova**) および OpenStack Networking (**neutron**) の通知と統合するフレームワークが実装されており、DNS レコードの自動生成が可能です。DNSaaS は Bind9 の統合もサポートしています。

1.1. 本ガイドの構成

- DNSaaS の手動インストール手順 (DNSaaS は現在 director デプロイメントに含まれていないため)
- コマンドラインインターフェースからの DNSaaS の管理および設定
- Bind9 との統合 (インスタンスレコードの自動作成を含む)

1.2. DNSAAS の要件

- 完全に機能する OpenStack Networking ベースの非高可用性 OpenStack 環境
- 自動作成のテスト用に読み込まれた OpenStack Image サービス (**glance**) のイメージ

1.3. DNSAAS サービス

DNSaaS のデプロイメントには、以下のコンポーネントが含まれます。

designate-api	OpenStack ネイティブの REST API を提供します。
designate-central	mysql データベースの要求を処理して、ストレージを連携します。
designate-mdns	標準の DNS プロトコルで他の DNS サーバーと通信するためだけに使用する小規模な MiniDNS サーバー
designate-pool-manager	DNSaaS が管理する DNS サーバーの状態を管理します。バックエンド DNS サーバーが DNSaaS と同期されるようにします。

designate-sink	nova および neutron 通知イベントをリッスンして、自動のレコード作成/削除をトリガーするために使用するオプションのサービス
designate-agent	ゾーン転送 (**AXFR**) を受け入れることができない DNS サーバーに使用します。BIND バックエンドの必要はありません。

注記

zone-manager サービスは、次のメジャーリリースで追加される予定です。このサービスはゾーンに対して定期的にタスクを実行して、失われたイベントを特定するメカニズムを提供します。

1.4. DNSAAS と COMPUTE および OPENSTACK NETWORKING の統合

DNSaaS のレコード管理は、**designate-sink** サービスが **designate-central** にメッセージを送信した時点で開始され、次に以下に記載のワークフローをトリガーします。

1. designate-sink は Compute から **instance boot/delete** イベントを受信するか、OpenStack Networking から **floating IP add/remove** イベントを受信します。これらのイベントは、OpenStack メッセージバスを使用して送信されます。

2. designate-sink は、仮想マシンの名前および設定済みのドメイン ID からホストの FQDN を構成します (以下参照)。

3. designate-sink は、**designate-central** に対して指定の名前および IP アドレスを持つレコードを追加/削除するように通知します。

4. designate-central は、DNSaaS データベースにレコードを追加したり、このデータベースからレコードを削除したりします (**designate-central** と **designate-mdns** の間で共有)。

5. designate-central は、このドメインのバックエンドの DNS サーバー (BIND9) に **DNS NOTIFY** を送信するように、**designate-pool-manager** に通知します。

6. バックエンドの DNS サーバーは **DNS NOTIFY** を受信して **AXFR** (ゾーン転送) 要求を **designate-mdns** に送信します。

7. designate-mdns は、データベースからこれらの変更を読み込み、**AXFR** 応答でバックエンド DNS サーバーに対してこれらの変更を送信します。

第2章 DNSAAS の手動インストール

1. コントローラー ノードに DNSaaS パッケージをインストールします。

```
# yum install openstack-designate-api openstack-designate-central
openstack-designate-sink openstack-designate-pool-manager openstack-
designate-mdns openstack-designate-common python-designate python-
designateclient openstack-designate-agent
```

2. DNSaaS および Pool Manager データベースを作成して、お使いの環境に合わせて **`IDENTIFIED BY 'ComplexAlphanumericPassword'`** の値を更新します。

```
# mysql -u root << EOF
CREATE DATABASE designate;
GRANT ALL ON designate.* TO 'designate'@'%' IDENTIFIED BY
'ComplexAlphanumericPassword';
GRANT ALL ON designate.* TO 'designate'@'localhost' IDENTIFIED BY
'ComplexAlphanumericPassword';
CREATE DATABASE designate_pool_manager;
GRANT ALL ON designate_pool_manager.* TO 'designate'@'%' IDENTIFIED BY
'ComplexAlphanumericPassword';
GRANT ALL ON designate_pool_manager.* TO 'designate'@'localhost'
IDENTIFIED BY 'ComplexAlphanumericPassword';
FLUSH PRIVILEGES;
quit
EOF
```

3. OpenStack Identity (keystone) で、DNSaaS サービスアカウントとエンドポイントを作成します。以下の例では、DNSaaS ホストの IP アドレスに **`192.168.100.20`** を使用します。これらのステップは、お使いの環境に合わせて変更する必要があります。

```
# openstack user create designate --password
ComplexAlphanumericPassword --email designate@localhost
# openstack role add --project services --user designate admin
# openstack service create dns --name designate --description
"Designate DNS Service"
# openstack endpoint create --region RegionOne designate public
http://192.168.100.20:9001
# openstack endpoint create --region RegionOne designate admin
http:/192.168.100.20:9001
# openstack endpoint create --region RegionOne designate internal
http://192.168.100.20:9001
```

4. DNSaaS のファイアウォールルールを追加します。

```
# iptables -I INPUT -p tcp -m multiport --dports 9001 -m comment --
comment "designate incoming" -j ACCEPT
# iptables -I INPUT -p tcp -m multiport --dports 5354 -m comment --
comment "Designate mdns incoming" -j ACCEPT
```

DNS をローカルでホストする場合は、必要なポートが開放されていることを確認してください。

```
# iptables -I INPUT -p tcp -m multiport --dports 953 -m comment --
comment "rndc incoming - bind only" -j ACCEPT
# service iptables save; service iptables restart
```

5. DNSaaS データベース接続を設定します。以下の手順では、DNSaaS ホストの IP アドレスが正しく入力されていることを確認します。**ComplexAlphanumericPassword** は、お使いの環境に合わせて値を変更してください。

```
# openstack-config --set /etc/designate/designate.conf
storage:sqlalchemy connection
mysql://designate:ComplexAlphanumericPassword@192.168.100.20/designate
# openstack-config --set /etc/designate/designate.conf
storage:sqlalchemy max_retries -1
# openstack-config --set /etc/designate/designate.conf
pool_manager_cache:sqlalchemy connection
mysql://designate:ComplexAlphanumericPassword@192.168.100.20/designate_
pool_manager
# openstack-config --set /etc/designate/designate.conf
pool_manager_cache:sqlalchemy max_retries -1
```

6. Identity サービス (**keystone**) に認証の設定を行います。お使いの環境に合わせて、**admin_password** オプションは調節するようにしてください。

```
# openstack-config --set /etc/designate/designate.conf
keystone_authtoken auth_uri http://192.168.100.20:5000/v2.0
# openstack-config --set /etc/designate/designate.conf
keystone_authtoken identity_uri http://192.168.100.20:35357/
# openstack-config --set /etc/designate/designate.conf
keystone_authtoken admin_tenant_name services
# openstack-config --set /etc/designate/designate.conf
keystone_authtoken admin_user designate
# openstack-config --set /etc/designate/designate.conf
keystone_authtoken admin_password ComplexAlphanumericPassword
```

7. RabbitMQ への DNSaaS 接続を設定します。

お使いの環境に合わせて、**rabbit_userid** と **rabbit_password** オプションを調節してください。

```
# openstack-config --set /etc/designate/designate.conf
oslo_messaging_rabbit rabbit_hosts 192.168.100.20:5672
# openstack-config --set /etc/designate/designate.conf
oslo_messaging_rabbit rabbit_ha_queues False
# openstack-config --set /etc/designate/designate.conf
oslo_messaging_rabbit rabbit_host 192.168.100.20
# openstack-config --set /etc/designate/designate.conf
oslo_messaging_rabbit rabbit_port 5672
# openstack-config --set /etc/designate/designate.conf
oslo_messaging_rabbit rabbit_userid amqp_user
# openstack-config --set /etc/designate/designate.conf
oslo_messaging_rabbit rabbit_password ComplexAlphanumericPassword
# openstack-config --set /etc/designate/designate.conf
oslo_messaging_rabbit rabbit_virtual_host /
```

8. DNSaaS の初期設定を追加します。

```
# openstack-config --set /etc/designate/designate.conf DEFAULT
notification_driver nova.openstack.common.notifier.rpc_notifier
# openstack-config --set /etc/designate/designate.conf DEFAULT
notification_driver messaging
# openstack-config --set /etc/designate/designate.conf DEFAULT
notification_topics notifications_designate
# openstack-config --set /etc/designate/designate.conf service:api
api_host 0.0.0.0
# openstack-config --set /etc/designate/designate.conf service:api
api_port 9001
# openstack-config --set /etc/designate/designate.conf service:api
auth_strategy keystone
# openstack-config --set /etc/designate/designate.conf service:api
enable_api_v1 True
# openstack-config --set /etc/designate/designate.conf service:api
enabled_extensions_v1 "diagnostics, quotas, reports, sync, touch"
# openstack-config --set /etc/designate/designate.conf service:api
enable_api_v2 True
# openstack-config --set /etc/designate/designate.conf service:api
enabled_extensions_v2 "quotas, reports"
```

9. Pool Manager を設定します。

注記

バックエンドが選択されていないため、プールターゲットの設定はまだ行いません。本手順で後ほど行います。

pool_id はハードコードされているため、以下に表示の **UUID** を使用してください。

```
# pool_id=794ccc2c-d751-44fe-b57f-8894c9f5c842
# nameserver_id=$(uuidgen)
# target_id=$(uuidgen)
# openstack-config --set /etc/designate/designate.conf
service:pool_manager pool_id $pool_id
# openstack-config --set /etc/designate/designate.conf pool:$pool_id
nameservers $nameserver_id
# openstack-config --set /etc/designate/designate.conf pool:$pool_id
targets $target_id
# openstack-config --set /etc/designate/designate.conf
pool_nameserver:$nameserver_id port 53
# openstack-config --set /etc/designate/designate.conf
pool_nameserver:$nameserver_id host 192.168.100.20
```

10. DNSaaS Sink を設定します。

注記

(まだドメインが存在しないため) Sink が使用するドメインはここでは設定しません。

```
# openstack-config --set /etc/designate/designate.conf service:sink
enabled_notification_handlers "nova_fixed, neutron_floatingip"
```

```
# openstack-config --set /etc/designate/designate.conf
handler:nova_fixed notification_topics notifications_designate
# openstack-config --set /etc/designate/designate.conf
handler:nova_fixed control_exchange nova
# openstack-config --set /etc/designate/designate.conf
handler:nova_fixed format "%(display_name)s.%(domain)s"
# openstack-config --set /etc/designate/designate.conf
handler:neutron_floatingip notification_topics notifications_designate
# openstack-config --set /etc/designate/designate.conf
handler:neutron_floatingip control_exchange neutron
# openstack-config --set /etc/designate/designate.conf
handler:neutron_floatingip format "%(octet0)s-%(octet1)s-%(octet2)s-%
(octet3)s.%(domain)s"
```

11. Compute および OpenStack Networking が通知を送信するように設定します。

注記

Ceilometer のエージェントも、通知をリッスンして消費します。特定の **`Designate`** 通知キュー (以下に表示) を作成して競合が起こらないようにします。

Kilo リリースでは、OpenStack Compute は通知プロバイダーとして **`messaging`** を使用します。以前は **`nova.openstack.common.notifier.rpc_notifier`** を使用していました。

```
# openstack-config --set /etc/nova/nova.conf DEFAULT
notification_topics notifications,notifications_designate
# openstack-config --set /etc/nova/nova.conf DEFAULT
notify_on_state_change vm_and_task_state
# openstack-config --set /etc/nova/nova.conf DEFAULT
instance_usage_audit_period hour
# openstack-config --set /etc/nova/nova.conf DEFAULT
instance_usage_audit true
# openstack-config --set /etc/neutron/neutron.conf DEFAULT
notification_driver neutron.openstack.common.notifier.rpc_notifier
# openstack-config --set /etc/neutron/neutron.conf DEFAULT
notification_topics notifications,notifications_designate
# openstack-service restart nova
# openstack-service restart neutron
```

12. 手動で **nova.conf** の **`notification_driver`** を確認します。

注記

nova.conf に複数の **`notification_drivers`** が含まれている可能性があるため、**openstack-config** コマンドで問題が発生する可能性があります。**`DEFAULT`** セクションで、以下の 2 つのエントリーが含まれていることを確認します。

```
notification_driver=ceilometer.compute.nova_notifier
notification_driver=messaging
```

注記

別のコンピュートノードを使用する場合には、**nova.conf** で以下の設定が必要となります。

```
notification_driver
=nova.openstack.common.notifier.rabbit_notifier,ceilometer.compu
te.nova_notifier
notification_driver =messaging
notification_topics=notifications,notifications_designate
```

13. DNSaaS と Pool Manager のキャッシュを同期します。

```
# designate-manage database sync
# designate-manage pool-manager-cache sync
```

14. DNSaaS サービスを有効化して、起動します。

```
# systemctl enable designate-central
# systemctl enable designate-api
# systemctl enable designate-mdns
# systemctl enable designate-pool-manager
# systemctl start designate-central
# systemctl start designate-api
# systemctl start designate-mdns
# systemctl start designate-pool-manager
```

注記

この時点では、プールの DNS ターゲットは作成していないため、DNSaaS デプロイメントは正常に機能しない可能性があります。

第3章 BIND9 のインストールおよび設定

以下のステップでは、Bind9 をインストールして DNSaaS と統合するように設定します。

3.1. BIND の基本インストール

1. BIND パッケージをインストールします。

```
# yum install bind bind-utils
```

2. named が受信接続をリッスンするように設定します。

```
# cp /etc/named.conf /etc/named.conf.orig
# sed -i -e "s/listen-on port.*/listen-on port 53 { 127.0.0.1;
192.168.100.20; };/" /etc/named.conf
```

3.2. BIND の設定

1. /etc/rndc.key に以下を書き込みます。

```
# rndc-confgen -a
```

2. `options` の前に以下を追加します。

```
# sed -i '/^options.*/i \
include "/etc/rndc.key"; \
controls { \
        inet 127.0.0.1 allow { localhost; } keys { "rndc-key"; }; \
};' /etc/named.conf
```

3. 既存のオプションをいくつか削除します (これらは後ほど再度記述します)。

```
# sed -i '/allow-query.*/d' /etc/named.conf
# sed -i '/recursion.*/d' /etc/named.conf
```

4. `options` の後に以下を追加します。

```
# sed -i '/^options.*/a \
        allow-new-zones yes; \
        allow-query { any; }; \
        recursion no;' /etc/named.conf
```

5. rndc 設定を作成します。コンピュートノードでは、rndc の設定が DNS サーバーをポイントする必要があります。以下に例を示します。

```
# cat << EOF > /etc/rndc.conf
include "/etc/rndc.key";
options {
        default-key "rndc-key";
```

```
        default-server 192.168.100.20;
        default-port 953;
};
EOF
```

6. named 設定をレビューします。

```
# named-checkconf /etc/named.conf
```

7. ファイルのパーミッションを修正します。

```
# setsebool -P named_write_master_zones on
# chmod g+w /var/named
# chown named:named /etc/rndc.conf
# chown named:named /etc/rndc.key
# chmod 600 /etc/rndc.key
```

8. named サービスを有効化して、起動します。

```
# systemctl enable named
# systemctl start named
```

9. named および **rndc** を検証します。

```
# dig @localhost localhost
# rndc status
```

3.3. BIND の DNSAAS プールターゲットの設定

1.プールターゲット設定を指定します。

```
# openstack-config --set /etc/designate/designate.conf
pool_target:$target_id type bind9
# openstack-config --set /etc/designate/designate.conf
pool_target:$target_id options "rndc_host: 192.168.100.20, rndc_port:
953, rndc_config_file: /etc/rndc.conf, rndc_key_file: /etc/rndc.key"
# openstack-config --set /etc/designate/designate.conf
pool_target:$target_id masters 192.168.100.20:5354
```

2. DNSaaS を再起動して、プールの変更を適用します。

```
# systemctl restart designate-api
# systemctl restart designate-central
# systemctl restart designate-mdns
# systemctl restart designate-pool-manager
# systemctl restart designate-sink
```

3.4. BIND のテスト

1. 以下の診断コマンドを実行します。

```
# netstat -tap | grep named
# netstat -tulpn | grep 53
# dig @192.168.100.20
```

2. DNSaaS ログのエラーをチェックします。Sink を設定していないため、今のところ Sink のエラーは無視します。

```
# cd /var/log/designate
# tail api.log
# tail central.log
# tail mdns.log
# tail pool-manager.log
# tail sink.log
```

3.5. DNSAAS と BIND9 の統合のテスト

1. サーバー用のエントリーを作成します。

```
# designate server-create --name $(hostname).
```

2. 前のステップで作成した DNS サーバーレコードを検証します。

```
# designate server-list
```

3. ドメインを作成します (**--name** オプションの最後の **.** を忘れないようにしてください)。

```
# designate domain-list
# designate domain-create --name example.com. --email root@example.com
# DOMAINID=$(designate domain-list | grep example.com | awk '{print
$2}')
```

注記

BIND に対して designate からドメインを作成する場合は、基本的に以下のようなコマンドを実行します。

```
# rndc -s 192.168.122.41 -p 953 -c /etc/rndc.conf -k /etc/rndc.key
addzone example.com '{ type slave; masters { 192.168.122.41 port 5354;
}; file "slave.example.com.ff532e15-55a9-4966-8f1e-b3eddb2891ba"; };'
```

4. レコードを作成してルックアップのテストを行います (**--name** オプションの最後の **.** を忘れないようにしてください)。

```
# designate record-create --name server1.example.com. --type A --data
1.2.3.4 $DOMAINID
# dig +short -p 53 @192.168.100.20 server1.example.com A
```

3.6. DNS レコードの自動生成の設定 (NOVA FIXED および NEUTRON FLOATING)

1. サンプルドメインの DNSaaS 設定を変更します。

```
# openstack-config --set /etc/designate/designate.conf
handler:nova_fixed domain_id $DOMAINID
# openstack-config --set /etc/designate/designate.conf
handler:neutron_floatingip domain_id $DOMAINID
# systemctl restart designate-api
# systemctl restart designate-central
# systemctl restart designate-mdns
# systemctl restart designate-pool-manager
# systemctl restart designate-sink
```

2. OpenStack Compute (nova) のレコード作成をテストします。

```
# glance image-list
# neutron net-list
# nova boot testserver --flavor m1.tiny --image cirros-0.3.4-x86_64 --
key-name yourkey --security-groups default --nic net-id=<Private Net
ID>
```

3. Sink ログをチェックします。

通知が正しく取得されている場合には、インスタンスが起動すると、**`create_record`** エントリーが表示されるはずです。

```
# tail /var/log/designate/sink.log
```

BIND で確認します。

```
# dig +short @192.168.100.20 testserver.example.com
```

これが機能しない場合には **`/var/named`** のファイルを確認することもできます。

3.7. OPENSTACK NETWORKING の FLOATING IP レコード作成のテスト

1. 以下の診断コマンドを実行します (**`pubnet1`** は環境に適した名前に置き換えます)。

```
# FLOATINGIP=$(neutron floatingip-create pubnet1 | grep
floating_ip_address | awk '{print $4}')
# nova add-floating-ip testserver $FLOATINGIP
# DNSRESULT=$(echo $FLOATINGIP |sed 's/\./-/g').example.com
# dig +short @192.168.100.20 $DNSRESULT
```

2. ログファイルで **`create_record`** のイベントが確認できるはずです。

```
# tail /var/log/designate/sink.log
```

3.8. OPENSTACK NETWORKING および COMPUTE DNS エントリーのクリーンアップ

1. 前のステップで作成したテスト用の Floating IP を削除します。

```
# nova remove-floating-ip testserver $FLOATINGIP
```

2. ログファイルで **delete_record** のイベントが確認できるはずです。

```
# tail /var/log/designate/sink.log
```

これで、レコードが削除されたはずです。

3. 前のステップで作成した **testserver** を削除します。

```
# designate record-list $DOMAINID
# nova delete testserver
```

ログファイルで別の **delete_record** エントリーが確認できるはずです。

```
# tail /var/log/designate/sink.log
```