Red Hat OpenStack Platform 16.2

OpenStack Identity と外部のユーザー管理サービスの統合

Active Directory または Red Hat Identity Management を外部認証バックエンドとして使用する方法

Last Updated: 2022-01-17

Red Hat OpenStack Platform 16.2 OpenStack Identity と外部のユーザー管理サービスの統合

Active Directory または Red Hat Identity Management を外部認証バックエンドとして使用する方法

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

OpenStack Identity (keystone) サービスを Microsoft Active Directory Domain Service (AD DS)、Red Hat Identity Management (IdM)、および LDAP と統合します。

目次

前書き

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、弊社 の CTO、Chris Wright のメッセージ を参照してください。

RED HAT ドキュメントへのフィードバック (英語のみ)

弊社ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。

ドキュメントへのダイレクトフィードバック (DDF) 機能の使用 (英語版のみ)

特定の文章、段落、またはコードブロックに対して直接コメントを送付するには、DDF の **Add Feedback** 機能を使用してください。なお、この機能は英語版のドキュメントでのみご利用いただけます。

1. **Multi-page HTML** 形式でドキュメントを表示します。
2. ドキュメントの右上隅に **Feedback** ボタンが表示されていることを確認してください。
3. コメントするテキスト部分をハイライト表示します。
4. **Add Feedback** をクリックします。
5. **Add Feedback** フィールドにコメントを入力します。
6. (オプション) ドキュメントチームが連絡を取り問題についてお伺いできるように、ご自分のメールアドレスを追加します。
7. **Submit** をクリックします。

第1章 OPENSTACK IDENTITY (KEYSTONE) と ACTIVE DIRECTORY の統合

OpenStack Identity (keystone) と Microsoft Active Directory Domain Service (AD DS) を統合することができます。Identity サービスは、特定の Active Directory Domain Services (AD DS) ユーザーを認証しますが、承認設定および重要なサービスアカウントは Identity サービスデータベースに保持されます。その結果、Identity サービスは、ユーザーアカウントの認証用に AD DS に読み取り専用でアクセスし、認証されたアカウントに割り当てられた権限の管理を継続します。

Identity サービスを AD DS と統合することで、AD DS ユーザーは Red Hat OpenStack Platform (RHOSP) に対して認証してリソースにアクセスできるようになります。Identity サービスや Image サービスなどの RHOSP サービスアカウントや承認管理は、Identity サービスのデータベースに残ります。パーミションとロールは、Identity サービスの管理ツールを使用して AD DS アカウントに割り当てられます。

OpenStack Identity と Active Directory を統合するプロセスには、以下の段階が含まれています。

1. Active Directory の認証情報を設定し、LDAPS 証明書をエクスポートする
2. OpenStack に LDAPS 証明書をインストールおよび設定する
3. 1 つまたは複数の LDAP バックエンドを使用するように director を設定する
4. Active Directory バックエンドにアクセスするようにコントローラーノードを設定する
5. OpenStack プロジェクトへの Active Directory ユーザーまたはグループのアクセスを設定する
6. ドメインおよびユーザー一覧が正しく作成されていることを確認する
7. (オプション) 管理者以外のユーザーの認証情報ファイルを作成する

1.1. ACTIVE DIRECTORY 認証情報の設定

Active Directory Domain Service (AD DS) が OpenStack Identity と統合するように設定するには、Identity サービスが使用する LDAP アカウントを設定し、Red Hat OpenStack Platform ユーザーのユーザーグループを作成し、Red Hat OpenStack Platform のデプロイメントで使用する LDAPS 証明書の公開鍵をエクスポートします。

前提条件

- Active Directory ドメインサービスが設定済みで、稼働していること。
- Red Hat OpenStack Platform が設定済みで、稼働していること。
- DNS 名前解決が完全に機能しており、かつ全ホストが適切に登録されていること。
- AD DS 認証トラフィックが LDAPS で暗号化され、ポート 636 を使用していること。
- 推奨: 単一の障害点を避けるために、高可用性または負荷分散ソリューションを備えた AD DS を実装していること。

手順

Active Directory サーバーで以下の手順を実行します。

1. LDAP ルックアップアカウントを作成します。このアカウントは、Identity サービスが AD DS LDAP サービスにクエリーを実行するのに使用されます。

```
PS C:\> New-ADUser -SamAccountName svc-ldap -Name "svc-ldap" -GivenName LDAP -
Surname Lookups -UserPrincipalName svc-ldap@lab.local  -Enabled $false -
PasswordNeverExpires $true -Path 'OU=labUsers,DC=lab,DC=local'
```

2. このアカウントのパスワードを設定し、有効にします。AD ドメインのパスワードの複雑さの要件を満たすパスワードを指定するように要求されます。

```
PS C:\> Set-ADAccountPassword svc-ldap -PassThru | Enable-ADAccount
```

3. **grp-openstack** という名前の RHOSP ユーザーグループを作成します。OpenStack Identity でパーミッションを割り当てることができるのは、このグループのメンバーのみです。

```
PS C:\> NEW-ADGroup -name "grp-openstack" -groupscope Global -path
"OU=labUsers,DC=lab,DC=local"
```

4. プロジェクトグループを作成します。

```
PS C:\> NEW-ADGroup -name "grp-openstack-demo" -groupscope Global -path
"OU=labUsers,DC=lab,DC=local"
PS C:\> NEW-ADGroup -name "grp-openstack-admin" -groupscope Global -path
"OU=labUsers,DC=lab,DC=local"
```

5. **svc-ldap** ユーザーを **grp-openstack** グループに追加します。

```
PS C:\> ADD-ADGroupMember "grp-openstack" -members "svc-ldap"
```

6. AD ドメインコントローラーから、**証明書 MMC** を使用して、DER で暗号化された **x509** .cer ファイルとして LDAPS 証明書の (秘密鍵ではなく) 公開鍵 をエクスポートします。このファイルを RHOSP 管理者に送信します。

7. AD DS ドメインの NetBIOS 名の取得

```
PS C:\> Get-ADDomain | select NetBIOSName
NetBIOSName
-----------
LAB
```

この値を RHOSP 管理者に送信します。

1.2. ACTIVE DIRECTORY LDAPS 証明書のインストール

OpenStack Identity (keystone) は、LDAPS クエリーを使用してユーザーアカウントを検証します。このトラフィックを暗号化するために、keystone は **keystone.conf** で定義されている証明書ファイルを使用します。LDAPS 証明書を設定するには、Active Directory から受け取った公開鍵を **.crt** 形式に変換し、その証明書を keystone が参照できる場所にコピーします。

注記

LDAP 認証に複数のドメインを使用する場合、**Unable to retrieve authorized projects** または **Peer's Certificate issuer is not recognized** など、さまざまなエラーが発生する可能性があります。これは、keystone が特定ドメインに誤った証明書を使用すると発生する可能性があります。回避策として、すべての LDAPS 公開鍵を単一の **.crt** バンドルにマージし、このファイルを使用するようにすべての keystone ドメインを設定します。

前提条件

- Active Directory の認証情報が設定されている。
- LDAPS 証明書が Active Directory からエクスポートされている。

Procedure

1. OpenStack Identity を実行中のノードに、LDAPS 公開鍵をコピーし、**.cer** から **.crt** に変換します。この例では、**addc.lab.local.cer** とうい名前の元の証明書ファイルを使用しています。

```
# openssl x509 -inform der -in addc.lab.local.cer -out addc.lab.local.crt
# cp addc.lab.local.crt /etc/pki/ca-trust/source/anchors
```

2. (オプション) **ldapsearch** などの診断のコマンドを実行する必要がある場合には、RHEL の証明書ストアに証明書を追加する必要もあります。

   a. **.cer** から **.pem** に変換します。この例では、**addc.lab.local.cer** とうい名前の元の証明書ファイルを使用しています。

```
# openssl x509 -inform der -in addc.lab.local.cer -out addc.lab.local.pem
```

   b. コントローラーノードに **.pem** をインストールします。たとえば、Red Hat Enterprise Linux の場合は以下を実行します。

```
# cp addc.lab.local.pem /etc/pki/ca-trust/source/anchors/
# update-ca-trust
```

1.3. ドメイン固有の LDAP バックエンドを使用する DIRECTOR の設定

director が 1 つ以上の LDAP バックエンドを使用するように設定するには、heat テンプレートで **KeystoneLDAPDomainEnable** フラグを **true** に設定し、各 LDAP バックエンドに関する情報が含まれる環境ファイルを設定します。次に、director は keystone ドメインごとに別の LDAP バックエンドを使用します。

注記

ドメイン設定ファイルのデフォルトのディレクトリーは **/etc/keystone/domains/** に設定されています。**keystone::domain_config_directory** hiera キーを使用して環境ファイル内に **ExtraConfig** パラメーターを追加して必要なパスを設定することによってオーバーライドすることができます。

Procedure

1. デプロイメントの heat テンプレートで、**KeystoneLDAPDomainEnable** フラグを **true** に設定します。これにより、**identity** 設定グループ内の keystone に **domain_specific_drivers_enabled** オプションが設定されます。

2. **tripleo-heat-templates** に **KeystoneLDAPBackendConfigs** パラメーターを設定して、LDAP バックエンド設定の仕様を追加します。その後、必要な LDAP オプションを指定できます。

3. **keystone_domain_specific_ldap_backend.yaml** 環境ファイルのコピーを作成します。

```
$ cp /usr/share/openstack-tripleo-heat-
templates/environments/services/keystone_domain_specific_ldap_backend.yaml
/home/stack/templates/
```

4. **/home/stack/templates/keystone_domain_specific_ldap_backend.yaml** 環境ファイルを編集して、デプロイメントに適した値を設定します。たとえば、以下のパラメーターは、**testdomain** という名前の keystone ドメイン向けの LDAP 設定を作成します。

```
parameter_defaults:
  KeystoneLDAPDomainEnable: true
  KeystoneLDAPBackendConfigs:
    testdomain:
      url: ldaps://192.0.2.250
      user: cn=openstack,ou=Users,dc=director,dc=example,dc=com
      password: RedactedComplexPassword
      suffix: dc=director,dc=example,dc=com
      user_tree_dn: ou=Users,dc=director,dc=example,dc=com
      user_filter: "(memberOf=cn=OSuser,ou=Groups,dc=director,dc=example,dc=com)"
      user_objectclass: person
      user_id_attribute: cn
```

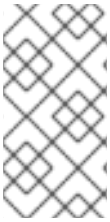
5. (オプション) 環境ファイルにドメインをさらに追加します。以下に例を示します。

```
KeystoneLDAPBackendConfigs:
  domain1:
    url: ldaps://domain1.example.com
    user: cn=openstack,ou=Users,dc=director,dc=example,dc=com
    password: RedactedComplexPassword
    ...
  domain2:
    url: ldaps://domain2.example.com
    user: cn=openstack,ou=Users,dc=director,dc=example,dc=com
    password: RedactedComplexPassword
    ...
```

これにより、**domain1** と **domain2** という名前の 2 つのドメインが指定され、各ドメインには、異なる LDAP ドメインが独自の設定で適用されます。

1.4. コントローラーノードでの OPENSTACK IDENTITY ドメインの設定

外部のユーザー管理サービスと統合する OpenStack Identity (keystone) を実行するコントローラーノードを設定するには、まず LDAP 認証を使用するように SELinux を設定し、コントローラーノードに **domains** ディレクトリーを作成します。次に、OpenStack Identity が複数のバックエンドを使用するように設定します。また、Dashboard が複数のドメインを使用するように設定します。



注記

director を使用している場合には、以下の手順で参照されている設定ファイルが Puppet によって管理されている点に注意してください。このため、**openstack overcloud deploy** コマンドを実行するたびに、自分で追加したカスタム設定が上書きされる可能性があります。

プランニング

設定ファイルを更新する場合には、特定の OpenStack サービスはコンテナー内で実行されるようになったことを認識する必要があります。これは、keystone、nova、cinder などのサービスが対象です。そのため、考慮すべき特定の管理プラクティスがいくつかあります。

- 物理ノードのホストオペレーティングシステム上の設定ファイル (例: **/etc/cinder/cinder.conf**) は更新しないでください。コンテナー化されたサービスはこのようなファイルを参照しません。
- コンテナー内で実行されている設定ファイルは更新しないでください。コンテナーを再起動すると変更が失われてしまいます。
  代わりに、コンテナー化されたサービスに変更を加える必要がある場合は、コンテナーの生成に使用される設定ファイルを更新する必要があります。これらのファイルは **/var/lib/config-data/puppet-generated/** 内に保管されています。

  以下に例を示します。

  - keystone: **/var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf**
  - cinder: **/var/lib/config-data/puppet-generated/cinder/etc/cinder/cinder.conf**
  - nova: **/var/lib/config-data/puppet-generated/nova/etc/nova/nova.conf**

  変更内容は、サービスを再起動した後に適用されます。例: **sudo systemctl restart tripleo_keystone**

Procedure

OpenStack Identity (keystone) サービスを実行するそれぞれのコントローラーノードで、この手順を実施します。

1. SELinux を設定します。

```
# setsebool -P authlogin_nsswitch_use_ldap=on
```

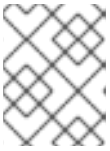
   出力には、以下のようなメッセージが含まれている場合がありますが、これは無視できます。

```
Full path required for exclude: net:[4026532245].
```

2. **domains** ディレクトリーを作成します。

```
# mkdir /var/lib/config-data/puppet-generated/keystone/etc/keystone/domains/
# chown 42425:42425 /var/lib/config-data/puppet-generated/keystone/etc/keystone/domains/
```

3. keystone が複数のバックエンドを使用するように設定します。



注記

dnf install crudini を使用して **crudini** をインストールする必要がある場合があります。

```
# crudini --set /var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf identity domain_specific_drivers_enabled true
# crudini --set /var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf identity domain_config_dir /etc/keystone/domains
# crudini --set /var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf assignment driver sql
```

4. Dashboard で複数のドメインを有効にします。以下の行を **/var/lib/config-data/puppet-generated/horizon/etc/openstack-dashboard/local_settings** に追加します。

```
OPENSTACK_API_VERSIONS = {
    "identity": 3
}
OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = 'Default'
```

注記

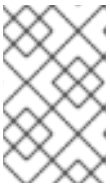
director を使用している場合には、**/var/lib/config-data/puppet-generated/horizon/etc/openstack-dashboard/local_settings** が Puppet によって管理されている点に注意してください。このため、カスタムの設定を追加しても、**openstack overcloud deploy** プロセスを実行するたびに上書きされる可能性があります。上書きされた場合には、この設定を毎回手動で追加し直す必要がある場合があります。

horizon コンテナーを再起動して、設定を適用します。

```
$ sudo systemctl restart tripleo_horizon
```

5. 前のステップでドメイン名として取得した NetBIOS 名の値で、外部サービスとの統合用の keystone ドメインを作成します。このアプローチでは、ログインプロセス中に、一貫したドメイン名をユーザーに提示することができます。以下の例では、**LAB** は、Identity サービスドメインとして使用する NetBIOS の名前に置き換えます。

```
$ openstack domain create LAB
```



注記

このコマンドが使用できない場合には、**# source overcloudrc-v3** のコマンドを実行して、コマンドラインセッションでの keystone v3 へのアクセスが有効化されているかどうかを確認します。

6. 統合する外部サービスの設定ファイルを作成します。

   - Active Directory Domain Service (AD DS):**/var/lib/config-data/puppet-generated/keystone/etc/keystone/domains/keystone.LAB.conf** (**LAB** は、前のステップで取得した NetBIOS 名に置き換えます) という名前の新規ファイルに LDAP 設定を入力し

ます。以下の設定例は、実際に使用する LDAP デプロイメントに合わせて編集する必要があります。

```
[ldap]
url                = ldaps://addc.lab.local:636
user               = CN=svc-ldap,OU=labUsers,DC=lab,DC=local
password           = RedactedComplexPassword
suffix             = DC=lab,DC=local
user_tree_dn       = OU=labUsers,DC=lab,DC=local
user_objectclass   = person
user_filter        = (|(memberOf=cn=grp-openstack,OU=labUsers,DC=lab,DC=local)
(memberOf=cn=grp-openstack-admin,OU=labUsers,DC=lab,DC=local)
(memberOf=memberOf=cn=grp-openstack-demo,OU=labUsers,DC=lab,DC=local))
user_id_attribute  = sAMAccountName
user_name_attribute = sAMAccountName
user_mail_attribute = mail
user_pass_attribute =
user_enabled_attribute = userAccountControl
user_enabled_mask  = 2
user_enabled_default = 512
user_attribute_ignore = password,tenant_id,tenants
group_objectclass  = group
group_tree_dn      = OU=labUsers,DC=lab,DC=local
group_filter       = (CN=grp-openstack*)
group_id_attribute = cn
group_name_attribute = name
use_tls            = False
tls_cacertfile     =/etc/pki/ca-trust/source/anchors/anchorsaddc.lab.local.pem

query_scope        = sub
chase_referrals    = false

[identity]
driver = ldap
```

各設定項目について説明します。

設定	説明
url	認証に使用する AD Domain Controller。LDAPS ポート **636** を使用します。
user	LDAP クエリーに使用する AD アカウントの **識別名**。たとえば、**Get-ADuser svc-ldap | select DistinguishedName** を使用する AD 内の **svc-ldap** アカウントの **識別名** の値を特定することができます。
password	上記で使用した AD アカウントのパスワード (プレーンテキスト形式)。

設定	説明
suffix	AD ドメインの **識別名**。この値は、**Get-ADDomain | select DistinguishedName** を使用して特定することができます。
user_tree_dn	OpenStack アカウントを含む **組織単位** (OU)。
user_objectclass	LDAP ユーザーの種別を定義します。AD には **person** の種別を使用します。
user_filter	Identity サービスに対して提示するユーザーをフィルタリングします。その結果、**grp-openstack** グループのメンバーのみに Identity サービスで定義されているパーミッションを付与することができます。この値には、グループの **完全な識別名** が必要です:**Get-ADGroup grp-openstack | select DistinguishedName**
user_id_attribute	ユーザー ID に使用する AD 値をマッピングします。
user_name_attribute	**names** に使用する AD 値をマッピングします。
user_mail_attribute	ユーザーのメールアドレスに使用する AD 値をマッピングします。
user_pass_attribute	この値は空白のままにします。
user_enabled_attribute	アカウントが有効にされているかどうかを検証する AD の設定。
user_enabled_mask	アカウントが有効化されているかを判断するために確認すべき値を定義します。ブール値が返されない場合に使用します。
user_enabled_default	アカウントが有効化されていることを示す AD 値。
user_attribute_ignore	Identity サービスが無視する必要のあるユーザー属性を定義します。
group_objectclass	**groups** に使用する AD 値をマッピングします。
group_tree_dn	そのユーザーグループを含む **組織単位** (OU)。

設定	説明
group_filter	Identity サービスに提示するグループをフィルタリングします。
group_id_attribute	グループ ID に使用する AD 値をマッピングします。
group_name_attribute	グループ名に使用する AD 値をマッピングします。
use_tls	TLS を使用するかどうかを定義します。STARTTLS ではなく LDAPS で暗号化する場合には、無効にする必要があります。
tls_cacertfile	**.crt** 証明書ファイルへのパスを指定します。
query_scope	**grp-openstack** グループに所属するユーザーを特定する場合に、Identity サービスがネスト化された子 OU 内での検索ができるように設定します。
chase_referrals	**false** に設定します。この設定により **python-ldap** が匿名のアクセスによる全参照を追跡しないようにします。

- Red Hat Identity Manager (IdM): **/var/lib/config-data/puppet-generated/keystone/etc/keystone/domains/keystone.LAB.conf** (**LAB** は、前のステップで作成したドメイン名に置き換えます) という名前の新規ファイルに LDAP 設定を入力します。以下の設定例は、実際に使用する IdM デプロイメントに合わせて編集する必要があります。

```
[ldap]
url =  ldaps://idm.lab.local
user = uid=svc-ldap,cn=users,cn=accounts,dc=lab,dc=local
user_filter = (memberOf=cn=grp-openstack,cn=groups,cn=accounts,dc=lab,dc=local)
password = RedactedComplexPassword
user_tree_dn = cn=users,cn=accounts,dc=lab,dc=local
user_objectclass = inetUser
user_id_attribute = uid
user_name_attribute = uid
user_mail_attribute = mail
user_pass_attribute =
group_tree_dn              = cn=groups,cn=accounts,dc=lab,dc=local
group_objectclass            = groupOfNames
group_id_attribute          = cn
group_name_attribute       =  cn
group_member_attribute  = member
group_desc_attribute       = description
use_tls                = False
query_scope                = sub
chase_referrals              = false
```

```
tls_cacertfile =/etc/pki/ca-trust/source/anchors/anchorsca.crt

[identity]
driver = ldap
```

各設定項目について説明します。

設定	説明
url	認証に使用する IdM サーバー。LDAPS ポート **636** を使用します。
user	LDAP クエリーに使用する IdM 内のアカウント。
password	上記で使用した IdM アカウントのパスワード (プレーンテキスト形式)。
user_filter	Identity サービスに対して提示するユーザーをフィルタリングします。その結果、**grp-openstack** グループのメンバーのみに Identity サービスで定義されているパーミッションを付与することができます。
user_tree_dn	IdM 内の OpenStack アカウントへのパス。
user_objectclass	LDAP ユーザーの種別を定義します。IdM には **inetUser** の種別を使用します。
user_id_attribute	ユーザー ID に使用する IdM 値をマッピングします。
user_name_attribute	**names** に使用する IdM 値をマッピングします。
user_mail_attribute	ユーザーのメールアドレスに使用する IdM 値をマッピングします。
user_pass_attribute	この値は空白のままにします。

注記

IdM グループとの統合で返されるのは直接のメンバーだけで、ネスト化されたグループは返されません。したがって、**LDAP_MATCHING_RULE_IN_CHAIN** または **memberof:1.2.840.113556.1.4.1941:** に依存するクエリーは、現在 IdM では機能しません。

7. 設定ファイルの所有権を keystone ユーザーに変更します。

```
# chown 42425:42425 /var/lib/config-data/puppet-
generated/keystone/etc/keystone/domains/keystone.LAB.conf
```

8. keystone サービスを再起動して、変更を適用します。

```
# sudo systemctl restart tripleo_keystone
```

関連資料

- 「OpenStack Identity ドメインへの管理ユーザーアクセス権の付与」
- 「ドメイン固有の LDAP バックエンドを使用する director の設定」

1.5. OPENSTACK IDENTITY ドメインへの管理ユーザーアクセス権の付与

admin ユーザーが OpenStack Identity (keystone) ドメインにアクセスして **Domain** タブを表示するのを許可するには、ドメインと **admin** ユーザーの ID を取得した後、ドメインのユーザーに **admin** ロールを割り当てます。

注記

これにより、OpenStack admin アカウントには外部サービスドメインのパーミッションは付与されません。この場合には、**ドメイン** という用語は、OpenStack が使用する keystone ドメインのことを指しています。

Procedure

この手順では、**LAB** ドメインを使用します。ドメイン名は、設定するドメインの実際の名前に置き換えます。

1. **LAB** ドメインの ID を取得します。

```
$ openstack domain show LAB
+---------+----------------------------------+
| Field   | Value                            |
+---------+----------------------------------+
| enabled | True                             |
| id      | 6800b0496429431ab1c4efbb3fe810d4 |
| name    | LAB                              |
+---------+----------------------------------+
```

2. **default** ドメインから **admin** ユーザーの ID を取得します。

```
$ openstack user list --domain default | grep admin
| 3d75388d351846c6a880e53b2508172a | admin      |
```

3. **admin** ロールの ID を取得します。

```
$ openstack role list
```

出力は、統合する外部サービスによって異なります。

- Active Directory Domain Service (AD DS):

```
+----------------------------------+-----------------+
| ID                               | Name            |
+----------------------------------+-----------------+
| 01d92614cd224a589bdf3b171afc5488 | admin           |
| 034e4620ed3d45969dfe8992af001514 | member          |
| 0aa377a807df4149b0a8c69b9560b106 | ResellerAdmin   |
| 9369f2bf754443f199c6d6b96479b1fa | heat_stack_user |
| cfea5760d9c948e7b362abc1d06e557f | reader          |
| d5cb454559e44b47aaa8821df4e11af1 | swiftoperator   |
| ef3d3f510a474d6c860b4098ad658a29 | service         |
+----------------------------------+-----------------+
```

- Red Hat Identity Manager (IdM):

```
+----------------------------------+---------------+
| ID                               | Name          |
+----------------------------------+---------------+
| 544d48aaffde48f1b3c31a52c35f01f9 | SwiftOperator |
| 6d005d783bf0436e882c55c62457d33d | ResellerAdmin |
| 785c70b150ee4c778fe4de088070b4cf | admin         |
| 9fe2ff9ee4384b1894a90878d3e92bab | _member_      |
+----------------------------------+---------------+
```

4. ドメインおよび admin の ID を使用して、keystone **LAB** ドメインの **admin** ロールに **admin** ユーザーを追加するコマンドを構築します。

```
# openstack role add --domain 6800b0496429431ab1c4efbb3fe810d4 --user
3d75388d351846c6a880e53b2508172a 785c70b150ee4c778fe4de088070b4cf
```

1.6. RED HAT OPENSTACK PLATFORM プロジェクトへの外部グループアクセス権の付与

複数の認証されたユーザーが Red Hat OpenStack Platform (RHOSP) リソースにアクセスできるようにするには、外部ユーザー管理サービスから特定のグループを承認し、RHOSP プロジェクトへのアクセス権限を付与します。この場合、OpenStack 管理者は各ユーザーをプロジェクト内のロールに手動で割り当てる必要はありません。その結果、これらのグループのすべてのメンバーは、事前に決定したプロジェクトにアクセスできます。

前提条件

- 外部サービスの管理者が以下の手順を完了していることを確認してください。
  - **grp-openstack-admin** という名前のグループを作成する。
  - **grp-openstack-demo** という名前のグループを作成する。
  - 必要に応じて、RHOSP ユーザーをこれらのグループの 1 つに追加する。
  - ユーザーを **grp-openstack** グループに追加する。
- OpenStack Identity ドメインを作成します。この手順では、**LAB** ドメインを使用します。

- RHOSP プロジェクトを作成するか、選択します。以下の手順では、**openstack project create --domain default --description "Demo Project" demo** コマンドで作成した **demo** という名前のプロジェクトを使用します。

Procedure

1. OpenStack Identity ドメインからユーザーグループの一覧を取得します。

```
# openstack group list --domain LAB
```

コマンド出力は、統合する外部のユーザー管理サービスによって異なります。

- Active Directory Domain Service (AD DS):

```
+------------------------------------------------------------------+--------------------+
| ID                                                               | Name               |
+------------------------------------------------------------------+--------------------+
| 185277be62ae17e498a69f98a59b66934fb1d6b7f745f14f5f68953a665b8851 | grp-
openstack       |
| a8d17f19f464c4548c18b97e4aa331820f9d3be52654aa8094e698a9182cbb88 | grp-
openstack-admin |
| d971bb3bd5e64a454cbd0cc7af4c0773e78d61b5f81321809f8323216938cae8 | grp-
openstack-demo  |
+------------------------------------------------------------------+--------------------+
```

- Red Hat Identity Manager (IdM):

```
+------------------------------------------------------------------+--------------------+
| ID                                                               | Name               |
+------------------------------------------------------------------+--------------------+
| 185277be62ae17e498a69f98a59b66934fb1d6b7f745f14f5f68953a665b8851 | grp-
openstack       |
| a8d17f19f464c4548c18b97e4aa331820f9d3be52654aa8094e698a9182cbb88 | grp-
openstack-admin |
| d971bb3bd5e64a454cbd0cc7af4c0773e78d61b5f81321809f8323216938cae8 | grp-
openstack-demo  |
+------------------------------------------------------------------+--------------------+
```

2. ロールの一覧を取得します。

```
# openstack role list
```

コマンド出力は、統合する外部のユーザー管理サービスによって異なります。

- Active Directory Domain Service (AD DS):

```
+----------------------------------+----------------+
| ID                               | Name           |
+----------------------------------+----------------+
| 01d92614cd224a589bdf3b171afc5488 | admin          |
| 034e4620ed3d45969dfe8992af001514 | member         |
| 0aa377a807df4149b0a8c69b9560b106 | ResellerAdmin  |
| 9369f2bf754443f199c6d6b96479b1fa | heat_stack_user |
| cfea5760d9c948e7b362abc1d06e557f | reader         |
```

```
| d5cb454559e44b47aaa8821df4e11af1 | swiftoperator   |
| ef3d3f510a474d6c860b4098ad658a29 | service         |
+----------------------------------+-----------------+
```

- Red Hat Identity Manager (IdM):

```
+----------------------------------+---------------+
| ID                               | Name          |
+----------------------------------+---------------+
| 0969957bce5e4f678ca6cef00e1abf8a | ResellerAdmin |
| 1fcb3c9b50aa46ee8196aaaecc2b76b7 | admin         |
| 9fe2ff9ee4384b1894a90878d3e92bab | _member_      |
| d3570730eb4b4780a7fed97eba197e1b | SwiftOperator |
+----------------------------------+---------------+
```

3. ユーザーグループを上記のロールの 1 つまたは複数に追加して、プロジェクトへのアクセス権を付与します。たとえば、**grp-openstack-demo** グループのユーザーを **demo** プロジェクトの一般ユーザーに指定するには、統合する外部サービスに応じて、グループを **member** または **_member_** ロールに追加する必要があります。

   - Active Directory Domain Service (AD DS):

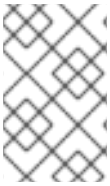
```
# openstack role add --project demo --group
d971bb3bd5e64a454cbd0cc7af4c0773e78d61b5f81321809f8323216938cae8  member
```

   - Red Hat Identity Manager (IdM):

```
$ openstack role add --project demo --group
d971bb3bd5e64a454cbd0cc7af4c0773e78d61b5f81321809f8323216938cae8
_member_
```

結果

grp-openstack-demo のメンバーは、ユーザー名とパスワードを入力し、**Domain** フィールドに **LAB** を入力して Dashboard にログインすることができます。



注記

ユーザーに **Error: Unable to retrieve container list.** というエラーメッセージが表示され、コンテナーの管理が可能であることが想定されている場合には、**SwiftOperator** ロールに追加する必要があります。

関連資料

- 「Red Hat OpenStack Platform プロジェクトへの外部ユーザーアクセス権の付与」

1.7. RED HAT OPENSTACK PLATFORM プロジェクトへの外部ユーザーアクセス権の付与

grp-openstack グループからの特定認証ユーザーに OpenStack リソースへのアクセス権限を付与するには、これらのユーザーに Red Hat OpenStack Platform (RHOSP) プロジェクトへの直接アクセス権限を付与できます。グループにアクセス権を付与する代わりに、個々のユーザーにアクセス権を付与する場合は、このプロセスを使用します。

前提条件

- 外部サービスの管理者が以下の手順を完了していることを確認してください。
  - RHOSP ユーザーを **grp-openstack** グループに追加する。
  - OpenStack Identity ドメインを作成する。この手順では、**LAB** ドメインを使用します。
- RHOSP プロジェクトを作成するか、選択します。以下の手順では、**openstack project create --domain default --description "Demo Project" demo** コマンドで作成した **demo** という名前のプロジェクトを使用します。

Procedure

1. OpenStack Identity ドメインからユーザーの一覧を取得します。

```
# openstack user list --domain LAB
 +----------------------------------------------------------------+----------------+
| ID                                                             | Name           |
+----------------------------------------------------------------+----------------+
| 1f24ec1f11aeb90520079c29f70afa060d22e2ce92b2eba7784c841ac418091e | user1          |
| 12c062faddc5f8b065434d9ff6fce03eb9259537c93b411224588686e9a38bf1 | user2          |
| afaf48031eb54c3e44e4cb0353f5b612084033ff70f63c22873d181fdae2e73c | user3          |
| e47fc21dcf0d9716d2663766023e2d8dc15a6d9b01453854a898cabb2396826e | user4
|
+----------------------------------------------------------------+----------------+
```

2. ロールの一覧を取得します。

```
# openstack role list
```

コマンド出力は、統合する外部のユーザー管理サービスによって異なります。

- Active Directory Domain Service (AD DS):

```
+----------------------------------+----------------+
```

```
| ID                               | Name           |
+----------------------------------+----------------+
| 01d92614cd224a589bdf3b171afc5488 | admin          |
| 034e4620ed3d45969dfe8992af001514 | member         |
| 0aa377a807df4149b0a8c69b9560b106 | ResellerAdmin  |
| 9369f2bf754443f199c6d6b96479b1fa | heat_stack_user |
| cfea5760d9c948e7b362abc1d06e557f | reader         |
| d5cb454559e44b47aaa8821df4e11af1 | swiftoperator  |
| ef3d3f510a474d6c860b4098ad658a29 | service        |
+----------------------------------+----------------+
```

- Red Hat Identity Manager (IdM):

```
+----------------------------------+---------------+
| ID                               | Name          |
+----------------------------------+---------------+
| 0969957bce5e4f678ca6cef00e1abf8a | ResellerAdmin |
| 1fcb3c9b50aa46ee8196aaaecc2b76b7 | admin         |
| 9fe2ff9ee4384b1894a90878d3e92bab | _member_      |
| d3570730eb4b4780a7fed97eba197e1b | SwiftOperator |
+----------------------------------+---------------+
```

3. 一覧表示されたロールの 1 つまたは複数にユーザーを追加して、RHOSP プロジェクトへのアクセス権を付与します。たとえば、**user1** を **demo** プロジェクトの一般ユーザーに指定するには、統合する外部サービスに応じて、ユーザーを **member** または **_member_** ロールに追加します。

   - Active Directory Domain Service (AD DS):

     ```
     # openstack role add --project demo --user
     1f24ec1f11aeb90520079c29f70afa060d22e2ce92b2eba7784c841ac418091e member
     ```

   - Red Hat Identity Manager (IdM):

     ```
     # openstack role add --project demo --user
     1f24ec1f11aeb90520079c29f70afa060d22e2ce92b2eba7784c841ac418091e _member_
     ```

4. **user1** を **demo** プロジェクトの管理ユーザーにするには、**admin** ロールに追加します。

   ```
   # openstack role add --project demo --user
   1f24ec1f11aeb90520079c29f70afa060d22e2ce92b2eba7784c841ac418091e admin
   ```

結果

user1 ユーザーは、外部ユーザー名とパスワードを入力し、**Domain** フィールドに **LAB** を入力して Dashboard にログインすることができます。

注記

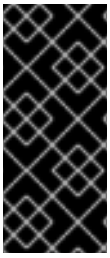
ユーザーに **Error: Unable to retrieve container list.** というエラーメッセージが表示され、コンテナーの管理が可能であることが想定されている場合には、**SwiftOperator** ロールに追加する必要があります。

関連資料

- 「Red Hat OpenStack Platform プロジェクトへの外部グループアクセス権の付与」

1.8. OPENSTACK IDENTITY ドメインおよびユーザーの一覧表示

利用可能なエントリーを表示するには、**openstack domain list** コマンドを使用します。Identity サービスに複数のドメインを設定すると、Dashboard のログインページに新しい **ドメイン** フィールドが有効になります。ユーザーは、ログイン認証情報にマッチするドメインを入力する必要があります。



重要

統合が完了したら、**Default** ドメインまたは新たに作成する keystone ドメインに新規プロジェクトを作成するかどうかを決定する必要があります。ワークフローとユーザーアカウントの管理方法を検討する必要があります。可能な場合には、**Default** ドメインを内部ドメインとして使用し、サービスアカウントと **admin** プロジェクトを管理し、外部ユーザーを別のドメインに維持します。

この例では、外部アカウントは **LAB** ドメインを指定する必要があります。**admin** のような組み込みの keystone アカウントには、ドメインに **Default** を指定する必要があります。

Procedure

1. ドメインの一覧を表示します。

```
# openstack domain list
+----------------------------------+---------+---------+-----------------------------------------------------------
--------------+
| ID                               | Name    | Enabled | Description                                               |
```

```
+----------------------------------+---------+---------+-----------------------------------------------------
--------------+
| 6800b0496429431ab1c4efbb3fe810d4 | LAB     | True    |
|
| default                          | Default | True    | Owns users and projects available on Identity API
v2. |
+----------------------------------+---------+---------+-----------------------------------------------------
--------------+
```

2. 特定のドメインのユーザー一覧を表示します。このコマンド例では、**--domain LAB** を指定し、**grp-openstack** グループのメンバーである LAB ドメイン内のユーザーを返します。

```
# openstack user list --domain LAB
```

--domain Default を追加して、組み込みの keystone アカウントを表示することもできます。

```
# openstack user list --domain Default
```

1.9. 非管理者ユーザーの認証情報ファイルの作成

OpenStack Identity のユーザーおよびドメインを設定したら、管理者以外のユーザーの認証情報ファイルを作成する必要がある場合があります。

手順

- 非管理者ユーザー用の認証情報 (RC) ファイルを作成します。この例では、ファイルで **user1** ユーザーを使用しています。

```
$ cat overcloudrc-v3-user1
# Clear any old environment that may conflict.
for key in $( set | awk '{FS="="}  /^OS_/ {print $1}' ); do unset $key ; done
export OS_USERNAME=user1
export NOVA_VERSION=1.1
export OS_PROJECT_NAME=demo
export OS_PASSWORD=RedactedComplexPassword
export OS_NO_CACHE=True
export COMPUTE_API_VERSION=1.1
export no_proxy=,10.0.0.5,192.168.2.11
export OS_CLOUDNAME=overcloud
export OS_AUTH_URL=https://10.0.0.5:5000/v3
export OS_AUTH_TYPE=password
export PYTHONWARNINGS="ignore:Certificate has no, ignore:A true
SSLContext object is not available"
export OS_IDENTITY_API_VERSION=3
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=LAB
```

1.10. OPENSTACK IDENTITY と外部のユーザー管理サービスの統合のテスト

OpenStack Identity (keystone) と Active Directory Domain Service (AD DS) が正常に統合されていることをテストするには、Dashboard 機能へのユーザーアクセスをテストします。

前提条件

- Active Directory (AD) または Red Hat Identity Manager (IdM) などの外部のユーザー管理サービスとの統合

Procedure

1. 外部のユーザー管理サービスにテストユーザーを作成し、そのユーザーを **grp-openstack** グループに追加します。
2. Red Hat OpenStack Platform で、**demo** プロジェクトの **_member_** ロールにユーザーを追加します。
3. AD テストユーザーの認証情報を使用して Dashboard にログインします。
4. 各タブをクリックし、エラーメッセージなしに正常に表示されているかどうかを確認します。
5. Dashboard を使用してテストインスタンスをビルドします。

注記

これらの手順で問題が発生した場合は、**admin** アカウントを使用して Dashboard にログインし、そのユーザーとして後続の手順を実施します。テストが成功した場合、OpenStack が予想通りに機能していること、および OpenStack Identity と Active Directory との統合設定のどこかに問題が存在することを意味します。

関連資料

- 「Active Directory との統合のトラブルシューティング」

1.11. ACTIVE DIRECTORY との統合のトラブルシューティング

OpenStack Identity で Active Directory との統合を使用する際にエラーが発生する場合には、LDAP コネクションをテストするか、証明書トラスト設定をテストする必要がある場合があります。LDAPS ポートにアクセスできることを確認する必要がある場合もあります。

注記

エラーのタイプおよび場所に応じて、以下の手順の関連ステップのみを実行します。

Procedure

1. **ldapsearch** コマンドを使用して Active Directory Domain Controller に対してテストクエリーをリモートで実行して、LDAP 接続をテストします。クエリーが成功した場合には、ネットワーク接続が機能しており、AD DS サービスが稼働中であることを確認できます。以下の例では、テストクエリーはサーバー **addc.lab.local** のポート**636** に対して実行されます。

```
# ldapsearch -Z -x -H ldaps://addc.lab.local:636 -D "svc-ldap@lab.local" -W -b
"OU=labUsers,DC=lab,DC=local" -s sub "(cn=*)" cn
```

注記

- **ldapsearch** は、**openldap-clients** パッケージに含まれています。このパッケージは、**# dnf install openldap-clients** のコマンドを実行するとインストールすることができます。
- このコマンドを実行すると、ホストオペレーティングシステム内で必要な証明書が特定されるはずです。

2. **ldapsearch** コマンドのテストの際に **Peer's Certificate issuer is not recognized.** というエラーを受け取った場合には、**TLS_CACERTDIR** パスが正しく設定されていることを確認してください。以下に例を示します。

```
TLS_CACERTDIR /etc/openldap/certs
```

3. 一時的な回避策として、証明書の検証を無効にすることを検討してください。

重要

この設定は、永続的には使用しないでください。

/etc/openldap/ldap.conf で、**TLS_REQCERT** パラメーターを **allow** に設定します。

```
TLS_REQCERT allow
```

この値を設定した後に **ldapsearch** クエリーが機能した場合には、証明書トラストが正しく設定されているかどうかをレビューする必要があります。

4. **nc** コマンドを使用して、LDAPS ポート **636** がリモートでアクセス可能であることを確認します。この例では、サーバー **addc.lab.local** に対してプローブを実行します。**ctrl-c** を押してプロンプトを終了します。

```
# nc -v addc.lab.local 636
Ncat: Version 6.40 ( http://nmap.org/ncat )
Ncat: Connected to 192.168.200.10:636.
^C
```

接続を確立できなかった場合には、ファイアウォールの設定に問題がある可能性があります。

第2章 OPENSTACK IDENTITY (KEYSTONE) と RED HAT IDENTITY MANAGER (IDM) の統合

OpenStack Identity (keystone) と Red Hat Identity Manager (IdM) を統合する場合、OpenStack Identity は特定の Red Hat Identity Management (IdM) ユーザーを認証しますが、承認設定および重要なサービスアカウントは Identity サービスデータベースに保持されます。この手順を実行すると、Identity サービスは、IdM に読み取り専用でアクセスしてユーザーアカウントの認証を行う一方で、認証されたアカウントに割り当てる権限を引き続き管理するようになります。**novajoin** を使用して、ノードを IdM に登録することもできます。

注記

この統合用の設定ファイルは、Puppet によって管理されます。このため、次回 **openstack overcloud deploy** コマンドを実行すると、自分で追加したカスタム設定が上書きされる可能性があります。設定ファイルを手動で編集するのではなく、director を使用して LDAP 認証を設定できます。

IdM 統合を計画して設定する前に、以下の主要な項目を確認してください。

- **認証**: パスワードを使用して、ユーザーが本人であることを検証するプロセス。
- **承認**: 認証されたユーザーに対して、アクセスしようとしているシステムの適切なパーミッションが付与されていることを確認するプロセス。
- **ドメイン**: Identity サービス内で設定する追加のバックエンド。たとえば、Identity サービスは、外部の IdM 環境内のユーザーを認証するように設定することができます。このように設定されたユーザーの集合は、**ドメイン** として考えることができます。

OpenStack Identity と IdM を統合するプロセスには、以下の段階が含まれています。

1. novajoin を使用して、アンダークラウドおよびオーバークラウドを IdM に登録する
2. Ansible を使用して、アンダークラウドおよびオーバークラウドに TLS-e を実装する
3. IdM サーバーの認証情報を設定し、LDAPS 証明書をエクスポートする
4. OpenStack に LDAPS 証明書をインストールおよび設定する
5. 1 つまたは複数の LDAP バックエンドを使用するように director を設定する
6. IdM バックエンドにアクセスするようにコントローラーノードを設定する
7. OpenStack プロジェクトへの IdM ユーザーまたはグループのアクセスを設定する
8. ドメインおよびユーザー一覧が正しく作成されていることを確認する
9. (オプション) 管理者以外のユーザーの認証情報ファイルを作成する

2.1. RED HAT IDENTITY MANAGER (IDM) との統合の計画

OpenStack Identity と Red Hat Identity Manager (IdM) の統合を計画する際には、両方のサービスが設定され稼動状態にあることを確認し、ユーザー管理、高可用性、およびファイアウォール設定に対する統合の影響を確認してください。

前提条件

- Red Hat Identity Management が設定済みで、稼働していること。
- Red Hat OpenStack Platform が設定済みで、稼働していること。
- DNS 名前解決が完全に機能しており、かつ全ホストが適切に登録されていること。

アクセス権限およびロール

この統合により、IdM ユーザーが OpenStack に対して認証を実行して、リソースにアクセスできるようになります。OpenStack のサービスアカウント (keystone、glance など) および承認管理 (パーミッションとロール) は Identity サービスのデータベースに残ります。パーミションとロールは、Identity サービスの管理ツールを使用して IdM アカウントに割り当てられます。

高可用性のオプション

この設定により、単一の IdM サーバーの可用性に依存するようになるため、Identity サービスがその IdM サーバー に対して認証できない場合には、プロジェクトユーザーが影響を受けることになります。このリスクを管理するオプションは複数あります。また、IdM サーバー の 1 つが利用できない場合には、keystone が異なる IdM サーバーにクエリーを実行するように設定することもできます。

停止の要件

- IdM バックエンドを追加するには、Identity サービスを再起動する必要があります。
- ユーザーは、IdM でアカウントが作成されるまでは、Dashboard にアクセスできません。ダウンタイムを短縮するには、この変更の前に十分余裕をもって IdM アカウントのプレステージを行うことを検討してください。

ファイアウォールの設定

ファイアウォールが IdM と OpenStack の間のトラフィックをフィルタリングしている場合には、以下のポートを介したアクセスを許可する必要があります。

ソース	送信先	タイプ	ポート
OpenStack コントローラーノード	Red Hat Identity Management	LDAPS	TCP 636

2.2. NOVAJOIN を使用した RED HAT IDENTITY MANAGER (IDM) へのノードの登録

novajoin は、デプロイメントプロセスの一部として、ノードを Red Hat Identity Manager (IdM) に登録するために使用するデフォルトのツールです。その結果、Red Hat OpenStack Platform デプロイメントで、ID、kerberos 認証情報、アクセス制御などの IdM の機能を統合することができます。残りの IdM 統合を続行する前に、登録プロセスを実行する必要があります。

登録プロセスには、以下の手順が含まれます。

1. アンダークラウドノードを認証局 (CA) に追加する
2. アンダークラウドノードを IdM に追加する
3. (オプション) オーバークラウドの DNS サーバーとして IdM サーバーを設定する
4. 環境ファイルを準備しオーバークラウドをデプロイする

5. IdM および RHOSP でオーバークラウドの登録をテストする
6. (オプション) IdM に novajoin の DNS エントリーを追加する

注記

現在、novajoin を使用した IdM の登録は、アンダークラウドとオーバークラウドのノードのみで利用可能です。オーバークラウドインスタンス向けの novajoin の統合は、今後のリリースでサポートされる見込みです。

2.2.1. 認証局へのアンダークラウドノードの追加

オーバークラウドをデプロイする前に、アンダークラウドノードに **python3-novajoin** パッケージをインストールし、**novajoin-ipa-setup** スクリプトを実行して、アンダークラウドを認証局 (CA) に追加します。

Procedure

1. アンダークラウドノードで、**python3-novajoin** パッケージをインストールします。

```
$ sudo dnf install python3-novajoin
```

2. アンダークラウドノードで **novajoin-ipa-setup** スクリプトを実行します。値はデプロイメントに応じて調整します。

```
$ sudo /usr/libexec/novajoin-ipa-setup \
    --principal admin \
    --password <IdM admin password> \
    --server <IdM server hostname> \
    --realm <realm> \
    --domain <overcloud cloud domain> \
    --hostname <undercloud hostname> \
    --precreate
```

ここで設定したワンタイムパスワード (OTP) を使用して、アンダークラウドを登録します。

2.2.2. Red Hat Identity Manager (IdM) へのアンダークラウドノードの追加

アンダークラウドノードを認証局 (CA) に追加したら、アンダークラウドを IdM に登録して novajoin を設定します。**undercloud.conf** ファイルの **[DEFAULT]** セクションで、以下の設定を行います。

Procedure

1. **novajoin** サービスを有効にします。

```
[DEFAULT]
enable_novajoin = true
```

2. アンダークラウドノードを IdM に登録できるように、ワンタイムパスワード (OTP) を設定します。

```
ipa_otp = <otp>
```

3. neutron の DHCP サーバーにより提供されるように、オーバークラウドのドメイン名を設定します。

```
overcloud_domain_name = <domain>
```

4. アンダークラウドのホスト名を設定します。

```
undercloud_hostname = <undercloud FQDN>
```

5. アンダークラウドのネームサーバーとして IdM を設定します。

```
undercloud_nameservers = <IdM IP>
```

6. より大規模な環境の場合には、novajoin の接続タイムアウト値を見直します。**undercloud.conf** ファイルで、**undercloud-timeout.yaml** という名前の新規ファイルへの参照を追加します。

```
hieradata_override = /home/stack/undercloud-timeout.yaml
```

undercloud-timeout.yaml に以下のオプションを追加します。タイムアウト値は秒単位で指定することができます (例: **5**)。

```
nova::api::vendordata_dynamic_connect_timeout: <timeout value>
nova::api::vendordata_dynamic_read_timeout: <timeout value>
```

7. **undercloud.conf** ファイルを保存します。

8. アンダークラウドのデプロイコマンドを実行して、既存のアンダークラウドに変更を適用します。

```
$ openstack undercloud install
```

2.2.3. オーバークラウドの DNS サーバーとしての Red Hat Identity Manager (IdM) の設定

IdM 環境を自動検出して、簡単に登録できるようにするには、IdM を DNS サーバーとして設定します。この手順はオプションですが、推奨されます。

手順

1. アンダークラウドに接続します。

```
$ source ~/stackrc
```

2. DNS ネームサーバーとして IdM を使用するようにコントロールプレーンサブネットを設定します。

```
$ openstack subnet set ctlplane-subnet --dns-nameserver  <idm_server_address>
```

3. IdM サーバーを使用するように環境ファイルの **DnsServers** パラメーターを設定します。

```
parameter_defaults:
  DnsServers: ["<idm_server_address>"]
```

このパラメーターは、通常カスタムの **network-environment.yaml** ファイルで定義されます。

2.2.4. 環境ファイルの準備と novajoin 登録によるオーバークラウドのデプロイ

IdM 統合でオーバークラウドをデプロイするには、環境ファイルを作成および編集し、オーバークラウドで定義するドメインに基づいて、カスタムドメインパラメーター **CloudDomain** および **CloudName** を使用するようにオーバークラウドを設定します。次に、すべての環境ファイルとデプロイメントに必要な追加の環境ファイルを指定して、オーバークラウドをデプロイします。

Procedure

1. **/usr/share/openstack-tripleo-heat-templates/environments/predictable-placement/custom-domain.yaml** 環境ファイルのコピーを作成します。

```
$ cp /usr/share/openstack-tripleo-heat-templates/environments/predictable-
placement/custom-domain.yaml \
  /home/stack/templates/custom-domain.yaml
```

2. **/home/stack/templates/custom-domain.yaml** 環境ファイルを編集して、デプロイメントに適した **CloudDomain** と **CloudName*** の値を設定します。

```
parameter_defaults:
  CloudDomain: lab.local
  CloudName: overcloud.lab.local
  CloudNameInternal: overcloud.internalapi.lab.local
  CloudNameStorage: overcloud.storage.lab.local
  CloudNameStorageManagement: overcloud.storagemgmt.lab.local
  CloudNameCtlplane: overcloud.ctlplane.lab.local
```

3. 環境に適した TLS の実装を選択します。

   - **enable-tls.yaml** 環境ファイルを使用して、カスタム証明書が含まれる外部エンドポイントを保護します。

     a. **/usr/share/openstack-tripleo-heat-templates/environments/ssl/enable-tls.yaml** を **/home/stack/templates** にコピーします。

     b. カスタム証明書および鍵が含まれるように **/home/stack/enable-tls.yaml** 環境ファイル変更します。

     c. 以下の環境ファイルをデプロイメントに追加して、内部および外部エンドポイントを保護します。

        - enable-internal-tls.yaml
        - tls-every-endpoints-dns.yaml
        - custom-domain.yaml
        - enable-tls.yaml

```
openstack overcloud deploy \
```

```
  --templates \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ssl/enable-
internal-tls.yaml \
  -e /usr/share/openstack-tripleo-heat-templates/environments/ssl/tls-
everywhere-endpoints-dns.yaml \
  -e /home/stack/templates/custom-domain.yaml \
  -e /home/stack/templates/enable-tls.yaml
```

- **haproxy-public-tls-certmonger.yaml** 環境ファイルを使用して、IdM が発行した証明書が含まれる外部エンドポイントを保護します。この実装では、novajoin が使用する VIP エンドポイントの DNS エントリーを作成する必要があります。

  a. novajoin が使用する VIP エンドポイントの DNS エントリーを作成する必要があります。**/home/stack/templates** のカスタム **network-environment.yaml** ファイルにあるオーバークラウドのネットワークを特定します。

```
parameter_defaults:
  ControlPlaneDefaultRoute: 192.168.24.1
  ExternalAllocationPools:
  - end: 10.0.0.149
    start: 10.0.0.101
  InternalApiAllocationPools:
  - end: 172.17.1.149
    start: 172.17.1.10
  StorageAllocationPools:
  - end: 172.17.3.149
    start: 172.17.3.10
  StorageMgmtAllocationPools:
  - end: 172.17.4.149
    start: 172.17.4.10
```

  b. heat テンプレート (例: **/home/stack/public_vib.yaml**) で、各オーバークラウドネットワークの仮想 IP アドレス の一覧を作成します。

```
parameter_defaults:
  ControlFixedIPs: [{'ip_address':'192.168.24.101'}]
  PublicVirtualFixedIPs: [{'ip_address':'10.0.0.101'}]
  InternalApiVirtualFixedIPs: [{'ip_address':'172.17.1.101'}]
  StorageVirtualFixedIPs: [{'ip_address':'172.17.3.101'}]
  StorageMgmtVirtualFixedIPs: [{'ip_address':'172.17.4.101'}]
  RedisVirtualFixedIPs: [{'ip_address':'172.17.1.102'}]
```

  c. それぞれの VIP について、DNS エントリーおよびゾーン (必要に応じて) を IdM に追加します。

```
ipa dnsrecord-add lab.local overcloud --a-rec 10.0.0.101
ipa dnszone-add ctlplane.lab.local
ipa dnsrecord-add ctlplane.lab.local overcloud --a-rec 192.168.24.101
ipa dnszone-add internalapi.lab.local
ipa dnsrecord-add internalapi.lab.local overcloud --a-rec 172.17.1.101
ipa dnszone-add storage.lab.local
ipa dnsrecord-add storage.lab.local overcloud --a-rec 172.17.3.101
ipa dnszone-add storagemgmt.lab.local
ipa dnsrecord-add storagemgmt.lab.local overcloud --a-rec 172.17.4.101
```

d. 以下の環境ファイルをデプロイメントに追加して、内部および外部エンドポイントを保護します。

- enable-internal-tls.yaml
- tls-everywhere-endpoints-dns.yaml
- haproxy-public-tls-certmonger.yaml
- custom-domain.yaml
- public_vib.yaml

```
openstack overcloud deploy \
  --templates \
   -e /usr/share/openstack-tripleo-heat-templates/environments/ssl/enable-internal-tls.yaml \
   -e /usr/share/openstack-tripleo-heat-templates/environments/ssl/tls-everywhere-endpoints-dns.yaml \
   -e /usr/share/openstack-tripleo-heat-templates/environments/services/haproxy-public-tls-certmonger.yaml \
   -e /home/stack/templates/custom-domain.yaml \
   -e /home/stack/templates/public-vip.yaml
```

注記

novajoin を使用して、既存のデプロイメントに TLS everywhere (TLS-e) を実装することはできません。

関連資料

- 「Ansible を使用した TLS-e の実装」

2.2.5. Red Hat Identity Manager (IdM) へのオーバークラウド登録のテスト

novajoin を使用して IdM へのアンダークラウドおよびオーバークラウドの登録を完了した後に、IdM でオーバークラウドノードを検索し、ホストエントリーに **Keytab:True** が含まれることを確認することで、登録が成功したことをテストできます。また、オーバークラウドノードにログインして、**sssd** コマンドを使用して IdM ユーザーにクエリーを行うことができることも確認できます。

1. IdM でオーバークラウドノードを特定し、ホストのエントリーに **Keytab:True** が含まれていることを確認します。

```
$ ipa host-show overcloud-node-01
  Host name: overcloud-node-01.lab.local
  Principal name: host/overcloud-node-01.lab.local@LAB.LOCAL
  Principal alias: host/overcloud-node-01.lab.local@LAB.LOCAL
  SSH public key fingerprint: <snip>
  Password: False
  Keytab: True
  Managed by: overcloud-node-01.lab.local
```

2. オーバークラウドノードにログインし、**sssd** を使用して IdM ユーザーにクエリーを行うことができることを確認します。たとえば、**susan** という名前の IdM ユーザーをクエリーするには、以下のコマンドを実行します。

```
$ getent passwd susan
uid=1108400007(susan) gid=1108400007(bob) groups=1108400007(susan)
```

2.3. ANSIBLE を使用した TLS-E の実装

Red Hat では、アンダークラウドおよびオーバークラウドに TLS-e を設定するのに、**novajoin** を使用するデフォルトの方式よりも、新たな Ansible ベースの **tripleo-ipa** を使用する方式を推奨しています。以下の手順を使用して、Red Hat OpenStack Platform の新規インストール、または TLS-e の設定が必要な既存のデプロイメントのいずれかに、TLS-e を実装することができます。事前にプロビジョニングされたノードに TLS-e を設定した Red Hat OpenStack Platform をデプロイする場合は、この方式を使用する必要があります。

注記

既存の環境に TLS-e を実装する場合は、引き続き **openstack undercloud install**、**openstack overcloud deploy** コマンド等のコマンドを実行する必要があります。これらの手順はべき等性を持ち、更新されたテンプレートおよび設定ファイルと一致するように既存のデプロイメント設定を調整するだけです。

2.3.1. アンダークラウドでの TLS-e の設定

前提条件

stack ユーザーの作成など、アンダークラウドの設定手順がすべて完了していること。詳細は、『Director Installation and Usage』を参照してください。

手順

1. ホストファイルを設定します。
   アンダークラウドの **/etc/resolv.conf** に、適切な検索ドメインおよびネームサーバーを設定します。たとえば、デプロイメントドメインが **example.com** で FreeIPA サーバーのドメインが **bigcorp.com** の場合、以下の行を /etc/resolv.conf に追加します。

   ```
   search example.com bigcorp.com
   nameserver $IDM_SERVER_IP_ADDR
   ```

2. 必要なソフトウェアをインストールします。

   ```
   sudo yum install -y python3-ipalib python3-ipaclient krb5-devel
   ```

3. ご自分の環境に固有の値で環境変数をエクスポートします。

   ```
   export IPA_DOMAIN=bigcorp.com
   export IPA_REALM=BIGCORP.COM
   export IPA_ADMIN_USER=$IPA_USER
   export IPA_ADMIN_PASSWORD=$IPA_PASSWORD
   export IPA_SERVER_HOSTNAME=ipa.bigcorp.com
   export UNDERCLOUD_FQDN=undercloud.example.com
   export USER=stack
   export CLOUD_DOMAIN=example.com
   ```

注記

IdM のユーザー認証情報は、新しいホストおよびサービスを追加できる管理ユーザーでなければなりません。

4. アンダークラウドで **undercloud-ipa-install.yaml** Ansible Playbook を実行します。

```
ansible-playbook \
--ssh-extra-args "-o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null" \
/usr/share/ansible/tripleo-playbooks/undercloud-ipa-install.yaml
```

5. undercloud.conf に以下のパラメーターを追加します。

```
undercloud_nameservers = $IDM_SERVER_IP_ADDR
overcloud_domain_name = example.com
```

6. アンダークラウドをデプロイします。

```
openstack undercloud install
```

検証

以下の手順を実施して、アンダークラウドが正しく登録されたことを確認します。

1. IdM のホストを一覧表示します。

```
$ kinit admin
$ ipa host-find
```

2. アンダークラウドに **/etc/novajoin/krb5.keytab** が存在することを確認します。

```
ls /etc/novajoin/krb5.keytab
```

注記

novajoin というディレクトリー名は、従来の方式に対応させる目的でのみ使用されています。

2.3.2. オーバークラウドでの TLS-e の設定

TLS everywhere (TLS-e) を設定したオーバークラウドをデプロイする場合、アンダークラウドおよびオーバークラウドの IP アドレスは自動的に IdM に登録されます。

注記

IP アドレスの自動登録を無効にするには、**IDMModifyDNS** heat パラメーターを false に設定します。

```
parameter_defaults:
  ....
  IdMModifyDNS: false
```

1. オーバークラウドをデプロイする前に、以下のような内容で YAML ファイル **tls-parameters.yaml** を作成します。お使いの環境に固有の値を選択してください。

   - **DnsServers** パラメーターの値は、IdM サーバーの IP アドレスを反映させる必要があります。
   - IdM サーバーのドメインがクラウドのドメインと異なる場合は、**DnsSearchDomains** パラメーターに追加します。たとえば、**DnsSearchDomains: ["example.com", "bigcorp.com"]** のように設定します。
   - 事前にプロビジョニングされたノードがある場合には、**IDMInstallClientPackages** パラメーターの値を **true** に設定して、オーバークラウドノードに必要なパッケージをインストールします。
   - **OS::TripleO::Services::IpaClient** パラメータに示す値は、**enable-internal-tls.yaml** ファイルのデフォルト設定を上書きします。**openstack overcloud deploy** コマンドで、**enable-internal-tls.yaml** の後に **tls-parameters.yaml** ファイルを指定するようにします。
   - アクティブ/アクティブとして設定された cinder と共に分散コンピュートノード (DCN) アーキテクチャーを実行している場合は、**EnableEtcdInternalTLS** パラメーターを **true** に追加および設定する必要があります。

```
parameter_defaults:
    DnsSearchDomains: ["example.com"]
    DnsServers: ["192.168.1.13"]
    CloudDomain: example.com
    CloudName: overcloud.example.com
    CloudNameInternal: overcloud.internalapi.example.com
    CloudNameStorage: overcloud.storage.example.com
    CloudNameStorageManagement: overcloud.storagemgmt.example.com
    CloudNameCtlplane: overcloud.ctlplane.example.com
    IdMServer: freeipa-0.redhat.local
    IdMDomain: redhat.local
    IdMInstallClientPackages: False

resource_registry:
      OS::TripleO::Services::IpaClient: /usr/share/openstack-tripleo-heat-
templates/deployment/ipa/ipaservices-baremetal-ansible.yaml
```

2. オーバークラウドをデプロイします。デプロイメントコマンドに tls-parameters.yaml を追加する必要があります。

```
DEFAULT_TEMPLATES=/usr/share/openstack-tripleo-heat-templates/
CUSTOM_TEMPLATES=/home/stack/templates

openstack overcloud deploy \
-e ${DEFAULT_TEMPLATES}/environments/ssl/tls-everywhere-endpoints-dns.yaml \
-e ${DEFAULT_TEMPLATES}/environments/services/haproxy-public-tls-certmonger.yaml \
-e ${DEFAULT_TEMPLATES}/environments/ssl/enable-internal-tls.yaml \
-e ${CUSTOM_TEMPLATES}/tls-parameters.yaml \
...
```

3. keystone にエンドポイント一覧のクエリーを行い、各エンドポイントが HTTPS を使用していることを確認します。

```
openstack overcloud endpoint list
```

2.4. TLS EVERYWHERE (TLS-E) による MEMCACHED トラフィックの暗号化

テクノロジープレビューとして、TLS-e で memcached トラフィックを暗号化できるようになりました。この機能は、novajoin と tripleo-ipa の両方で機能します。

1. 以下の内容で **memcached.yaml** という名前の環境ファイルを作成し、memcached の TLS サポートを追加します。**memcached_node_ips** パラメーターセクションの値を、実際の環境に固有の InternalAPI ホスト名または IP アドレスに置き換えます。

```
parameter_defaults:
  MemcachedTLS: true
  MemcachedPort: 11212
  ExtraConfig:
    memcached_port: 11212
    memcached_authtoken_port: 11211
    memcached_node_ips: "%{alias('memcached_node_names')}"
    nova::cache::backend: "dogpile.cache.pymemcache"
    heat::cache::backend: "dogpile.cache.pymemcache"
    ceilometer::cache_backend: "dogpile.cache.pymemcache"
```

2. オーバークラウドのデプロイプロセスに **memcached.yaml** 環境ファイルを追加します。

```
openstack overcloud deploy --templates \
-e /usr/share/openstack-tripleo-heat-templates/environments/ssl/enable-internal-tls.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/ssl/tls-everywhere-endpoints-dns.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/services/haproxy-public-tls-certmonger.yaml \
-e /home/stack/memcached.yaml
...
```

関連情報

- novajoin を使用した TLSe のデプロイに関する詳細は、「novajoin を使用した Red Hat Identity Manager (IdM) へのノードの登録」を参照してください。
- tripleo-ipa を使用した TLSe のデプロイに関する詳細は、「Ansible を使用した TLS-e の実装」を参照してください。

この機能は、本リリースでは **テクノロジープレビュー** として提供しているため、Red Hat では全面的にはサポートしていません。これは、テスト目的のみでご利用いただく機能で、実稼働環境にデプロイすべきではありません。テクノロジープレビュー機能についての詳しい情報は、「対象範囲の詳細」を参照してください。

2.5. RED HAT IDENTITY MANAGER (IDM) サーバーの認証情報の設定

Red Hat Identity Manager (IdM) が OpenStack Identity と統合するように設定するには、Identity サービスが使用する LDAP アカウントを設定して、Red Hat OpenStack ユーザーのユーザーグループを作成し、ルックアップアカウントのパスワードを設定します。

前提条件

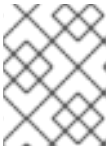
- Red Hat Identity Manager (IdM) が設定済みで、稼働していること。
- Red Hat OpenStack Platform (RHOSP) が設定済みで、稼働していること。
- DNS 名前解決が完全に機能しており、かつ全ホストが適切に登録されていること。
- IdM 認証トラフィックが LDAPS で暗号化され、ポート 636 を使用していること。
- 推奨: 単一の障害点を避けるために、高可用性または負荷分散ソリューションを備えた IdM を実装していること。

手順

IdM サーバーで以下の手順を実行します。

1. OpenStack Identity サービスで使用する LDAP ルックアップアカウントを作成して、IdM LDAP サービスにクエリーを実行します。

```
# kinit admin
# ipa user-add
First name: OpenStack
Last name: LDAP
User  [radministrator]: svc-ldap
```



注記

作成が完了したら、このアカウントのパスワード期限の設定を確認してください。

2. **grp-openstack** という名前の RHOSP ユーザーグループを作成します。OpenStack Identity でパーミッションを割り当てることができるのは、このグループのメンバーのみです。

```
# ipa group-add --desc="OpenStack Users" grp-openstack
```

3. **svc-ldap** アカウントのパスワードを設定して、**grp-openstack** グループに追加します。

```
# ipa passwd svc-ldap
# ipa group-add-member --users=svc-ldap grp-openstack
```

4. **svc-ldap** ユーザーとしてログインし、プロンプトが表示されたらパスワードを変更します。

```
# kinit svc-ldap
```

2.6. RED HAT IDENTITY MANAGER (IDM) LDAPS 証明書のインストール

OpenStack Identity (keystone) は、LDAPS クエリーを使用してユーザーアカウントを検証します。このトラフィックを暗号化するために、keystone は **keystone.conf** で定義されている証明書ファイルを使用します。LDAPS 証明書をインストールするには、Red Hat Identity Manager (IdM) サーバーから keystone が参照できる場所に証明書をコピーし、それを **.crt** から **.pem** 形式に変換します。

注記

LDAP 認証に複数のドメインを使用する場合、**Unable to retrieve authorized projects** または **Peer's Certificate issuer is not recognized** など、さまざまなエラーが発生する可能性があります。これは、keystone が特定ドメインに誤った証明書を使用すると発生する可能性があります。回避策として、すべての LDAPS 公開鍵を単一の **.crt** バンドルにマージし、このファイルを使用するようにすべての keystone ドメインを設定します。

前提条件

- IdM サーバーの認証情報が設定されている。

手順

1. IdM の環境で、LDAPS 証明書を見つけます。このファイルの場所は、**/etc/openldap/ldap.conf** で確認することができます。

```
TLS_CACERT /etc/ipa/ca.crt
```

2. keystone サービスを実行しているコントローラーノードにファイルをコピーします。たとえば、**scp** コマンドは **ca.crt** ファイルを **node.lab.local** にコピーします。

```
# scp /etc/ipa/ca.crt root@node.lab.local:/root/
```

3. **ca.crt** ファイルを証明書のディレクトリーにコピーします。keystone サービスは、この場所を使用して証明書にアクセスします。

```
# cp ca.crt /etc/pki/ca-trust/source/anchors
```

4. (オプション) **ldapsearch** などの診断のコマンドを実行する必要がある場合には、RHEL の証明書ストアに証明書を追加する必要もあります。

   a. 3.コントローラーノードで **.crt** を **.pem** 形式に変換します。

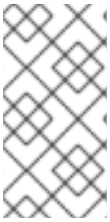
```
# openssl x509 -in ca.crt -out ca.pem -outform PEM
```

   b. コントローラーノードに **.pem** をインストールします。たとえば、Red Hat Enterprise Linux の場合は以下を実行します。

```
# cp ca.pem /etc/pki/ca-trust/source/anchors/
# update-ca-trust
```

2.7. ドメイン固有の LDAP バックエンドを使用する DIRECTOR の設定

director が 1 つ以上の LDAP バックエンドを使用するように設定するには、heat テンプレートで **KeystoneLDAPDomainEnable** フラグを **true** に設定し、各 LDAP バックエンドに関する情報が含まれる環境ファイルを設定します。次に、director は keystone ドメインごとに別の LDAP バックエンドを使用します。



注記

ドメイン設定ファイルのデフォルトのディレクトリーは **/etc/keystone/domains/** に設定されています。**keystone::domain_config_directory** hiera キーを使用して環境ファイル内に **ExtraConfig** パラメーターを追加して必要なパスを設定することによってオーバーライドすることができます。

Procedure

1. デプロイメントの heat テンプレートで、**KeystoneLDAPDomainEnable** フラグを **true** に設定します。これにより、**identity** 設定グループ内の keystone に **domain_specific_drivers_enabled** オプションが設定されます。

2. **tripleo-heat-templates** に **KeystoneLDAPBackendConfigs** パラメーターを設定して、LDAP バックエンド設定の仕様を追加します。その後、必要な LDAP オプションを指定できます。

3. **keystone_domain_specific_ldap_backend.yaml** 環境ファイルのコピーを作成します。

```
$ cp /usr/share/openstack-tripleo-heat-
templates/environments/services/keystone_domain_specific_ldap_backend.yaml
/home/stack/templates/
```

4. **/home/stack/templates/keystone_domain_specific_ldap_backend.yaml** 環境ファイルを編集して、デプロイメントに適した値を設定します。たとえば、以下のパラメーターは、**testdomain** という名前の keystone ドメイン向けの LDAP 設定を作成します。

```
parameter_defaults:
  KeystoneLDAPDomainEnable: true
  KeystoneLDAPBackendConfigs:
    testdomain:
      url: ldaps://192.0.2.250
      user: cn=openstack,ou=Users,dc=director,dc=example,dc=com
      password: RedactedComplexPassword
      suffix: dc=director,dc=example,dc=com
      user_tree_dn: ou=Users,dc=director,dc=example,dc=com
      user_filter: "(memberOf=cn=OSuser,ou=Groups,dc=director,dc=example,dc=com)"
      user_objectclass: person
      user_id_attribute: cn
```

5. (オプション) 環境ファイルにドメインをさらに追加します。以下に例を示します。

```
KeystoneLDAPBackendConfigs:
  domain1:
    url: ldaps://domain1.example.com
    user: cn=openstack,ou=Users,dc=director,dc=example,dc=com
    password: RedactedComplexPassword
    ...
  domain2:
    url: ldaps://domain2.example.com
    user: cn=openstack,ou=Users,dc=director,dc=example,dc=com
    password: RedactedComplexPassword
    ...
```

これにより、**domain1** と **domain2** という名前の 2 つのドメインが指定され、各ドメインには、異なる LDAP ドメインが独自の設定で適用されます。

2.8. コントローラーノードでの OPENSTACK IDENTITY ドメインの設定

外部のユーザー管理サービスと統合する OpenStack Identity (keystone) を実行するコントローラーノードを設定するには、まず LDAP 認証を使用するように SELinux を設定し、コントローラーノードに **domains** ディレクトリーを作成します。次に、OpenStack Identity が複数のバックエンドを使用するように設定します。また、Dashboard が複数のドメインを使用するように設定します。

注記

director を使用している場合には、以下の手順で参照されている設定ファイルが Puppet によって管理されている点に注意してください。このため、**openstack overcloud deploy** コマンドを実行するたびに、自分で追加したカスタム設定が上書きされる可能性があります。

プランニング

設定ファイルを更新する場合には、特定の OpenStack サービスはコンテナー内で実行されるようになったことを認識する必要があります。これは、keystone、nova、cinder などのサービスが対象です。そのため、考慮すべき特定の管理プラクティスがいくつかあります。

- 物理ノードのホストオペレーティングシステム上の設定ファイル (例: **/etc/cinder/cinder.conf**) は更新しないでください。コンテナー化されたサービスはこのようなファイルを参照しません。
- コンテナー内で実行されている設定ファイルは更新しないでください。コンテナーを再起動すると変更が失われてしまいます。
  代わりに、コンテナー化されたサービスに変更を加える必要がある場合は、コンテナーの生成に使用される設定ファイルを更新する必要があります。これらのファイルは **/var/lib/config-data/puppet-generated/** 内に保管されています。

  以下に例を示します。

  - keystone: **/var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf**
  - cinder: **/var/lib/config-data/puppet-generated/cinder/etc/cinder/cinder.conf**
  - nova: **/var/lib/config-data/puppet-generated/nova/etc/nova/nova.conf**

  変更内容は、サービスを再起動した後に適用されます。例: **sudo systemctl restart tripleo_keystone**

Procedure

OpenStack Identity (keystone) サービスを実行するそれぞれのコントローラーノードで、この手順を実施します。

1. SELinux を設定します。

```
# setsebool -P authlogin_nsswitch_use_ldap=on
```

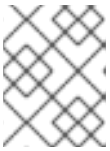
   出力には、以下のようなメッセージが含まれている場合がありますが、これは無視できます。

```
Full path required for exclude: net:[4026532245].
```

2. **domains** ディレクトリーを作成します。

```
# mkdir /var/lib/config-data/puppet-generated/keystone/etc/keystone/domains/
# chown 42425:42425 /var/lib/config-data/puppet-generated/keystone/etc/keystone/domains/
```

3. keystone が複数のバックエンドを使用するように設定します。



注記

dnf install crudini を使用して **crudini** をインストールする必要がある場合があります。

```
# crudini --set /var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf
identity domain_specific_drivers_enabled true
# crudini --set /var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf
identity domain_config_dir /etc/keystone/domains
# crudini --set /var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf
assignment driver sql
```

4. Dashboard で複数のドメインを有効にします。以下の行を **/var/lib/config-data/puppet-generated/horizon/etc/openstack-dashboard/local_settings** に追加します。

```
OPENSTACK_API_VERSIONS = {
    "identity": 3
}
OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = 'Default'
```

注記

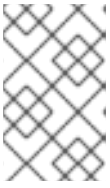
director を使用している場合には、**/var/lib/config-data/puppet-generated/horizon/etc/openstack-dashboard/local_settings** が Puppet によって管理されている点に注意してください。このため、カスタムの設定を追加しても、**openstack overcloud deploy** プロセスを実行するたびに上書きされる可能性があります。上書きされた場合には、この設定を毎回手動で追加し直す必要がある場合があります。

horizon コンテナーを再起動して、設定を適用します。

```
$ sudo systemctl restart tripleo_horizon
```

5. 前のステップでドメイン名として取得した NetBIOS 名の値で、外部サービスとの統合用の keystone ドメインを作成します。このアプローチでは、ログインプロセス中に、一貫したドメイン名をユーザーに提示することができます。以下の例では、**LAB** は、Identity サービスドメインとして使用する NetBIOS の名前に置き換えます。

```
$ openstack domain create LAB
```



注記

このコマンドが使用できない場合には、**# source overcloudrc-v3** のコマンドを実行して、コマンドラインセッションでの keystone v3 へのアクセスが有効化されているかどうかを確認します。

6. 統合する外部サービスの設定ファイルを作成します。

- Active Directory Domain Service (AD DS):**/var/lib/config-data/puppet-generated/keystone/etc/keystone/domains/keystone.LAB.conf** (**LAB** は、前のステップで取得した NetBIOS 名に置き換えます) という名前の新規ファイルに LDAP 設定を入力します。以下の設定例は、実際に使用する LDAP デプロイメントに合わせて編集する必要があります。

```
[ldap]
url                = ldaps://addc.lab.local:636
user               = CN=svc-ldap,OU=labUsers,DC=lab,DC=local
password           = RedactedComplexPassword
suffix             = DC=lab,DC=local
user_tree_dn       = OU=labUsers,DC=lab,DC=local
user_objectclass   = person
user_filter        = (|(memberOf=cn=grp-openstack,OU=labUsers,DC=lab,DC=local)
(memberOf=cn=grp-openstack-admin,OU=labUsers,DC=lab,DC=local)
(memberOf=memberOf=cn=grp-openstack-demo,OU=labUsers,DC=lab,DC=local))
user_id_attribute  = sAMAccountName
user_name_attribute = sAMAccountName
user_mail_attribute = mail
user_pass_attribute =
user_enabled_attribute = userAccountControl
user_enabled_mask  = 2
user_enabled_default = 512
user_attribute_ignore = password,tenant_id,tenants
group_objectclass  = group
group_tree_dn      = OU=labUsers,DC=lab,DC=local
group_filter       = (CN=grp-openstack*)
group_id_attribute = cn
group_name_attribute = name
use_tls            = False
tls_cacertfile     =/etc/pki/ca-trust/source/anchors/anchorsaddc.lab.local.pem

query_scope        = sub
chase_referrals    = false

[identity]
driver = ldap
```

各設定項目について説明します。

設定	説明
url	認証に使用する AD Domain Controller。LDAPS ポート **636** を使用します。
user	LDAP クエリーに使用する AD アカウントの**識別名**。たとえば、**Get-ADuser svc-ldap | select DistinguishedName** を使用する AD 内の **svc-ldap** アカウントの**識別名** の値を特定することができます。

設定	説明
password	上記で使用した AD アカウントのパスワード (プレーンテキスト形式)。
suffix	AD ドメインの **識別名**。この値は、**Get-ADDomain | select DistinguishedName** を使用して特定することができます。
user_tree_dn	OpenStack アカウントを含む **組織単位** (OU)。
user_objectclass	LDAP ユーザーの種別を定義します。AD には **person** の種別を使用します。
user_filter	Identity サービスに対して提示するユーザーをフィルタリングします。その結果、**grp-openstack** グループのメンバーのみに Identity サービスで定義されているパーミッションを付与することができます。この値には、グループの **完全な識別名** が必要です:**Get-ADGroup grp-openstack | select DistinguishedName**
user_id_attribute	ユーザー ID に使用する AD 値をマッピングします。
user_name_attribute	**names** に使用する AD 値をマッピングします。
user_mail_attribute	ユーザーのメールアドレスに使用する AD 値をマッピングします。
user_pass_attribute	この値は空白のままにします。
user_enabled_attribute	アカウントが有効にされているかどうかを検証する AD の設定。
user_enabled_mask	アカウントが有効化されているかを判断するために確認すべき値を定義します。ブール値が返されない場合に使用します。
user_enabled_default	アカウントが有効化されていることを示す AD 値。
user_attribute_ignore	Identity サービスが無視する必要のあるユーザー属性を定義します。
group_objectclass	**groups** に使用する AD 値をマッピングします。

設定	説明
group_tree_dn	そのユーザーグループを含む **組織単位** (OU)。
group_filter	Identity サービスに提示するグループをフィルタリングします。
group_id_attribute	グループ ID に使用する AD 値をマッピングします。
group_name_attribute	グループ名に使用する AD 値をマッピングします。
use_tls	TLS を使用するかどうかを定義します。STARTTLS ではなく LDAPS で暗号化する場合には、無効にする必要があります。
tls_cacertfile	**.crt** 証明書ファイルへのパスを指定します。
query_scope	**grp-openstack** グループに所属するユーザーを特定する場合に、Identity サービスがネスト化された子 OU 内での検索ができるように設定します。
chase_referrals	**false** に設定します。この設定により **python-ldap** が匿名のアクセスによる全参照を追跡しないようにします。

- Red Hat Identity Manager (IdM): **/var/lib/config-data/puppet-generated/keystone/etc/keystone/domains/keystone.LAB.conf** (**LAB** は、前のステップで作成したドメイン名に置き換えます) という名前の新規ファイルに LDAP 設定を入力します。以下の設定例は、実際に使用する IdM デプロイメントに合わせて編集する必要があります。

```
[ldap]
url =  ldaps://idm.lab.local
user = uid=svc-ldap,cn=users,cn=accounts,dc=lab,dc=local
user_filter = (memberOf=cn=grp-openstack,cn=groups,cn=accounts,dc=lab,dc=local)
password = RedactedComplexPassword
user_tree_dn = cn=users,cn=accounts,dc=lab,dc=local
user_objectclass = inetUser
user_id_attribute = uid
user_name_attribute = uid
user_mail_attribute = mail
user_pass_attribute =
group_tree_dn              = cn=groups,cn=accounts,dc=lab,dc=local
group_objectclass            = groupOfNames
group_id_attribute          = cn
group_name_attribute       =  cn
group_member_attribute  = member
group_desc_attribute        = description
```

```
use_tls                 = False
query_scope             = sub
chase_referrals         = false
tls_cacertfile =/etc/pki/ca-trust/source/anchors/anchorsca.crt

[identity]
driver = ldap
```

各設定項目について説明します。

設定	説明
url	認証に使用する IdM サーバー。LDAPS ポート **636** を使用します。
user	LDAP クエリーに使用する IdM 内のアカウント。
password	上記で使用した IdM アカウントのパスワード (プレーンテキスト形式)。
user_filter	Identity サービスに対して提示するユーザーをフィルタリングします。その結果、**grp-openstack** グループのメンバーのみに Identity サービスで定義されているパーミッションを付与することができます。
user_tree_dn	IdM 内の OpenStack アカウントへのパス。
user_objectclass	LDAP ユーザーの種別を定義します。IdM には **inetUser** の種別を使用します。
user_id_attribute	ユーザー ID に使用する IdM 値をマッピングします。
user_name_attribute	**names** に使用する IdM 値をマッピングします。
user_mail_attribute	ユーザーのメールアドレスに使用する IdM 値をマッピングします。
user_pass_attribute	この値は空白のままにします。

注記

IdM グループとの統合で返されるのは直接のメンバーだけで、ネスト化されたグループは返されません。したがって、**LDAP_MATCHING_RULE_IN_CHAIN** または **memberof:1.2.840.113556.1.4.1941:** に依存するクエリーは、現在 IdM では機能しません。

7. 設定ファイルの所有権を keystone ユーザーに変更します。

```
# chown 42425:42425 /var/lib/config-data/puppet-
generated/keystone/etc/keystone/domains/keystone.LAB.conf
```

8. keystone サービスを再起動して、変更を適用します。

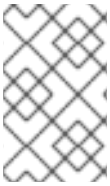
```
# sudo systemctl restart tripleo_keystone
```

関連資料

- 「OpenStack Identity ドメインへの管理ユーザーアクセス権の付与」
- 「ドメイン固有の LDAP バックエンドを使用する director の設定」

2.9. OPENSTACK IDENTITY ドメインへの管理ユーザーアクセス権の付与

admin ユーザーが OpenStack Identity (keystone) ドメインにアクセスして **Domain** タブを表示するのを許可するには、ドメインと **admin** ユーザーの ID を取得した後、ドメインのユーザーに **admin** ロールを割り当てます。



注記

これにより、OpenStack admin アカウントには外部サービスドメインのパーミッションは付与されません。この場合には、**ドメイン** という用語は、OpenStack が使用する keystone ドメインのことを指しています。

Procedure

この手順では、**LAB** ドメインを使用します。ドメイン名は、設定するドメインの実際の名前に置き換えます。

1. **LAB** ドメインの ID を取得します。

```
$ openstack domain show LAB
+---------+----------------------------------+
| Field   | Value                            |
+---------+----------------------------------+
| enabled | True                             |
| id      | 6800b0496429431ab1c4efbb3fe810d4 |
| name    | LAB                              |
+---------+----------------------------------+
```

2. **default** ドメインから **admin** ユーザーの ID を取得します。

```
$ openstack user list --domain default | grep admin
| 3d75388d351846c6a880e53b2508172a | admin      |
```

3. **admin** ロールの ID を取得します。

```
$ openstack role list
```

出力は、統合する外部サービスによって異なります。

- Active Directory Domain Service (AD DS):

```
+----------------------------------+-----------------+
| ID                               | Name            |
+----------------------------------+-----------------+
| 01d92614cd224a589bdf3b171afc5488 | admin           |
| 034e4620ed3d45969dfe8992af001514 | member          |
| 0aa377a807df4149b0a8c69b9560b106 | ResellerAdmin   |
| 9369f2bf754443f199c6d6b96479b1fa | heat_stack_user |
| cfea5760d9c948e7b362abc1d06e557f | reader          |
| d5cb454559e44b47aaa8821df4e11af1 | swiftoperator   |
| ef3d3f510a474d6c860b4098ad658a29 | service         |
+----------------------------------+-----------------+
```

- Red Hat Identity Manager (IdM):

```
+----------------------------------+---------------+
| ID                               | Name          |
+----------------------------------+---------------+
| 544d48aaffde48f1b3c31a52c35f01f9 | SwiftOperator |
| 6d005d783bf0436e882c55c62457d33d | ResellerAdmin |
| 785c70b150ee4c778fe4de088070b4cf | admin         |
| 9fe2ff9ee4384b1894a90878d3e92bab | _member_      |
+----------------------------------+---------------+
```

4. ドメインおよび admin の ID を使用して、keystone **LAB** ドメインの **admin** ロールに **admin** ユーザーを追加するコマンドを構築します。

```
# openstack role add --domain 6800b0496429431ab1c4efbb3fe810d4 --user
3d75388d351846c6a880e53b2508172a 785c70b150ee4c778fe4de088070b4cf
```

2.10. RED HAT OPENSTACK PLATFORM プロジェクトへの外部グループアクセス権の付与

複数の認証されたユーザーが Red Hat OpenStack Platform (RHOSP) リソースにアクセスできるようにするには、外部ユーザー管理サービスから特定のグループを承認し、RHOSP プロジェクトへのアクセス権限を付与します。この場合、OpenStack 管理者は各ユーザーをプロジェクト内のロールに手動で割り当てる必要はありません。その結果、これらのグループのすべてのメンバーは、事前に決定したプロジェクトにアクセスできます。

前提条件

- 外部サービスの管理者が以下の手順を完了していることを確認してください。
  - **grp-openstack-admin** という名前のグループを作成する。
  - **grp-openstack-demo** という名前のグループを作成する。
  - 必要に応じて、RHOSP ユーザーをこれらのグループの 1 つに追加する。
  - ユーザーを **grp-openstack** グループに追加する。
- OpenStack Identity ドメインを作成します。この手順では、**LAB** ドメインを使用します。

- RHOSP プロジェクトを作成するか、選択します。以下の手順では、**openstack project create --domain default --description "Demo Project" demo** コマンドで作成した **demo** という名前のプロジェクトを使用します。

Procedure

1. OpenStack Identity ドメインからユーザーグループの一覧を取得します。

```
# openstack group list --domain LAB
```

コマンド出力は、統合する外部のユーザー管理サービスによって異なります。

- Active Directory Domain Service (AD DS):

```
+------------------------------------------------------------------+--------------------+
| ID                                                               | Name               |
+------------------------------------------------------------------+--------------------+
| 185277be62ae17e498a69f98a59b66934fb1d6b7f745f14f5f68953a665b8851 | grp-
openstack       |
| a8d17f19f464c4548c18b97e4aa331820f9d3be52654aa8094e698a9182cbb88 | grp-
openstack-admin |
| d971bb3bd5e64a454cbd0cc7af4c0773e78d61b5f81321809f8323216938cae8 | grp-
openstack-demo  |
+------------------------------------------------------------------+--------------------+
```

- Red Hat Identity Manager (IdM):

```
+------------------------------------------------------------------+--------------------+
| ID                                                               | Name               |
+------------------------------------------------------------------+--------------------+
| 185277be62ae17e498a69f98a59b66934fb1d6b7f745f14f5f68953a665b8851 | grp-
openstack       |
| a8d17f19f464c4548c18b97e4aa331820f9d3be52654aa8094e698a9182cbb88 | grp-
openstack-admin |
| d971bb3bd5e64a454cbd0cc7af4c0773e78d61b5f81321809f8323216938cae8 | grp-
openstack-demo  |
+------------------------------------------------------------------+--------------------+
```

2. ロールの一覧を取得します。

```
# openstack role list
```

コマンド出力は、統合する外部のユーザー管理サービスによって異なります。

- Active Directory Domain Service (AD DS):

```
+----------------------------------+----------------+
| ID                               | Name           |
+----------------------------------+----------------+
| 01d92614cd224a589bdf3b171afc5488 | admin          |
| 034e4620ed3d45969dfe8992af001514 | member         |
| 0aa377a807df4149b0a8c69b9560b106 | ResellerAdmin  |
| 9369f2bf754443f199c6d6b96479b1fa | heat_stack_user |
| cfea5760d9c948e7b362abc1d06e557f | reader         |
```

```
| d5cb454559e44b47aaa8821df4e11af1 | swiftoperator   |
| ef3d3f510a474d6c860b4098ad658a29 | service         |
+----------------------------------+-----------------+
```

- Red Hat Identity Manager (IdM):

```
+----------------------------------+---------------+
| ID                               | Name          |
+----------------------------------+---------------+
| 0969957bce5e4f678ca6cef00e1abf8a | ResellerAdmin |
| 1fcb3c9b50aa46ee8196aaaecc2b76b7 | admin         |
| 9fe2ff9ee4384b1894a90878d3e92bab | _member_      |
| d3570730eb4b4780a7fed97eba197e1b | SwiftOperator |
+----------------------------------+---------------+
```

3. ユーザーグループを上記のロールの 1 つまたは複数に追加して、プロジェクトへのアクセス権を付与します。たとえば、**grp-openstack-demo** グループのユーザーを **demo** プロジェクトの一般ユーザーに指定するには、統合する外部サービスに応じて、グループを **member** または **_member_** ロールに追加する必要があります。

   - Active Directory Domain Service (AD DS):

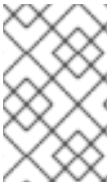
```
# openstack role add --project demo --group
d971bb3bd5e64a454cbd0cc7af4c0773e78d61b5f81321809f8323216938cae8  member
```

   - Red Hat Identity Manager (IdM):

```
$ openstack role add --project demo --group
d971bb3bd5e64a454cbd0cc7af4c0773e78d61b5f81321809f8323216938cae8
_member_
```

結果

grp-openstack-demo のメンバーは、ユーザー名とパスワードを入力し、**Domain** フィールドに **LAB** を入力して Dashboard にログインすることができます。



注記

ユーザーに **Error: Unable to retrieve container list.** というエラーメッセージが表示され、コンテナーの管理が可能であることが想定されている場合には、**SwiftOperator** ロールに追加する必要があります。

関連資料

- 「Red Hat OpenStack Platform プロジェクトへの外部ユーザーアクセス権の付与」

2.11. RED HAT OPENSTACK PLATFORM プロジェクトへの外部ユーザーアクセス権の付与

grp-openstack グループからの特定認証ユーザーに OpenStack リソースへのアクセス権限を付与するには、これらのユーザーに Red Hat OpenStack Platform (RHOSP) プロジェクトへの直接アクセス権限を付与できます。グループにアクセス権を付与する代わりに、個々のユーザーにアクセス権を付与する場合は、このプロセスを使用します。

前提条件

- 外部サービスの管理者が以下の手順を完了していることを確認してください。
  - RHOSP ユーザーを **grp-openstack** グループに追加する。
  - OpenStack Identity ドメインを作成する。この手順では、**LAB** ドメインを使用します。
- RHOSP プロジェクトを作成するか、選択します。以下の手順では、**openstack project create --domain default --description "Demo Project" demo** コマンドで作成した **demo** という名前のプロジェクトを使用します。

Procedure

1. OpenStack Identity ドメインからユーザーの一覧を取得します。

```
# openstack user list --domain LAB
 +----------------------------------------------------------------+----------------+
| ID                                                             | Name           |
+----------------------------------------------------------------+----------------+
| 1f24ec1f11aeb90520079c29f70afa060d22e2ce92b2eba7784c841ac418091e | user1          |
| 12c062faddc5f8b065434d9ff6fce03eb9259537c93b411224588686e9a38bf1 | user2          |
| afaf48031eb54c3e44e4cb0353f5b612084033ff70f63c22873d181fdae2e73c | user3          |
| e47fc21dcf0d9716d2663766023e2d8dc15a6d9b01453854a898cabb2396826e | user4
|
+----------------------------------------------------------------+----------------+
```

2. ロールの一覧を取得します。

```
# openstack role list
```

コマンド出力は、統合する外部のユーザー管理サービスによって異なります。

- Active Directory Domain Service (AD DS):

```
+----------------------------------+----------------+
```

```
| ID                               | Name           |
+----------------------------------+----------------+
| 01d92614cd224a589bdf3b171afc5488 | admin          |
| 034e4620ed3d45969dfe8992af001514 | member         |
| 0aa377a807df4149b0a8c69b9560b106 | ResellerAdmin  |
| 9369f2bf754443f199c6d6b96479b1fa | heat_stack_user |
| cfea5760d9c948e7b362abc1d06e557f | reader         |
| d5cb454559e44b47aaa8821df4e11af1 | swiftoperator  |
| ef3d3f510a474d6c860b4098ad658a29 | service        |
+----------------------------------+----------------+
```

- Red Hat Identity Manager (IdM):

```
+----------------------------------+---------------+
| ID                               | Name          |
+----------------------------------+---------------+
| 0969957bce5e4f678ca6cef00e1abf8a | ResellerAdmin |
| 1fcb3c9b50aa46ee8196aaaecc2b76b7 | admin         |
| 9fe2ff9ee4384b1894a90878d3e92bab | _member_      |
| d3570730eb4b4780a7fed97eba197e1b | SwiftOperator |
+----------------------------------+---------------+
```

3. 一覧表示されたロールの 1 つまたは複数にユーザーを追加して、RHOSP プロジェクトへのアクセス権を付与します。たとえば、**user1** を **demo** プロジェクトの一般ユーザーに指定するには、統合する外部サービスに応じて、ユーザーを **member** または **_member_** ロールに追加します。

   - Active Directory Domain Service (AD DS):

     ```
     # openstack role add --project demo --user
     1f24ec1f11aeb90520079c29f70afa060d22e2ce92b2eba7784c841ac418091e member
     ```

   - Red Hat Identity Manager (IdM):

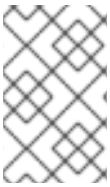
     ```
     # openstack role add --project demo --user
     1f24ec1f11aeb90520079c29f70afa060d22e2ce92b2eba7784c841ac418091e _member_
     ```

4. **user1** を **demo** プロジェクトの管理ユーザーにするには、**admin** ロールに追加します。

   ```
   # openstack role add --project demo --user
   1f24ec1f11aeb90520079c29f70afa060d22e2ce92b2eba7784c841ac418091e admin
   ```

結果

user1 ユーザーは、外部ユーザー名とパスワードを入力し、**Domain** フィールドに **LAB** を入力して Dashboard にログインすることができます。



注記

ユーザーに **Error: Unable to retrieve container list.** というエラーメッセージが表示され、コンテナーの管理が可能であることが想定されている場合には、**SwiftOperator** ロールに追加する必要があります。

関連資料

- 「Red Hat OpenStack Platform プロジェクトへの外部グループアクセス権の付与」

2.12. OPENSTACK IDENTITY ドメインおよびユーザーの一覧表示

利用可能なエントリーを表示するには、**openstack domain list** コマンドを使用します。Identity サービスに複数のドメインを設定すると、Dashboard のログインページに新しい **ドメイン** フィールドが有効になります。ユーザーは、ログイン認証情報にマッチするドメインを入力する必要があります。

重要

統合が完了したら、**Default** ドメインまたは新たに作成する keystone ドメインに新規プロジェクトを作成するかどうかを決定する必要があります。ワークフローとユーザーアカウントの管理方法を検討する必要があります。可能な場合には、**Default** ドメインを内部ドメインとして使用し、サービスアカウントと **admin** プロジェクトを管理し、外部ユーザーを別のドメインに維持します。

この例では、外部アカウントは **LAB** ドメインを指定する必要があります。**admin** のような組み込みの keystone アカウントには、ドメインに **Default** を指定する必要があります。

Procedure

1. ドメインの一覧を表示します。

```
# openstack domain list
+----------------------------------+---------+---------+-----------------------------------------------------------
--------------+
| ID                               | Name    | Enabled | Description                                                |
```

```
+----------------------------------+---------+---------+-----------------------------------------------------
--------------+
| 6800b0496429431ab1c4efbb3fe810d4 | LAB     | True    |
|
| default                          | Default | True    | Owns users and projects available on Identity API
v2. |
+----------------------------------+---------+---------+-----------------------------------------------------
--------------+
```

2. 特定のドメインのユーザー一覧を表示します。このコマンド例では、**--domain LAB** を指定し、**grp-openstack** グループのメンバーである LAB ドメイン内のユーザーを返します。

```
# openstack user list --domain LAB
```

--domain Default を追加して、組み込みの keystone アカウントを表示することもできます。

```
# openstack user list --domain Default
```

2.13. 非管理者ユーザーの認証情報ファイルの作成

OpenStack Identity のユーザーおよびドメインを設定したら、管理者以外のユーザーの認証情報ファイルを作成する必要がある場合があります。

手順

- 非管理者ユーザー用の認証情報 (RC) ファイルを作成します。この例では、ファイルで **user1** ユーザーを使用しています。

```
$ cat overcloudrc-v3-user1
# Clear any old environment that may conflict.
for key in $( set | awk '{FS="="}  /^OS_/ {print $1}' ); do unset $key ; done
export OS_USERNAME=user1
export NOVA_VERSION=1.1
export OS_PROJECT_NAME=demo
export OS_PASSWORD=RedactedComplexPassword
export OS_NO_CACHE=True
export COMPUTE_API_VERSION=1.1
export no_proxy=,10.0.0.5,192.168.2.11
export OS_CLOUDNAME=overcloud
export OS_AUTH_URL=https://10.0.0.5:5000/v3
export OS_AUTH_TYPE=password
export PYTHONWARNINGS="ignore:Certificate has no, ignore:A true
SSLContext object is not available"
export OS_IDENTITY_API_VERSION=3
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=LAB
```

2.14. OPENSTACK IDENTITY と外部のユーザー管理サービスの統合のテスト

OpenStack Identity (keystone) と Active Directory Domain Service (AD DS) が正常に統合されていることをテストするには、Dashboard 機能へのユーザーアクセスをテストします。

前提条件

- Active Directory (AD) または Red Hat Identity Manager (IdM) などの外部のユーザー管理サービスとの統合

Procedure

1. 外部のユーザー管理サービスにテストユーザーを作成し、そのユーザーを **grp-openstack** グループに追加します。
2. Red Hat OpenStack Platform で、**demo** プロジェクトの **_member_** ロールにユーザーを追加します。
3. AD テストユーザーの認証情報を使用して Dashboard にログインします。
4. 各タブをクリックし、エラーメッセージなしに正常に表示されているかどうかを確認します。
5. Dashboard を使用してテストインスタンスをビルドします。

注記

これらの手順で問題が発生した場合は、**admin** アカウントを使用して Dashboard にログインし、そのユーザーとして後続の手順を実施します。テストが成功した場合、OpenStack が予想通りに機能していること、および OpenStack Identity と Active Directory との統合設定のどこかに問題が存在することを意味します。

関連資料

- 「Active Directory との統合のトラブルシューティング」

2.15. RED HAT IDENTITY MANAGER (IDM) の統合に関するトラブルシューティング

OpenStack Identity で Red Hat Identity Manager (IdM) との統合を使用する際にエラーが発生する場合には、LDAP コネクションをテストするか、証明書トラスト設定をテストする必要がある場合があります。LDAPS ポートにアクセスできることを確認する必要がある場合もあります。

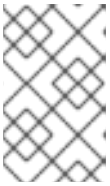
注記

エラーのタイプおよび場所に応じて、以下の手順の関連ステップのみを実行します。

Procedure

1. **ldapsearch** コマンドを使用して IdM サーバーに対してテストクエリーをリモートで実行して、LDAP 接続をテストします。クエリーが成功した場合には、ネットワーク接続が機能しており、IdM サービスが稼働中であることを確認できます。以下の例では、テストクエリーはサーバー **idm.lab.local** のポート **636** に対して実行されます。

```
# ldapsearch -D "cn=directory manager" -H ldaps://idm.lab.local:636 -b "dc=lab,dc=local" -s sub "(objectclass=*)" -w RedactedComplexPassword
```



注記

ldapsearch は、**openldap-clients** パッケージに含まれています。このパッケージは、**# dnf install openldap-clients** のコマンドを実行するとインストールすることができます。

2. **nc** コマンドを使用して、LDAPS ポート **636** がリモートでアクセス可能であることを確認します。この例では、サーバー **idm.lab.local** に対してプローブを実行します。**ctrl-c** を押してプロンプトを終了します。

```
# nc -v idm.lab.local 636
Ncat: Version 6.40 ( http://nmap.org/ncat )
Ncat: Connected to 192.168.200.10:636.
^C
```

接続を確立できなかった場合には、ファイアウォールの設定に問題がある可能性があります。