Red Hat OpenStack Platform 13

与 Identity Service 集成

使用 Active Directory 或红帽身份管理作为外部身份验证后端

Last Updated: 2023-04-20

Red Hat OpenStack Platform 13 与 Identity Service 集成

使用 Active Directory 或红帽身份管理作为外部身份验证后端

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.

法律通告

摘要

使用 Active Directory 或红帽身份管理作为外部身份验证后端。

目录

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。有关更多详情，请参阅我们的首席技术官 Chris Wright 提供的消息。

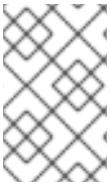
身份服务

身份服务（代号 *keystone*）为 Red Hat OpenStack Platform 13 提供身份验证和授权。

本指南论述了如何将身份服务与 Microsoft Active Directory 域服务(AD DS)、红帽身份管理(IdM)和 LDAP 集成。

第 1 章 ACTIVE DIRECTORY 集成

本章论述了如何将身份服务(keystone)与 Active Directory 域服务集成。在这种情况下，身份服务验证某些 Active Directory 域服务(AD DS)用户，同时在 Identity Service 数据库中保留授权设置和关键服务帐户。因此，Identity Service 对 AD DS 具有对用户帐户进行身份验证的只读访问权限，同时保持对分配给身份验证帐户的特权的管理。



注意

如果使用 director，请参阅 *第 4 章 将特定于域的 LDAP 后端与 director 一起使用*。这是因为以下引用的配置文件由 Puppet 管理。因此，每当运行 **openstack overcloud deploy** 过程时，您添加的任何自定义配置可能会被覆盖。

1.1. 关键术语

- *身份验证* - 使用密码验证用户是否声称的身份的进程。
- *授权* - 验证经过身份验证的用户对试图访问的资源有正确的权限。
- *域* - 此术语与 AD DS 域不同，而是是指在身份服务中配置的额外命名空间，用于对用户、组和项目进行分区。这些独立的域可以配置为对不同 LDAP 或 AD DS 环境中的用户进行身份验证。

1.2. 假设

这个示例部署进行以下假设：

- Active Directory Domain Services 配置且可操作。
- Red Hat OpenStack Platform 是配置且可操作。
- DNS 名称解析可以完全正常工作，所有主机都正确注册。
- AD DS 身份验证流量使用 LDAPS 加密，使用端口 636。

1.3. 影响声明

这些步骤允许 AD DS 用户对 OpenStack 进行验证并访问资源。OpenStack 服务帐户（如 keystone 和 glance）和授权管理（权限、角色、项目）将保留在 Identity Service 数据库中。使用身份服务管理工具将权限和角色分配到 AD DS 帐户。

1.3.1. 高可用性选项

此配置创建依赖于单个 Active Directory Domain Controller 的可用性；如果 Identity Service 无法向 AD Domain Controller 进行身份验证，则项目用户将会受到影响。有很多选项可用于管理这个风险；例如，您可以将身份服务配置为查询 DNS 别名或负载平衡设备，而不是单独的 AD 域控制器。您还可以将 keystone 配置为查询不同的域控制器，它应该不可用。

1.3.2. 中断要求

- 需要重新启动身份服务以添加 AD DS 后端。
- 所有节点上的 Compute 服务都需要重新启动，以便切换到 keystone v3。

- 在 AD DS 中创建其帐户后，用户才能访问仪表板。为减少停机，请考虑预先创建 AD DS 帐户，在此更改前就好。

1.3.3. 防火墙配置

如果防火墙在 AD DS 和 OpenStack 之间过滤流量，则需要允许通过以下端口进行访问：

源	目的地	类型	端口
OpenStack Controller 节点	Active Directory Domain Controller	LDAPS	TCP 636

1.4. 配置 ACTIVE DIRECTORY DOMAIN SERVICES

本节介绍了 Active Directory 管理员需要完成的任务：

表 1.1. 配置步骤

任务	详情
创建一个服务帐户。	这可根据您的服务帐户的命名约定进行命名，例如：**svc-ldap**。这可以是常规域用户帐户。不需要管理员特权。
创建用户组。	如果用户需要访问 OpenStack，则他们必须是此组的成员。这可以通过您的用户组命名规则进行命名，例如：**grp-openstack**。如果此组的成员也是 *项目* 组的成员，则可以在控制面板中授予项目的访问权限。
创建项目组。	每个 OpenStack 项目都需要一个对应的 AD 组。例如，**grp-openstack-demo** 和 **grp-openstack-admin**。
配置服务帐户。	服务帐户 **svc-ldap** 必须是 **grp-openstack** 组的成员。
导出 LDAPS 公钥。	以以下格式导出公钥（而不是私钥）：DER **-encoded x509 .cer** 文件。
将密钥发送到 OpenStack 管理员。	OpenStack 管理员将使用此密钥加密 OpenStack 和 Active Directory 之间的 LDAPS 通信。
检索 AD DS 域的 NetBIOS 名称。	OpenStack 管理员将该名称用于 Keystone 域，允许在环境之间实现一致的域命名。

例如，以下过程显示了在 Active Directory Domain Controller 上运行的 PowerShell 命令：

1. 创建 LDAP 查找帐户。身份服务使用此帐户查询 AD DS LDAP 服务：

```
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. 为此帐户设置密码，然后启用它。系统会提示您输入一个满足 AD 域复杂性要求的密码：

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

3. 为 OpenStack 用户创建一个组，取名为 **grp-openstack**。

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

4. 创建项目组：

```
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. 将 **svc-ldap** 用户添加到 **grp-openstack** 组中：

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

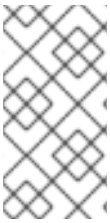
6. 在 AD Domain Controller 中，使用 证书 **MMC** 将 LDAPS 证书的公钥（而不是私钥）导出为 DER 编码的 **x509** .cer 文件。将此文件发送到 OpenStack 管理员。

7. 检索 AD DS 域的 NetBIOS 名称。

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

将此值发送到 OpenStack 管理员。

1.5. 配置 LDAPS 证书



注意

当使用多个域进行 LDAP 身份验证时，您可能会收到各种错误，如 **Unable to retrieve authorized projects**，否则 **无法识别 Peer 的证书签发者**。如果 keystone 对某个域使用不正确的证书，会出现这种情况。作为临时解决方案，将所有 LDAPS 公钥合并到一个 **.crt** 捆绑包中，并配置所有 keystone 域以使用此文件。

Keystone 使用 LDAPS 查询来验证用户帐户。要加密此流量，keystone 使用 **keystone.conf** 定义的证书文件。此流程将从 Active Directory 接收的公钥转换为 **.crt** 格式，并将其复制到 keystone 可以引用的位置。

1. 将 LDAPS 公钥复制到运行 OpenStack Identity (keystone)的节点，并将 **.cer** 转换为 **.crt**。这个示例使用名为 **addc.lab.local.cer** 的源证书文件：

```
# 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
```

注意

另外，如果您需要运行诊断命令，如 **ldapsearch**，您还需要将证书添加到 RHEL 证书存储中：

1. 将 **.cer** 转换为 **.pem**。这个示例使用名为 **addc.lab.local.cer** 的源证书文件：

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

2. 在 OpenStack 控制器上安装 **.pem**。例如，在 Red Hat Enterprise Linux 中：

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

1.6. 配置身份服务

这些步骤准备 Identity Service (keystone)，以便与 AD DS 集成。

注意

如果使用 director，请注意以下引用的配置文件由 Puppet 管理。因此，每当运行 **openstack overcloud deploy** 过程时，您添加的任何自定义配置可能会被覆盖。要将这些设置应用到基于 director 的部署中，请参阅 *第 4 章 将特定于域的 LDAP 后端与 director 一起使用*。

1.6.1. 配置控制器

注意

如果要更新任何配置文件，您需要注意某些 OpenStack 服务现在在容器内运行；这适用于 keystone、nova 和 cinder 等。因此，有一些管理实践需要考虑：

- 不要更新您在物理节点的主机操作系统上找到的任何配置文件，例如：**/etc/cinder/cinder.conf**。这是因为容器化服务不引用此文件。
- 不要更新容器中运行的配置文件。这是因为重启容器后会丢失任何更改。相反，如果您需要对容器化服务添加任何更改，则需要更新用于生成容器的配置文件。它们存储在 **/var/lib/config-data/puppet-generated/** 中

  例如：

- 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**

  重启容器后，所有更改都会被应用。例如：**sudo docker restart keystone**

在运行 keystone 服务的每个 OpenStack 节点上执行这个步骤：

1. 配置 SELinux：

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

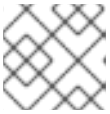
输出可能包含类似于如下的消息。可以忽略它们：

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

2. 创建 **域** 目录：

```
# 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. 将 keystone 配置为使用多个后端：



注意

您可能需要使用 **yum install crudini** 安装 crudini。

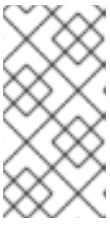
```
# crudini --set /var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf
identity domain_specific_drivers_enabled true
# crudini --set /var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf
identity domain_config_dir /etc/keystone/domains
# crudini --set /var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.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

注意

如果使用 director，请注意 **/var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf** 由 Puppet 管理。因此，每当运行 **openstack overcloud deploy** 过程时，您添加的任何自定义配置可能会被覆盖。因此，您可能需要每次手动重新添加此配置。有关基于 director 的部署，请参阅 *第 4 章 将特定于域的 LDAP 后端与 director 一起使用*。

4. 在控制面板中启用多个域.将这些行添加到 **/var/lib/config-data/puppet-generated/horizon/etc/openstack-dashboard/local_settings** 中：

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



注意

如果使用 director，请注意 **/var/lib/config-data/puppet-generated/horizon/etc/openstack-dashboard/local_settings** 由 Puppet 管理。因此，每当运行 **openstack overcloud deploy** 过程时，您添加的任何自定义配置可能会被覆盖。因此，您可能需要每次手动重新添加此配置。

重启 horizon 容器以应用设置：

```
$ sudo docker restart horizon
```

5. 配置额外后端：
在本例中，**LAB** 是用作 Identity Service 域的 NetBIOS 名称。

a. 为 AD DS 集成创建 Keystone 域。
使用之前检索的 NetBIOS 名称值作为域名。此方法允许您在登录过程中向用户显示一致的域名。例如，如果 NetBIOS 名称是 **LAB** ：

```
$ openstack domain create LAB
```

注意

如果此命令不可用，请运行 **# source overcloudrc-v3**，检查您已为命令行会话启用了 keystone v3。

b. 创建配置文件：
要添加 AD DS 后端，请在名为 **/var/lib/config-data/puppet-generated/keystone/etc/domains/keystone. LAB.conf** 的新文件中输入 LDAP 设置。您需要编辑以下示例设置，以适应 AD DS 部署：

```
[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
```

每个设置的说明：

设置	Description
url	用于身份验证的 AD Domain Controller。使用 LDAPS 端口 **636**。
user	用于 LDAP 查询的 AD 帐户的可辨识 *名称*。例如，您可以使用 **Get-ADuser *svc-ldap* | 选择 *DistinguishedName* 找到 AD 中的 svc-ldap 帐户的可辨识名称**值
password	以上使用的 AD 帐户的纯文本密码。
suffix	AD 域的可辨识 *名称*。您可以使用 **Get-ADDomain | 选择 DistinguishedName**找到这个值
user_tree_dn	*包含 OpenStack* 帐户的组织单元(OU)。
user_objectclass	定义 LDAP 用户的类型。对于 AD，请使用 **person** 类型。
user_filter	过滤呈现给身份服务的用户。因此，只有 grp-openstack 组的成员才能具有身份服务中定义的权限。这个值需要完整的可辨uished Name: **Get-ADGroup grp-openstack | 选择 *DistinguishedName***
user_id_attribute	将 AD 值映射到用户 ID。
user_name_attribute	将 AD 值映射到 *名称*。
user_mail_attribute	将 AD 值映射到用户电子邮件地址。
user_pass_attribute	将这个值留空。
user_enabled_attribute	验证帐户是否已启用的 AD 设置。
user_enabled_mask	定义要检查的值，以确定帐户是否已启用。在未返回布尔值时使用。
user_enabled_default	表示帐户已启用的 AD 值。
user_attribute_ignore	定义 Identity Service 应该忽略的用户属性。
group_objectclass	将 AD 值映射到 *组*。

设置	Description
group_tree_dn	包含 *用户组(OU)的机构单元* (OU)。
group_filter	过滤呈现给身份服务的组。
group_id_attribute	映射用于组 ID 的 AD 值。
group_name_attribute	将 AD 值映射到组名称。
use_tls	定义要使用 TLS。如果您要使用 LDAPS 而不是 STARTTLS 加密，则需要禁用此设置。
tls_cacertfile	指定 *.crt* 证书文件的路径。
query_scope	将身份服务配置为还在嵌套子 OUs 中搜索，当查找属于 **grp-openstack** 组的成员的用户。
chase_referrals	设置为 **false**，此设置可防止 **python-ldap** 辅助所有使用匿名访问的引用。

6. 将配置文件的所有权改为 keystone 用户：

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

7. 重启 keystone 服务以应用更改：

```
# sudo docker restart keystone
```

8. 授予 **admin** 用户对域的访问权限：

注意

这不向 OpenStack admin 帐户授予实际 AD DS 域中的任何权限。在本例中，术语 *域* 指的是 OpenStack 对 keystone 域的用法。

a. 获取 **LAB** 域的 **ID** ：

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

b. 获取 *admin* 用户的 **ID** 值：

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

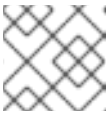
c. 获取 *admin* 角色的 **ID** 值：

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

d. 使用返回的域和 admin ID 来构造将 **admin** 用户添加到 keystone **LAB** 域的 **admin** 角色的命令：

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

e. 通过在该命令中添加 NetBIOS 名称来查看 AD DS 域中的用户列表：



注意

可能需要过些时间，LDAP 才能在重启或服务重启后变为可查询。

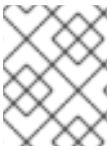
```
# openstack user list --domain LAB
```

f. 查看本地 Identity Service 数据库中的服务帐户：

```
# openstack user list --domain default
```

1.6.2. 允许 Active Directory 组成员访问项目

为了允许经过身份验证的用户访问 OpenStack 资源，建议的做法是授权特定的 Active Directory 组授予项目的访问权限。这样可保存 OpenStack 管理员，不必将每个用户分配到项目中的角色。相反，Active Directory 组在项目中被授予角色。因此，属于这些 Active Directory 组成员的 Active Directory 用户将能够访问预先确定的项目。



注意

如果您希望手动管理单个 Active Directory 用户的授权，请参阅 第 1.6.3 节 “允许 Active Directory 用户访问项目”。

本节假设 Active Directory 管理员已完成这些步骤：

- 在 Active Directory 中，创建名为 **grp-openstack-admin** 的组。
- 在 Active Directory 中，创建名为 **grp-openstack-demo** 的组。
- 根据需要，将您的 Active Directory 用户添加到上述组之一。

- 将您的 Active Directory 用户添加到 **grp-openstack** 组。
- 思考一个指定的项目。本例使用名为 **demo** 的项目，通过 **openstack project create --domain default --description "Demo Project" demo** 创建的。

这些步骤将为 AD 组分配角色。然后，组成员具有访问 OpenStack 资源的权限。

1. 检索 AD 组列表：

```
# openstack group list --domain LAB
+------------------------------------------------------------------+---------------------+
| ID                                                               | Name                |
+------------------------------------------------------------------+---------------------+
| 185277be62ae17e498a69f98a59b66934fb1d6b7f745f14f5f68953a665b8851 | grp-openstack       |
| a8d17f19f464c4548c18b97e4aa331820f9d3be52654aa8094e698a9182cbb88 | grp-openstack-admin |
| d971bb3bd5e64a454cbd0cc7af4c0773e78d61b5f81321809f8323216938cae8 | grp-openstack-demo  |
+------------------------------------------------------------------+---------------------+
```

2. 检索角色列表：

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

3. 向 Active Directory 组授予项目的访问权限，方法是将它们添加到一个或多个这些角色中。例如，如果您希望 **grp-openstack-demo** 组中的用户是 **demo** 项目的一般用户，您必须将组添加到 **_member_** 角色中：

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

因此，当域字段输入 **LAB** 时，**grp-openstack-demo** 的成员可以通过输入其 AD DS 用户名和密码来登录仪表板：

注意

如果用户收到错误 **Error: Unable to retrieve container list**，并且希望能够管理容器，则它们必须添加到 **SwiftOperator** 角色中。

1.6.3. 允许 Active Directory 用户访问项目

可以授予属于 **grp-openstack** AD 组的成员的 AD DS 用户，可授予在仪表板中登录到项目的权限：

1. *检索 AD 用户列表：*

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

2. *检索角色列表：*

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

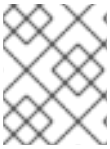
3. *通过向一个或多个角色添加项目访问权限，授予用户对项目的访问权限。例如，如果您希望* ***user1*** *是* ***demo*** *项目的一般用户，您可以将它们添加到* ***member*** *角色中：*

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

*或者，如果您希望 **user1** 是 **demo** 项目的管理员用户，您可以将它们添加到 **admin** 角色中：*

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

*因此，当在 **Domain** 字段中输入 **LAB** 时，**user1** 能够通过输入 AD DS 用户名和密码来登录仪表板：*



注意

*如果用户收到错误 **Error: Unable to retrieve container list**，并且希望能够管理容器，则它们必须添加到 **SwiftOperator** 角色中。*

1.7. 授予对 DOMAIN 选项卡的访问权限

*要允许 **admin** 用户看到 **Domain** 选项卡，您需要为它分配 **默认** 域中的 **admin** 角色：*

1. *查找 **admin** 用户的 UUID：*

```
$ openstack user list | grep admin
| a6a8adb6356f4a879f079485dad1321b | admin      |
```

2. ***将默认** 域中的 **admin** 角色添加到 **admin** 用户：*

```
$ openstack role add --domain default --user a6a8adb6356f4a879f079485dad1321b admin
```

*因此，**admin** 用户现在可以看到 **Domain** 选项卡。*

1.8. 创建新项目

*完成这些集成步骤后，当您创建新项目时，您需要确定它是在 **Default** 域中创建它，还是在刚刚创建的 keystone 域中。您可以通过考虑您的工作流以及如何管理用户帐户来实现此决定。Default 域可以被视为*

内部域，用于管理服务帐户和 admin 项目。为了分离目的，您可能想要将 AD 支持的用户保存在单独的 keystone 域中。

1.9. 仪表板登录过程中的更改

在 Identity Service 中配置多个域会在仪表板登录页面中启用一个新的 Domain 字段。用户应该输入与其登录凭据匹配的域。此字段必须手动填写 keystone 中存在的一个域。使用 openstack 命令列出可用的条目。

在本例中，AD DS 帐户需要指定 **LAB** 域。内置 keystone 帐户（如 admin ）必须将 **Default** 指定为其域：

```
# openstack domain list
+----------------------------------+---------+---------+-----------------------------------------------------------------
-----+
| ID                               | Name    | Enabled | Description                                                     |
+----------------------------------+---------+---------+-----------------------------------------------------------------
-----+
| 6800b0496429431ab1c4efbb3fe810d4 | LAB     | True    |
|
| default                          | Default | True    | Owns users and tenants (i.e. projects) available on Identity
API v2. |
+----------------------------------+---------+---------+-----------------------------------------------------------------
-----+
```

1.10. 对命令行的更改

对于某些命令，您可能需要指定适用的域。例如，在此命令中附加 **--domain LAB** 可返回 LAB 域中的用户（属于 grp-openstack 组的成员）：

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

附加 **--domain Default** 会返回内置 keystone 帐户：

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

1.11. 测试 AD DS 集成

此流程通过测试用户访问仪表板功能来验证 AD DS 集成：

1. 在 AD 中创建测试用户，并将用户添加到 **grp-openstack** AD DS 组。
2. 将用户添加到 **demo** 租户的 **_member_** 角色。
3. 使用 AD test 用户的凭据登录控制面板。
4. 单击每个选项卡，以确认它们已成功显示且无错误消息。
5. 使用控制面板构建测试实例。

注意

如果您在执行这些步骤时遇到问题，请使用内置的 admin 帐户执行第 3-5 步。如果成功，这表明 OpenStack 仍按预期工作，并且 AD → 身份集成设置中的某个位置存在问题。请参阅 第 1.13 节 "故障排除"。

1.12. 为非管理员用户创建 RC 文件

您可能需要为非管理员用户创建 RC 文件。例如：

```
$ 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
```

1.13. 故障排除

1.13.1. 测试 LDAP 连接

注意

此命令需要在主机操作系统中查找所需的证书。如需更多信息，请参阅 配置 LDAPS 证书 部分。

*使用 **ldapsearch** 对 Active Directory 域控制器远程执行测试查询。此处的成功结果表示网络连接正常工作，AD DS 服务已启动。在本例中，对端口 **636** 的 server **addc.lab.local** 执行测试查询：*

```
# 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
```

注意

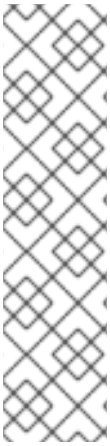
***LDAPSearch** 是 **openldap-clients** 软件包的一部分。您可以使用 **# yum install openldap-clients** 安装它*

1.13.2. 测试证书信任配置

如果您在使用 ldapsearch 进行测试时收到了错误 **Peer 的证书签发者，** 请确认您的 **TLS_CACERTDIR** 路径设置正确。例如：

- /etc/openldap/ldap.conf

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



注意

作为临时解决方案，您可能需要禁用证书验证。

不得永久配置此设置 ：

- **/etc/openldap/ldap.conf**

```
TLS_REQCERT allow
```

如果在设置这个值后 **ldapsearch** 查询可以正常工作，您可能需要检查您的证书信任是否正确。

1.13.3. 测试端口访问

使用 **nc** 检查 LDAPS 端口 **636** 是否可以远程访问。在本例中，对 server **addc.lab.local** 执行探测。按 ctrl-c 退出提示符。

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

无法建立连接可能会代表防火墙配置问题。

第 2 章 IDENTITY MANAGEMENT 集成

规划 OpenStack 身份与 Red Hat Identity Manager (IdM)集成时，请确保这两个服务都已配置并正在运行，并审查集成与用户管理和防火墙设置的影响。Red Hat Identity Manager IdM 依赖于 SRV 记录来实现负载平衡。您不应该将负载均衡器放在 IdM 的前面。

前提条件

- Red Hat Identity Management 是配置且可操作。
- Red Hat OpenStack Platform 是配置且可操作。
- DNS 名称解析可以完全正常工作，所有主机都正确注册。

权限和角色

此集成允许 IdM 用户对 OpenStack 进行验证并访问资源。OpenStack 服务帐户（如 keystone 和 glance）和授权管理（权限和角色）将保留在 Identity Service 数据库中。使用身份服务管理工具为 IdM 帐户分配权限和角色。

高可用性选项

这个配置会依赖于单个 IdM 服务器的可用性：如果 Identity Service 无法向 IdM 服务器进行身份验证，则项目用户将会受到影响。不建议将负载均衡器放在 IdM 前，但您可以将 keystone 配置为查询不同的 IdM 服务器，应该不可用。

中断要求

- 为了添加 IdM 后端，需要重启 Identity Service。
- 在 IdM 中创建其帐户后，用户才能访问仪表板。要减少停机时间，请考虑预先创建 IdM 帐户。

防火墙配置

IdM 和 OpenStack 之间的通信包括以下内容：

- 验证用户
- IdM 每隔两小时从控制器检索证书撤销列表(CRL)
- 在过期时为新证书的 certmonger 请求

注意

如果初始请求失败，则定期 certmonger 任务将继续请求新证书。

如果在 IdM 和 OpenStack 间过滤防火墙流量，您需要允许通过以下端口进行访问：

+

源	目的地	类型	端口
OpenStack Controller 节点	Red Hat Identity Management	LDAPS	TCP 636

2.1. 配置 IDM 服务器

在 IdM 服务器中运行这些命令：

1. 创建 LDAP 查找帐户。Identity Service 使用这个帐户查询 IdM LDAP 服务：

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

注意

创建之后，查看此帐户的密码过期设置。

2. 为 OpenStack 用户创建一个组，取名为 grp-openstack。只有此组成员才能在 OpenStack 身份服务中分配权限。

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

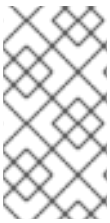
3. 设置 svc-ldap 帐户密码，并将其添加到 grp-openstack 组中：

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

4. 以 svc-ldap 用户身份登录，并在系统提示时执行密码更改：

```
# kinit svc-ldap
```

2.2. 配置 LDAPS 证书



注意

当使用多个域进行 LDAP 身份验证时，您可能会收到各种错误，如 **Unable to retrieve authorized projects**，否则 **无法识别 Peer 的证书签发者**。如果 keystone 对某个域使用不正确的证书，会出现这种情况。作为临时解决方案，将所有 LDAPS 公钥合并到一个 **.crt** 捆绑包中，并配置所有 keystone 域以使用此文件。

1. 在 IdM 环境中，找到 LDAPS 证书。此文件可以使用 /etc/openldap/ldap.conf ：

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

2. 将文件复制到运行 keystone 服务的 OpenStack 节点。例如，这个命令使用 scp 将 ca.crt 复制到名为 node.lab.local 的节点上：

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

3. 在 OpenStack 节点上，将 .crt 转换为 .pem ：

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

4. 将 .crt 复制到证书目录中。这是 keystone 服务用来访问证书的位置：

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

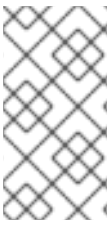
注意

另外，如果您需要运行诊断命令，如 **ldapsearch**，您还需要将证书添加到 RHEL 证书存储中。例如：

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

2.3. 配置身份服务

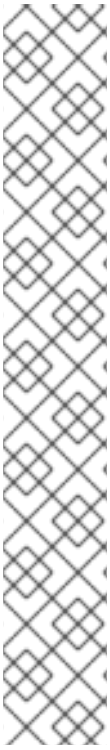
这些步骤准备 Identity Service 以与 IdM 集成。



注意

如果使用 director，请注意以下引用的配置文件由 Puppet 管理。因此，每当运行 **openstack overcloud deploy** 过程时，您添加的任何自定义配置可能会被覆盖。要将这些设置应用到基于 director 的部署中，请参阅 第 4 章 将特定于域的 LDAP 后端与 director 一起使用。

2.3.1. 配置控制器



注意

如果要更新任何配置文件，您需要注意某些 OpenStack 服务现在在容器内运行；这适用于 keystone、nova 和 cinder 等。因此，有一些管理实践需要考虑：

- 不要更新您在物理节点的主机操作系统上找到的任何配置文件，例如：**/etc/cinder/cinder.conf**。这是因为容器化服务不引用此文件。
- 不要更新容器中运行的配置文件。这是因为重启容器后会丢失任何更改。
  相反，如果您需要对容器化服务添加任何更改，则需要更新用于生成容器的配置文件。它们存储在 **/var/lib/config-data/puppet-generated/** 中

  例如：

- 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**
  重启容器后，所有更改都会被应用。例如：**sudo docker restart keystone**

在运行 keystone 服务的控制器上执行这个步骤：

1. 配置 SELinux：

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

输出可能包含类似于如下的消息。可以忽略它们：

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

2. 创建 域 目录：

```
# 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. 配置身份服务以使用多个后端：

注意

您可能需要使用 **yum install crudini** 安装 crudini。

```
# 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
```

注意

如果使用 director，请注意 **/var/lib/config-data/puppet-generated/keystone/etc/keystone/keystone.conf** 由 Puppet 管理。因此，每当运行 **openstack overcloud deploy** 过程时，您添加的任何自定义配置可能会被覆盖。因此，您可能需要每次手动重新添加此配置。有关基于 director 的部署，请参阅 第 4 章 将特定于域的 LDAP 后端与 director 一起使用 。

4. 在控制面板中启用多个域.将这些行添加到 /var/lib/config-data/puppet-generated/horizon/etc/openstack-dashboard/local_settings 中：

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

注意

如果使用 director，请注意 **/var/lib/config-data/puppet-generated/horizon/etc/openstack-dashboard/local_settings** 由 Puppet 管理。因此，每当运行 **openstack overcloud deploy** 过程时，您添加的任何自定义配置可能会被覆盖。因此，您可能需要每次手动重新添加此配置。

重启 horizon 容器以应用设置：

```
$ sudo docker restart horizon
```

5. 配置额外后端：

a. *为 IdM 集成创建 keystone 域。您需要决定新 keystone 域的名称，然后创建该域。例如，这个命令会创建一个名为* ***LAB*** *的 Keystone 域：*

```
$ openstack domain create LAB
```

b. *创建配置文件：*
要添加 IdM 后端，请在名为 ***/var/lib/config-data/puppet-generated/keystone/etc/domains/keystone. LAB.conf*** *的新文件中输入 LDAP 设置。您需要编辑以下示例设置，以适应 IdM 部署：*

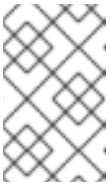
```
[ldap]
url =  ldaps://idm.lab.local
user = uid=svc-ldap,cn=users,cn=accounts,dc=lab,dc=local
user_filter = (memberOf=cn=grp-openstack,cn=groups,cn=accounts,dc=lab,dc=local)
password = RedactedComplexPassword
user_tree_dn = cn=users,cn=accounts,dc=lab,dc=local
user_objectclass = inetUser
user_id_attribute = 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
```

每个设置的说明：

设置	Description
url	用于身份验证的 IdM 服务器。使用 LDAPS 端口 **636**。
user	用于 LDAP 查询的 IdM 中的帐户。
password	以上使用的 IdM 帐户的纯文本密码。
user_filter	过滤呈现给身份服务的用户。因此，只有 **grp-openstack** 组的成员才能具有身份服务中定义的权限。
user_tree_dn	IdM 中 OpenStack 帐户的路径。

设置	Description
user_objectclass	定义 LDAP 用户的类型。对于 IdM，使用 **inetUser** 类型。
user_id_attribute	映射用于用户 ID 的 IdM 值。
user_name_attribute	将 IdM 值映射到 *名称*。
user_mail_attribute	映射用于用户电子邮件地址的 IdM 值。
user_pass_attribute	将这个值留空。



注意

*与 IdM 组集成将只返回直接成员，而不返回嵌套组。因此，依赖 **LDAP_MATCHING_RULE_IN_CHAIN** 或 **memberof** 的查询：**1.2.840.113556.1.4.1941** 当前无法使用 IdM。*

6. *将配置文件的所有权改为 keystone 用户：*

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

7. *授予 admin 用户对域的访问权限：*

注意

这不向 OpenStack admin 帐户授予 IdM 中的任何权限。在本例中，术语域指的是 OpenStack 对 keystone 域的用法。

a. *获取 LAB 域的 **ID** ：*

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

b. *获取 admin 用户的 **ID** 值：*

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

| 3d75388d351846c6a880e53b2508172a | admin      |
```

c. *获取 admin 角色的 **ID** 值：*

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

d. *使用返回的域和 admin ID 来构造将 admin 用户添加到 keystone LAB 域的 admin 角色的命令：*

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

8. *重启 keystone 服务以应用更改：*

```
$ sudo docker restart keystone
```

9. *通过在命令中添加 keystone 域名来查看 IdM 域中的用户列表：*

```
$ openstack user list --domain LAB
```

10. *查看本地 keystone 数据库中的服务帐户：*

```
$ openstack user list --domain default
```

2.3.2. 允许 IdM 组成员访问项目

要允许经过身份验证的用户访问 OpenStack 资源，推荐的方法是授权某些 IdM 组授予项目访问权限。这样可保存 OpenStack 管理员，不必将每个用户分配到项目中的角色。相反，IdM 组在项目中被授予角色。因此，属于这些 IdM 组成员的 IdM 用户将能够访问预先确定的项目。

注意

如果您希望手动管理单个 IdM 用户的授权，请查看 第 2.3.3 节 "允许 IdM 用户访问项目"。

本节假设 IdM 管理员已经完成这些步骤：

- *在 IdM 中创建名为 **grp-openstack-admin** 的组。*
- *在 IdM 中创建一个名为 **grp-openstack-demo** 的组。*
- *根据需要将 IdM 用户添加到上述组之一。*
- *将 IdM 用户添加到 **grp-openstack** 组。*
- *思考一个指定的项目。本例使用名为 **demo** 的项目，通过 **openstack project create --domain default --description "Demo Project" demo** 创建的。*

这些步骤为 IdM 组分配角色。然后，组成员具有访问 OpenStack 资源的权限。

1. *检索 IdM 组列表：*

```
$ openstack group list --domain LAB
+------------------------------------------------------------------+--------------------+
| ID                                                               | Name               |
+------------------------------------------------------------------+--------------------+
| 185277be62ae17e498a69f98a59b66934fb1d6b7f745f14f5f68953a665b8851 | grp-
openstack       |
| a8d17f19f464c4548c18b97e4aa331820f9d3be52654aa8094e698a9182cbb88 | grp-
openstack-admin |
| d971bb3bd5e64a454cbd0cc7af4c0773e78d61b5f81321809f8323216938cae8 | grp-
openstack-demo  |
+------------------------------------------------------------------+--------------------+
```

2. *检索角色列表：*

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

3. *将 IdM 组授予项目的访问权限，方法是将它们添加到一个或多个这些角色中。例如，如果您希望* ***grp-openstack-demo*** *组中的用户是* ***demo*** *项目的一般用户，您必须将组添加到* ***_member_*** *角色中：*

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

因此，当域字段输入 ***LAB*** *时，* ***grp-openstack-demo*** *的成员可以通过输入其 IdM 用户名和密码来登录控制面板：*

注意

如果用户收到错误 **Error: Unable to retrieve container list**，并且希望能够管理容器，则它们必须添加到 **SwiftOperator** 角色中。

2.3.3. 允许 IdM 用户访问项目

属于 **grp-openstack** IdM 组的成员的 IdM 用户被授予在仪表板中登录到项目的权限：

1. 检索 IdM 用户列表：

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

2. 检索角色列表：

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

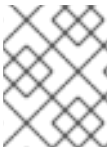
3. 通过向一个或多个角色添加项目访问权限，授予用户对项目的访问权限。例如，如果您希望 **user1** 是 **demo** 项目的一般用户，您可以将它们添加到 **member** 角色中：

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

或者，如果您希望 **user1** 是 **demo** 项目的管理员用户，您可以将它们添加到 **admin** 角色中：

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

因此，在 **Domain** 字段中输入 **LAB** 时，**user1** 就可以通过输入其 IdM 用户名和密码来登录到仪表板：



注意

如果用户收到错误 ***Error: Unable to retrieve container list****，并且希望能够管理容器，则它们必须添加到* ***SwiftOperator*** *角色中。*

2.4. 授予对 DOMAIN 选项卡的访问权限

要允许 ***admin*** *用户看到* ***Domain*** *选项卡，您需要为它分配* ***默认*** *域中的* ***admin*** *角色：*

1. *查找* ***admin*** *用户的 UUID：*

```
$ openstack user list | grep admin
| a6a8adb6356f4a879f079485dad1321b | admin      |
```

2. ***将默认*** *域中的* ***admin*** *角色添加到* ***admin*** *用户：*

```
$ openstack role add --domain default --user a6a8adb6356f4a879f079485dad1321b admin
```

因此， ***admin*** *用户现在可以看到* ***Domain*** *选项卡。*

2.5. 创建新项目

完成这些集成步骤后，当您创建新项目时，您需要确定它是在 ***Default*** *域中创建它，还是在刚刚创建的 keystone 域中。您可以通过考虑您的工作流以及如何管理用户帐户来实现此决定。* ***Default*** *域可以被视为内部域（用于服务帐户和* ***admin*** *项目），因此您的 AD 支持的用户可能被放置到不同的 keystone 域中；这严格不需要与 IdM 用户所处的相同 Keystone 域，用于分离目的，可能有多个 keystone 域。*

2.5.1. 仪表板登录过程中的更改

在 Identity Service 中配置多个域会在仪表板登录页面中启用一个新的 Domain 字段。
用户应该输入与其登录凭据匹配的域。此字段必须手动填写 keystone 中存在的一个域。使用 openstack 命令列出可用的条目。

在本例中，IdM 帐户需要指定 ***LAB*** *域。内置 keystone 帐户（如 admin ）必须将* ***Default*** *指定为其域：*

```
$ openstack domain list
+----------------------------------+---------+---------+-----------------------------------------------------------------
-----+
| ID                               | Name    | Enabled | Description                                                     |
+----------------------------------+---------+---------+-----------------------------------------------------------------
-----+
| 6800b0496429431ab1c4efbb3fe810d4 | LAB     | True    |
|
| default                          | Default | True    | Owns users and tenants (i.e. projects) available on Identity
API v2. |
+----------------------------------+---------+---------+-----------------------------------------------------------------
-----+
```

2.5.2. 对命令行的更改

对于某些命令，您可能需要指定适用的域。例如，在此命令中附加 **--domain LAB** 可返回 LAB 域中的用户（属于 grp-openstack 组的成员）：

```
$ openstack user list --domain LAB
```

附加 **--domain Default** 会返回内置 keystone 帐户：

```
$ openstack user list --domain Default
```

2.5.3. 测试 IdM 集成

此流程通过测试用户访问仪表板功能来验证 IdM 集成：

1. 在 IdM 中创建测试用户，并将用户添加到 **grp-openstack** IdM 组中。
2. 将用户添加到 **demo** 租户的 **_member_** 角色。
3. 使用 IdM test 用户的凭据登录控制面板。
4. 单击每个选项卡，以确认它们已成功显示且无错误消息。
5. 使用控制面板构建测试实例。

注意

如果您在执行这些步骤时遇到问题，请使用内置的 admin 帐户执行第 3-5 步。如果成功，这表明 OpenStack 仍按预期工作，并且 IdM → Identity 集成设置中的某个问题存在。请参阅 第 2.7 节 “故障排除”。

2.6. 为非管理员用户创建 RC 文件

您可能需要为非管理员用户创建 RC 文件。例如：

```
$ 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
```

2.7. 故障排除

2.7.1. 测试 LDAP 连接

使用 ldapsearch 对 IdM 服务器远程执行测试查询。此处的成功结果表示网络连接正常工作，IdM 服务已启动。在本例中，测试查询针对端口 636 上的服务器 idm.lab.local 执行：

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

注意

LDAPSearch 是 openldap-clients 软件包的一部分。您可以使用 **# yum install openldap-clients** 来安装它。

2.7.2. 测试端口访问

使用 nc 检查 LDAPS 端口(636)是否可以远程访问。在本例中，探测针对服务器 idm.lab.local 执行。按 ctrl-c 退出提示符。

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

无法建立连接可能会代表防火墙配置问题。

第 3 章 使用 NOVAJOIN 与 IDM 集成

novajoin 允许您在部署过程中使用 Red Hat Identity Manager (IdM) 注册节点。因此，您可以将 IdM 功能与您的 OpenStack 部署集成，包括身份、kerberos 凭证和访问控制。

注意

通过 novajoin 的 IdM 注册目前仅适用于 undercloud 和 overcloud 节点。overcloud 实例的 novajoin 集成应该在以后的版本中被支持。

3.1. 在 UNDERCLOUD 中安装并配置 NOVAJOIN

3.1.1. 将 undercloud 添加到 CA

在部署 overcloud 之前，您必须将 undercloud 添加到证书颁发机构(CA)中：

1. 在 undercloud 节点上，安装 **python-novajoin** 软件包：

```
$ sudo yum install python-novajoin
```

2. 在 undercloud 节点上，运行 **novajoin-ipa-setup** 脚本，调整值以适合您的部署：

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

在以下部分中，您将使用生成的一次性密码(OTP)注册 undercloud。

3.1.2. 将 undercloud 添加到 IdM

此流程使用 IdM 注册 undercloud，并配置 novajoin。在 **undercloud.conf** 中配置以下设置（在 **[DEFAULT]** 部分中）：

1. 默认情况下，禁用 novajoin 服务。启用它：

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

2. 您需要设置一个一次性密码(OTP)，以使用 IdM 注册 undercloud 节点：

```
ipa_otp = <otp>
```

3. 确保 neutron 的 DHCP 服务器提供 overcloud 的域名与 IdM 域（您小写的 kerberos 域）匹配：

```
overcloud_domain_name = <domain>
```

4. 为 undercloud 设置适当的主机名：

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

5. 将 IdM 设置为 undercloud 的名称服务器：

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

6. 对于较大的环境，您需要查看 novajoin 连接超时值。在 **undercloud.conf** 中，添加对名为 **undercloud-timeout.yaml** 的新文件的引用：

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

在 **undercloud-timeout.yaml** 中添加以下选项。您可以指定超时值（以秒为单位），例如 **5** ：

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

7. 保存 **undercloud.conf** 文件。

8. 运行 undercloud 部署命令，将更改应用到现有的 undercloud：

```
$ openstack undercloud install
```

验证

1. 检查 **keytab** 文件是否有 undercloud 的密钥条目：

```
[root@undercloud-0 ~]# klist -kt
Keytab name: FILE:/etc/krb5.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
   1 04/28/2020 12:22:06 host/undercloud-0.redhat.local@REDHAT.LOCAL
   1 04/28/2020 12:22:06 host/undercloud-0.redhat.local@REDHAT.LOCAL


[root@undercloud-0 ~]# klist -kt /etc/novajoin/krb5.keytab
Keytab name: FILE:/etc/novajoin/krb5.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
   1 04/28/2020 12:22:26 nova/undercloud-0.redhat.local@REDHAT.LOCAL
   1 04/28/2020 12:22:26 nova/undercloud-0.redhat.local@REDHAT.LOCAL
```

2. 使用主机原则测试系统 **/etc/krb.keytab** 文件：

```
[root@undercloud-0 ~]# kinit -k
[root@undercloud-0 ~]# klist
Ticket cache: KEYRING:persistent:0:0
Default principal: host/undercloud-0.redhat.local@REDHAT.LOCAL

Valid starting       Expires              Service principal
05/04/2020 10:34:30  05/05/2020 10:34:30  krbtgt/REDHAT.LOCAL@REDHAT.LOCAL

[root@undercloud-0 ~]# kdestroy
Other credential caches present, use -A to destroy all
```

3. 使用 nova 原则测试 novajoin **/etc/novajoin/krb.keytab** 文件：

```
[root@undercloud-0 ~]# kinit -kt /etc/novajoin/krb5.keytab 'nova/undercloud-
0.redhat.local@REDHAT.LOCAL'
[root@undercloud-0 ~]# klist
Ticket cache: KEYRING:persistent:0:0
Default principal: nova/undercloud-0.redhat.local@REDHAT.LOCAL

Valid starting       Expires              Service principal
05/04/2020 10:39:14  05/05/2020 10:39:14  krbtgt/REDHAT.LOCAL@REDHAT.LOCAL
```

3.2. 在 OVERCLOUD 中安装和配置 NOVAJOIN

这些部分描述了如何使用 IdM 注册 overcloud 节点。

3.2.1. 配置 overcloud DNS

要自动检测 IdM 环境并更容易注册，请考虑使用 IdM 作为您的 DNS 服务器：

1. 连接到 undercloud：

```
$ source ~/stackrc
```

2. 将 control plane 子网配置为使用 IdM 作为 DNS 名称服务器：

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

3. 在环境文件中设置 **DnsServers** 参数以使用您的 IdM 服务器：

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

此参数通常在自定义 **network-environment.yaml** 文件中定义。

3.2.2. 将 overcloud 配置为使用 novajoin

1. 要启用 IdM 集成，请创建一个 **/usr/share/openstack-tripleo-heat-templates/environments/predictable-placement/custom-domain.yaml** 环境文件的副本：

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

2. 编辑 **/home/stack/templates/custom-domain.yaml** 环境文件，并将 **CloudDomain** 和 **CloudName*** 值设置为适合您的部署。例如：

```
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. *在 overcloud 部署过程中包含以下环境文件：*

   - ***/usr/share/openstack-tripleo-heat-templates/environments/ssl/enable-internal-tls.yaml***
   - ***/usr/share/openstack-tripleo-heat-templates/environments/ssl/tls-everywhere-endpoints-dns.yaml***
   - ***/home/stack/templates/custom-domain.yaml***
     例如：

```
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 \
```

   因此，部署的 overcloud 节点将自动注册到 IdM。

4. *这仅为内部端点设置 TLS。对于外部端点，您可以使用普通的方法为 **/usr/share/openstack-tripleo-heat-templates/environments/ssl/enable-tls.yaml** 环境文件（必须修改它来添加自定义证书和密钥）。因此，您的 **openstack deploy** 命令与此类似：*

```
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
```

5. *另外，您还可以使用 IdM 发布公共证书。在这种情况下，您需要使用 **/usr/share/openstack-tripleo-heat-templates/environments/services/haproxy-public-tls-certmonger.yaml** 环境文件。例如：*

```
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 /usr/share/openstack-tripleo-heat-templates/environments/services/haproxy-public-tls-certmonger.yaml
```

3.3. 在 IDM 中验证节点

1. *在 IdM 中找到 overcloud 节点，并确认主机条目包含 **Keytab:True** ：*

```
$ 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. SSH 到节点，并确认 sssd 可以查询 IdM 用户。例如，查询名为 **susan** 的 IdM 用户：

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

3.4. 为 NOVAJOIN 配置 DNS 条目

如果您使用 **haproxy-public-tls-certmonger.yaml** 模板为端点发布公共证书，那么您将需要为 Novajoin 所用的 VIP 端点手动创建 DNS 条目：

1. 识别 overcloud 网络。您可以在 **/home/stack/virt/network/network-environment.yaml** 中找到它们：

```
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
```

2. 为每个 overcloud 网络创建虚拟 IP 地址(VIP)列表。例如：/home/stack/virt/public_vip.yaml

```
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'}]
```

3. 在 IdM 中为每个 VIP 添加 DNS 条目。您可能还需要创建新区域。以下示例演示了 IdM 的 DNS 记录和区创建：

```
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
```

第 4 章 将特定于域的 LDAP 后端与 DIRECTOR 一起使用

Red Hat OpenStack Platform director 可将 keystone 配置为使用一个或多个 LDAP 后端。这种方法会导致为每个 keystone 域创建单独的 LDAP 后端。

4.1. 设置配置选项

对于使用 Red Hat OpenStack Platform director 的部署，您需要在 heat 模板中将 **KeystoneLDAPDomainEnable** 标志设置为 **true** ；因此，这会在 keystone 中配置 **domain_specific_drivers_enabled** 选项（在 **identity** 配置组中）。

注意

域配置文件的默认目录设置为 **/etc/keystone/domains/**。您可以使用 **keystone::domain_config_directory** hiera 键设置所需的路径来覆盖它，并将它添加为环境文件中的 **ExtraConfig** 参数。

您还必须添加 LDAP 后端配置的规格。这是使用 **tripleo-heat-templates** 中的 **KeystoneLDAPBackendConfigs** 参数实现，然后您可以在其中指定所需的 LDAP 选项。

4.2. 配置 DIRECTOR 部署

1. 创建 **keystone_domain_specific_ldap_backend.yaml** 环境文件的副本：

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

2. 编辑 **/home/stack/templates/keystone_domain_specific_ldap_backend.yaml** 环境文件，并将值设置为适合您的部署。例如，这些条目为名为 **testdomain** 的 keystone 域创建 LDAP 配置：

```
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
```

3. 您还可以配置环境文件来指定多个域。例如：

```
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
...
```

*这将产生两个名为 **domain1** 和 **domain2** 的域；每个域都有自己的配置，每个域都有不同的 LDAP 域。*