OpenStack-Ansible Documentation: os_keystone role

Release 18.1.0.dev443

OpenStack-Ansible Contributors

Oct 26, 2023

CONTENTS

CHAPTER

ONE

CONFIGURING THE IDENTITY SERVICE (KEYSTONE) (OPTIONAL)

Customize your keystone deployment in `/etc/openstack_deploy/user_variables.yml`.

1.1 Securing keystone communication with SSL certificates

The OpenStack-Ansible project provides the ability to secure keystone communications with self-signed or user-provided SSL certificates. By default, self-signed certificates are in use. However, you can provide your own certificates by using the following Ansible variables in `/etc/openstack_deploy/user_variables.yml`:

```
keystone_user_ssl_cert:          # Path to certificate
keystone_user_ssl_key:           # Path to private key
keystone_user_ssl_ca_cert:       # Path to CA certificate
```

Note: If you are providing certificates, keys, and CA file for a CA without chain of trust (or an invalid/self-generated ca), the variables `keystone_service_internaluri_insecure` and `keystone_service_adminuri_insecure` should be set to `True`.

Refer to Securing services with SSL certificates for more information on these configuration options and how you can provide your own certificates and keys to use with keystone.

1.2 Implementing LDAP (or Active Directory) backends

You can use the built-in keystone support for services if you already have LDAP or Active Directory (AD) infrastructure on your deployment. Keystone uses the existing users, groups, and user-group relationships to handle authentication and access control in an OpenStack deployment.

Note: We do not recommend configuring the default domain in keystone to use LDAP or AD identity backends. Create additional domains in keystone and configure either LDAP or active directory backends for that domain.

This is critical in situations where the identity backend cannot be reached due to network issues or other problems. In those situations, the administrative users in the default domain would still be able to authenticate to keystone using the default domain which is not backed by LDAP or AD.

You can add domains with LDAP backends by adding variables in /etc/openstack_deploy/user_variables.yml. For example, this dictionary adds a new keystone domain called Users that is backed by an LDAP server:

```
keystone_ldap:
  Users:
    url: "ldap://10.10.10.10"
    user: "root"
    password: "secrete"
```

Adding the YAML block above causes the keystone playbook to create a /etc/keystone/domains/keystone.Users.conf file within each keystone service container that configures the LDAP-backed domain called Users.

You can create more complex configurations that use LDAP filtering and consume LDAP as a read-only resource. The following example shows how to apply these configurations:

```
keystone_ldap:
  MyCorporation:
      url: "ldaps://ldap.example.com"
      user_tree_dn: "ou=Users,o=MyCorporation"
      group_tree_dn: "cn=openstack-users,ou=Users,o=MyCorporation"
      user_objectclass: "inetOrgPerson"
      user_id_attribute: "cn"
      user_name_attribute: "uid"
      user_filter: "(groupMembership=cn=openstack-users,ou=Users,
↪o=MyCorporation)"
```

In the *MyCorporation* example above, keystone uses the LDAP server as a read-only resource. The configuration also ensures that keystone filters the list of possible users to the ones that exist in the cn=openstack-users,ou=Users,o=MyCorporation group.

Horizon offers multi-domain support that can be enabled with an Ansible variable during deployment:

```
horizon_keystone_multidomain_support: True
```

Enabling multi-domain support in horizon adds the Domain input field on the horizon login page and it adds other domain-specific features in the keystone section.

More details regarding valid configuration for the LDAP Identity backend can be found in the Keystone Developer Documentation and the OpenStack Administrator Guide.

CHAPTER

TWO

SCENARIO - CONFIGURING KEYSTONE FEDERATION

Federation for keystone can be utilised in two ways:

- Supporting keystone as a Service Provider (SP): consuming identity assertions issued by an external Identity Provider, such as SAML assertions or OpenID Connect claims.
- Supporting keystone as an Identity Provider (IdP): fulfilling authentication requests on behalf of Service Providers.

> **Important:** It is also possible to have one keystone act as an SP that consumes Identity from another keystone acting as an IdP. This will be discussed further in this document.

In keystone federation, the IdP and SP exchange information securely to enable a user on the IdP cloud to access resources of the SP cloud.

The following procedure describes how to set up federation:

1. Configure keystone SPs.
2. Configure the IdP:
   - Configure keystone as an IdP.
   - Configure Active Directory Federation Services (ADFS) 3.0 as an IdP.
3. Configure the service provider:
   - Configure keystone as a federated service provider.
   - Configure keystone mappings.
4. Run the authentication wrapper to use keystone-as-a-Service-Provider federation.

2.1 Configuring keystone-to-keystone federation

In keystone-to-keystone federation (k2k), the IdP and SP keystone instances exchange information securely to enable a user on the IdP cloud to access resources of the SP cloud.

> **Important:** This section applies only to federation between keystone IdP and keystone SP. It does not apply to non-keystone IdP.

The k2k authentication flow involves the following steps:

1. Log in to the IdP with your credentials.
2. Send a request to the IdP to generate an assertion for a given SP.
3. Submit the assertion to the SP on the configured `sp_url` endpoint. The Shibboleth service running on the SP receives the assertion and verifies it. If it is valid, a session with the client starts and returns the session ID in a cookie.
4. Connect to the SP on the configured `auth_url` endpoint, providing the Shibboleth cookie with the session ID. The SP responds with an unscoped token that you use to access the SP.
5. You connect to the keystone service on the SP with the unscoped token, and the desired domain and project, and receive a scoped token and the service catalog.
6. With your token, you can now make API requests to the endpoints in the catalog.

2.1.1 Keystone-to-keystone federation authentication wrapper

The following steps above involve manually sending API requests.

Note: The infrastructure for the command line utilities that performs these steps for the user does not exist.

To obtain access to a SP cloud, OpenStack-Ansible provides a script that wraps the above steps. The script is called `federated-login.sh` and is used as follows:

```
# ./scripts/federated-login.sh -p project [-d domain] sp_id
```

- `project` is the project in the SP cloud that you want to access.
- `domain` is the domain in which the project lives (the default domain is used if this argument is not given).
- `sp_id` is the unique ID of the SP. This is given in the IdP configuration.

The script outputs the results of all the steps in the authentication flow to the console. At the end, it prints the available endpoints from the catalog and the scoped token provided by the SP.

Use the endpoints and token with the openstack command line client as follows:

```
# openstack --os-token=<token> --os-url=<service-endpoint> [options]
```

Or, alternatively:

```
# export OS_TOKEN=<token>
# export OS_URL=<service-endpoint>
# openstack [options]
```

Ensure you select the appropriate endpoint for your operation. For example, if you want to work with servers, the `OS_URL` argument must be set to the compute endpoint.

Note: At this time, the OpenStack client is unable to find endpoints in the service catalog when using a federated login.

2.2 Configure keystone as a federated Service Provider

In OpenStack-Ansible, the default installation of keystone uses NGINX and `uWSGI`, however when deploying federation we instead use Apache with `uWSGI`. The additional configuration of keystone as a federation service provider adds Apache `mod_shib` or `mod_auth_openidc` and configures it to respond to authentication specific request locations from a client.

Note: There are alternative methods of implementing federation, but at this time only SAML2-based federation using the Shibboleth SP via mod_shib or OIDC-based federation using mod_auth_openidc are supported in Openstack-Ansible. Currently only one of these apache modules is supported at a time, with a single trusted IdP in the keystone_sp.trusted_idp_list.

When requests are sent to those locations, Apache hands off the request to the `shibd` daemon or `mod_auth_openidc` module.

Note: Handing off happens only with requests pertaining to authentication.

2.2.1 Service provider configuration using keystone_sp

keystone_sp is a dictionary variable which contains various settings that describe both the SP and the IDPs it trusts. The values required in keystone_sp will differ slightly between Shibboleth SAML based deployments and OIDC mod_auth_openidc deployments.

The following settings can be set to configure a service provider (SP) for both SAML or OIDC deployments:

1. `apache_mod` can be used to switch between mod_shib and mod_auth_openidc. If left undefined or misspelled Shibboleth will be used by default. Valid values are shibboleth or mod_auth_openidc. (Optional)
2. `cadf_notifications` toggle Cloud Auditing Data Federation (CADF) Notifications. These are off by default. Valid values are true or false. More information is available in the keystone developer documentation.
3. `cadf_notifications_opt_out` ignore producing certain CADF notifications when CADF notifications are enabled. The keystone documentation recommends to opt out of: identity.authenticate.success, identity.authenticate.pending and identity.authenticate.failed notifications as they are noisy.
4. `trusted_dashboard_list` is the list of trusted URLs that keystone accepts redirects for Web Single-Sign. This setting ensures that keystone only sends token data back to trusted servers. This is performed as a precaution, specifically to prevent man-in-the-middle (MITM) attacks. This list contains all URLs that horizon is presented on, suffixed by `/auth/websso/` which is the path for horizons WebSSO component. The trusted_dashboard_list may comprise of IP addresses and/or DNS names.
5. `trusted_idp_list` is a dictionary attribute containing the list of settings which correspond to each trusted IDP for the SP.

The following are attributes that can be set on an entry in the `trusted_idp_list`. Note while trusted_idp_list is a list, it can currently only support one entry.

1. `name` is the name of the IDP used within Keystone and is the name shown by default in Horizon within the IDP dropdown.

2. `display_name` an alternative name for your IDP to be displayed on Horizon, should your name attribute not be user friendly. (Optional)

3. `domain_id` is the domain the IDP will be created in. If this is not set on creation of a new IDP, a new domain will be autogenerated with a random ID. It is this domain that will become associated to the IDP. (Optional)

4. **`entity_ids` is a list of reference entity IDs. This specifys the** redirection of the login request to the SP when authenticating to IDP.

5. `federated_identities` is a mapping list of domain, project, group, and users. See Configure Identity Service (keystone) mappings for more information. (Optional)

6. `protocols` is a list of protocols supported for the IDP and the set of mappings and attributes for each protocol. This only supports protocols with the name `saml2` or `openid`.

7. `mapping` is the local to remote mapping configuration for federated users. See Configure Identity Service (keystone) mappings for more information.

2.2.2 Service provider configuration for SAML using Shibboleth

In addition to the attributes in keystone_sp defined above the following attributes can be set in the entry for the `trusted_idp_list` for Shibboleth SAML-based deployments.

1. `cert_duration_years` designates the valid duration for the SPs signing certificate (for example, `/etc/shibboleth/sp-key.pem`).

2. `metadata_uri` is the location of the IdPs metadata. This provides the SP with the signing key and all the IdPs supported endpoints.

3. `metadata_file` is the file name of the local cached version of the metadata which will be stored in `/var/cache/shibboleth/`.

4. `metadata_reload` is the number of seconds between metadata refresh polls.

5. `protocols.attributes` add to the Shibboleth attributes mapping directory. See Configure Identity Service (keystone) mappings for more information. .. _Configure Identity Service (keystone) mappings: configure-federation-mapping.html

Below is an example keystone_sp for setting up keystone to be a SAML-based service provider to an IDP using Shibboleth with CADF notifications on.

```
keystone_sp:
  apache_mod: 'shibboleth'
  cert_duration_years: 5
  cadf_notifications: true
  cadf_notifications_opt_out:
    - identity.authenticate.failed
    - identity.authenticate.pending
    - identity.authenticate.success
  trusted_dashboard_list:
    - "https://{{ external_lb_vip_address }}/auth/websso/"
    - "https://{{ horizon_server_name }}/auth/websso/"
```

(continues on next page)

(continued from previous page)

```
trusted_idp_list:
  - name: 'testshib-idp'
    entity_ids:
      - 'https://idp.testshib.org/idp/shibboleth'
    metadata_uri: 'http://www.testshib.org/metadata/testshib-
↪providers.xml'
    metadata_file: 'metadata-testshib-idp.xml'
    metadata_reload: 1800
    federated_identities:
      - domain: default
        project: fedproject
        group: fedgroup
        role: _member_
    protocols:
      - name: saml2
        mapping:
          name: testshib-idp-mapping
          rules:
            - remote:
                - type: eppn
              local:
                - group:
                    name: fedgroup
                    domain:
                      name: Default
                - user:
                    name: '{0}'
```

Handle the shibd service configuration through the following files in /etc/shibboleth/ in the keystone containers:

- **sp-cert.pem, sp-key.pem: The os-keystone-install.yml playbook** uses these files generated on the first keystone container to replicate them to the other keystone containers. The SP and the IdP use these files as signing credentials in communications.
- shibboleth2.xml: The os-keystone-install.yml playbook writes the files contents, basing on the structure of the configuration of the keystone_sp attribute in the /etc/openstack_deploy/user_variables.yml file. It contains the list of trusted IdPs, the entityID by which the SP is known, and other facilitating configurations.
- attribute-map.xml: The os-keystone-install.yml playbook writes the files contents, basing on the structure of the configuration of the keystone_sp attribute in the /etc/openstack_deploy/user_variables.yml file. It contains the default attribute mappings that work for any basic Shibboleth-type IDP setup, but also contains any additional attribute mappings set out in the structure of the keystone_sp attribute.
- shibd.logger: This file is left alone by OpenStack-Ansible. It is useful when troubleshooting issues with federated authentication, or when discovering what attributes published by an IdP are not currently being understood by your SPs attribute map. To enable debug logging, change log4j.rootCategory=INFO to log4j.rootCategory=DEBUG at the top of the file. The log file is output to /var/log/shibboleth/shibd.log.

2.2.3 Service provider configuration for keystone-to-keystone (k2k) with Shibboleth

Please set the following attributes:

1. `keystone_public_endpoint` is automatically set by default to the public endpoints URI. This performs redirections and ensures token references refer to the public endpoint.
2. `horizon_keystone_endpoint` is automatically set by default to the public v3 API endpoint URL for keystone. Web-based single sign-on for horizon requires the use of the keystone v3 API. The value for this must use the same DNS name or IP address registered in the SSL certificate used for the endpoint.
3. It is a requirement to have a HTTPS public endpoint for the keystone endpoint if the IdP is ADFS. Keystone or an SSL offloading load balancer provides the endpoint.
4. Set `keystone_service_publicuri_proto` to https. This ensures keystone publishes https in its references and ensures that Shibboleth is configured to know that it expects SSL URLs in the assertions (otherwise it will invalidate the assertions).
5. Most professional IDPs such as ADFS and Google require that a trusted SP have a trusted certificate that is not self-signed.
6. Ensure the endpoint URI and the certificate match when using SSL for the keystone endpoint. For example, if the certificate does not have the IP address of the endpoint, then the endpoint must be published with the appropriate name registered on the certificate. When using a DNS name for the keystone endpoint, both `keystone_public_endpoint` and `horizon_keystone_endpoint` must be set to use the DNS name.
7. `horizon_endpoint_type` must be set to `publicURL` to ensure that horizon uses the public endpoint for all its references and queries.

Below is an example keystone_sp for setting up keystone to be a SAML-based service provider to a keystone IDP using Shibboleth with CADF notifications on. For k2k the `trusted_dashboard_list` may include Horizon entries from multiple clouds.

```yaml
keystone_sp:
  apache_mod: 'shibboleth'
  cert_duration_years: 5
  cadf_notifications: true
  cadf_notifications_opt_out:
    - identity.authenticate.failed
    - identity.authenticate.pending
    - identity.authenticate.success
  trusted_dashboard_list:
    - "https://{{ horizon_server_name }}/auth/websso/"
    - "https://{{ external_lb_vip_address }}/auth/websso/"
  trusted_idp_list:
    - name: "keystone-idp"
      entity_ids:
         - 'https://keystone-idp:5000/v3/OS-FEDERATION/saml2/idp'
      metadata_uri: 'https://keystone-idp:5000/v3/OS-FEDERATION/
↪saml2/metadata'
      metadata_file: 'metadata-keystone-idp.xml'
      metadata_reload: 1800
```

(continues on next page)

(continued from previous page)

```
      federated_identities:
        - domain: default
          project: fedproject
          group: fedgroup
          role: _member_
      protocols:
        - name: saml2
          mapping:
            name: keystone-idp-mapping
            rules:
              - remote:
                  - type: openstack_user
                local:
                  - group:
                      name: fedgroup
                      domain:
                        name: Default
                    user:
                      name: '{0}'
          attributes:
            - name: openstack_user
              id: openstack_user
            - name: openstack_roles
              id: openstack_roles
            - name: openstack_project
              id: openstack_project
            - name: openstack_user_domain
              id: openstack_user_domain
            - name: openstack_project_domain
              id: openstack_project_domain
```

2.2.4 Service provider configuration for OIDC using mod_auth_openidc

In addition to the generic attributes in keystone_sp defined previously the following attributes can be set in the entry for the `trusted_idp_list` for mod_auth_openidc OIDC-based deployments.

1. `oidc_provider_metadata_url` URL where OpenID Connect Provider metadata can be found.

2. `oidc_client_id` is the Client identifier used in calls to the statically configured OpenID Connect Provider.

3. `oidc_client_secret` is the Client secret used in calls to the statically configured OpenID Connect Provider.

4. `oidc_crypto_passphrase` the crypto passphrase is a password used for encryption of state cookies and cache entries in mod_auth_openidc. Mod_auth_openidcs documentation does not specify any format or restrictions for this password. This should be set to a randomly generated string of a sensible length.

5. `oidc_redirect_path` is the path component of the redirect_uri for this OpenID Connect client; this is a vanity URL that must ONLY point to a path on your server protected by this module but

it must NOT point to any actual content that needs to be served.

6. `oidc_claim_prefix` is the prefix to use when setting claims in the HTTP headers/environment variables. Defaults to OIDC-.
7. `oidc_resp_type` is the response type (or OpenID Connect Flow) used. Defaults to id_token.
8. `oidc_scope` can be used to change the OpenID Connect scope(s) that are requested from the IDP. Defaults to openid email profile.
9. `oidc_auth_verify_jwks_uri` is the URL on which the signing keys for this OP are hosted, in JWK formatting (Optional)
10. `oidc_outgoing_proxy` Specify an outgoing proxy for your network. This is typically used to allow the necessary outgoing requests from keystone to the IDP in non routed environments. (Optional)
11. `oidc_state_max_number_of_cookies` can be used to specify the maximum number of state cookies i.e. the maximum number of parallel outstanding authentication requests. (Optional)
12. `oidc_auth_request_params` can be used to define extra parameters that will be sent along with the Authorization Request. (Optional)
13. `oidc_default_url` can be used to define a default URL to be used in case of 3rd-party-init-SSO when no explicit target_link_uri has been provided. The user is also redirected to this URL in case an invalid authorization response was received. (Optional)
14. `oidc_claim_delimiter` can be used to change the delimiter to use when setting multi-valued claims in the HTTP headers/environment variables. (Optional)

Please refer to the zmartzone/mod_auth_openidc configuration documentation for more information and a full list of possible configuration options.

Below is an example keystone_sp for setting up keystone to be a OIDC-based service provider to an IDP using mod_auth_openidc with CADF notifications on.

```
keystone_sp:
  apache_mod: 'mod_auth_openidc'
  cadf_notifications: true
  cadf_notifications_opt_out:
    - identity.authenticate.failed
    - identity.authenticate.pending
    - identity.authenticate.success
  trusted_dashboard_list:
    - "https://{{ horizon_server_name }}/auth/websso/"
    - "https://{{ external_lb_vip_address }}/auth/websso/"
  trusted_idp_list:
    - name: "oidc-idp"
      oidc_provider_metadata_url: https://identity-provider/.well-
↪known/openid-configuration
      oidc_client_id: keystone
      oidc_client_secret: <SECRET>
      oidc_crypto_passphrase: <RANDOM STRING>
      oidc_redirect_path: /oidc_redirect
      oidc_auth_request_params: param=some+url+encoded+value&
↪param2=and+another+one
```

(continues on next page)

(continued from previous page)

```
      entity_ids:
        - 'https://identity-provider/openid-endpoint/'
      federated_identities:
        - domain: default
          project: fedproject
          group: fedgroup
          role: _member_
      protocols:
        - name: openid
          mapping:
            name: openid-mapping
            rules:
              - remote:
                  - type: OIDC-email
                local:
                  - group:
                      name: fedgroup
                      domain:
                        name: Default
                    user:
                      name: '{0}'
```

It should be noted that mod_auth_openidc is automatically configured to use the existing OSA memcached servers to temporarily persist state data. This is of particular importance for high availability deployments with multiple instances of keystone. The OIDC authentication session state is persisted in memcached to allow different phases of the authentication flow to be handled by different keystone instances due to the round-robin behaviour of the loadbalancer.

2.3 Configure keystone as a federated Identity Provider

The IdP configuration for keystone provides a dictionary attribute with the key `keystone_idp`. The following is a complete example:

```
keystone_idp:
  certfile: "/etc/keystone/ssl/idp_signing_cert.pem"
  keyfile: "/etc/keystone/ssl/idp_signing_key.pem"
  self_signed_cert_subject: "/C=US/ST=Texas/L=San Antonio/O=IT/CN={{ external_
↪lb_vip_address }}"
  regen_cert: false
  idp_entity_id: "{{ keystone_service_publicuri }}/v3/OS-FEDERATION/saml2/idp"
  idp_sso_endpoint: "{{ keystone_service_publicuri }}/v3/OS-FEDERATION/saml2/
↪sso"
  idp_metadata_path: /etc/keystone/saml2_idp_metadata.xml
  service_providers:
    - id: "sp_1"
      auth_url: https://example.com:5000/v3/OS-FEDERATION/identity_providers/
↪idp/protocols/saml2/auth
      sp_url: https://example.com:5000/Shibboleth.sso/SAML2/ECP
```

(continues on next page)

(continued from previous page)

```
organization_name: example_company
organization_display_name: Example Corp.
organization_url: example.com
contact_company: example_company
contact_name: John
contact_surname: Smith
contact_email: jsmith@example.com
contact_telephone: 555-55-5555
contact_type: technical
```

The following list is a reference of allowed settings:

- `certfile` defines the location and filename of the SSL certificate that the IdP uses to sign assertions. This file must be in a location that is accessible to the keystone system user.
- `keyfile` defines the location and filename of the SSL private key that the IdP uses to sign assertions. This file must be in a location that is accessible to the keystone system user.
- `self_signed_cert_subject` is the subject in the SSL signing certificate. The common name of the certificate must match the hostname configuration in the service provider(s) for this IdP.
- `regen_cert` by default is set to `False`. When set to `True`, the next Ansible run replaces the existing signing certificate with a new one. This setting is added as a convenience mechanism to renew a certificate when it is close to its expiration date.
- `idp_entity_id` is the entity ID. The service providers use this as a unique identifier for each IdP. `<keystone-public-endpoint>/OS-FEDERATION/saml2/idp` is the value we recommend for this setting.
- `idp_sso_endpoint` is the single sign-on endpoint for this IdP. `<keystone-public-endpoint>/OS-FEDERATION/saml2/sso>` is the value we recommend for this setting.
- `idp_metadata_path` is the location and filename where the metadata for this IdP is cached. The keystone system user must have access to this location.
- `service_providers` is a list of the known SPs that use the keystone instance as IdP. For each SP, provide three values: `id` as a unique identifier, `auth_url` as the authentication endpoint of the SP, and `sp_url` endpoint for posting SAML2 assertions.
- `organization_name`, `organization_display_name`, `organization_url`, `contact_company`, `contact_name`, `contact_surname`, `contact_email`, `contact_telephone` and `contact_type` are settings that describe the identity provider. These settings are all optional.

2.3.1 Configuring ADFS 3.0 as an identity provider

To install ADFS:

- Prerequisites for ADFS from Microsoft Technet
- ADFS installation procedure from Microsoft Technet

2.3.2 Configuring ADFS

1. Ensure the ADFS server trusts the SPs keystone certificate. We recommend to have the ADFS CA (or a public CA) sign a certificate request for the keystone service.
2. In the ADFS Management Console, choose `Add Relying Party Trust`.
3. Select `Import data about the relying party published online or on a local network` and enter the URL for the SP Metadata (for example, `https://<SP_IP_ADDRESS or DNS_NAME>:5000/Shibboleth.sso/Metadata`)

   Note: ADFS may give a warning message. The message states that ADFS skipped some of the content gathered from metadata because it is not supported by ADFS

4. Continuing the wizard, select `Permit all users to access this relying party`.
5. In the `Add Transform Claim Rule Wizard`, select `Pass Through or Filter an Incoming Claim`.
6. Name the rule (for example, `Pass Through UPN`) and select the `UPN` Incoming claim type.
7. Click *OK* to apply the rule and finalize the setup.

2.4 Configure keystone mappings

The federated_identities functionality can be used to create projects, groups, roles and domains before your federation attribute mappings route users towards those resources. If you manage creation of projects, groups, roles and domains via a separate mechanism making use of federated_identities is not required.

```yaml
federated_identities:
  - domain: default
    project: fedproject
    group: fedgroup
    role: _member_
```

1. `project`: The project that federation users have access to. If the project does not already exist, create it in the domain with the name, `domain`.
2. `group`: The keystone group that federation users belong. If the group does not already exist, create it in the domain with the name, `domain`.
3. `role`: The role that federation users use in that project. Create the role if it does not already exist.
4. `domain`: The domain where the `project` lives, and where the you assign roles. Create the domain if it does not already exist. This should be the ID of the domain.

Ansible implements the equivalent of the following OpenStack CLI commands:

```
# if the domain does not already exist
openstack domain create Default

# if the group does not already exist
openstack group create fedgroup --domain Default

# if the role does not already exist
openstack role create _member_

# if the project does not already exist
openstack project create --domain default fedproject

# map the role to the project and user group in the domain
openstack role add --project fedproject --group fedgroup _member_
```

To extend simply add more entries to the list. For example:

```
federated_identities:
  - domain: default
    project: fedproject
    group: fedgroup
    role: _member_
  - domain: default
    project: fedproject2
    group: fedgroup2
    role: _member_
```

2.4.1 Keystone federation attribute mapping

Attribute mapping adds a set of rules to map federation attributes to keystone users and groups. IdP specifies one mapping per protocol.

Use mapping objects multiple times by different combinations of IdP and protocol.

The details of how the mapping engine works, the schema, and various rule examples are in the keystone developer documentation.

For example, SP attribute mapping configuration for an ADFS IdP:

```
mapping:
  name: adfs-IdP-mapping
  rules:
    - remote:
        - type: upn
      local:
        - group:
            name: fedgroup
            domain:
              name: Default
        - user:
```

(continues on next page)

(continued from previous page)

```
            name: '{0}'
attributes:
  - name: 'http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn'
    id: upn
```

Each IdP for an SP needs to be set up with a mapping. This tells the SP how to interpret the attributes provided to the SP from the IdP.

In this example, the IdP publishes the `upn` attribute. As this is not in the standard Shibboleth attribute map (see `/etc/shibboleth/attribute-map.xml` in the keystone containers), the configuration of the IdP has extra mapping through the `attributes` dictionary.

The `mapping` dictionary is a YAML representation similar to the keystone mapping property which Ansible uploads. The above mapping produces the following in keystone.

```
root@aio1_keystone_container-783aa4c0:~# openstack mapping list
+------------------+
| ID               |
+------------------+
| adfs-IdP-mapping |
+------------------+

root@aio1_keystone_container-783aa4c0:~# openstack mapping show adfs-IdP-
↪mapping
+-------+-----------------------------------------------------------------------
↪-------------------------------------------------------------------+
| Field | Value                                                                ␣
↪                                                                   |
+-------+-----------------------------------------------------------------------
↪-------------------------------------------------------------------+
| id    | adfs-IdP-mapping                                                     ␣
↪                                                                   |
| rules | [{"remote": [{"type": "upn"}], "local": [{"group": {"domain": {"name
↪": "Default"}, "name": "fedgroup"}}, {"user": {"name": "{0}"}}]}] |
+-------+-----------------------------------------------------------------------
↪-------------------------------------------------------------------+

root@aio1_keystone_container-783aa4c0:~# openstack mapping show adfs-IdP-
↪mapping | awk -F\| '/rules/ {print $3}' | python -mjson.tool
[
    {
        "remote": [
            {
                "type": "upn"
            }
        ],
        "local": [
            {
                "group": {
                    "domain": {
```

(continues on next page)

(continued from previous page)

```
                        "name": "Default"
                    },
                    "name": "fedgroup"
                }
            },
            {
                "user": {
                    "name": "{0}"
                }
            }
        ]
    }
]
```

The interpretation of the above mapping rule is that any federation user authenticated by the IdP maps to an `ephemeral` user in keystone. The user is a member of a group named `fedgroup`. This is in a domain called `Default`. As we have specified the domain, the users assignments in the keystone backend will be looked up alongside the assignments made in the mapping. The users ID and Name (federation uses the same value for both properties) for all OpenStack services is the value of `upn`.

To clone or view the source code for this repository, visit the role repository for os_keystone.

CHAPTER

THREE

DEFAULT VARIABLES

```
## Verbosity Options
debug: False

# Set the host which will execute the shade modules
# for the service setup. The host must already have
# clouds.yaml properly configured.
keystone_service_setup_host: "{{ openstack_service_setup_host | default(
↪'localhost') }}"
keystone_service_setup_host_python_interpreter: "{{ openstack_service_setup_
↪host_python_interpreter | default((keystone_service_setup_host == 'localhost
↪') | ternary(ansible_playbook_python, ansible_facts['python']['executable
↪'])) }}"

# Set the package install state for distribution packages
# Options are 'present' and 'latest'
keystone_package_state: "{{ package_state | default('latest') }}"

# Set installation method.
keystone_install_method: "{{ service_install_method | default('source') }}"
keystone_venv_python_executable: "{{ openstack_venv_python_executable |␣
↪default('python3') }}"

# Centos shibboleth repository options
keystone_centos_shibboleth_mirror: "http://download.opensuse.org/repositories/
↪security:/shibboleth/CentOS_7/"
keystone_centos_shibboleth_key: "http://download.opensuse.org/repositories/
↪security:/shibboleth/CentOS_7//repodata/repomd.xml.key"

# Role standard API override this option in the OS variable files
keystone_shibboleth_repo: {}

keystone_git_repo: https://opendev.org/openstack/keystone
keystone_git_install_branch: master
keystone_upper_constraints_url: "{{ requirements_git_url | default('https://
↪releases.openstack.org/constraints/upper/' ~ requirements_git_install_
↪branch | default('master')) }}"
keystone_git_constraints:
  - "--constraint {{ keystone_upper_constraints_url }}"
```

(continues on next page)

(continued from previous page)

```
keystone_pip_install_args: "{{ pip_install_options | default('') }}"

# Name of the virtual env to deploy into
keystone_venv_tag: "{{ venv_tag | default('untagged') }}"
keystone_bin: "{{ _keystone_bin }}"

keystone_fatal_deprecations: False

## System info
keystone_system_user_name: keystone
keystone_system_group_name: keystone
keystone_system_additional_groups:
  - ssl_cert

keystone_system_shell: /bin/bash
keystone_system_comment: keystone system user
keystone_system_user_home: "/var/lib/{{ keystone_system_user_name }}"

## Drivers
keystone_auth_methods: "password,token,application_credential"
keystone_identity_driver: sql
keystone_token_provider: fernet
keystone_token_expiration: 43200
keystone_token_cache_time: 3600

# Set the revocation driver used within keystone.
keystone_revocation_driver: sql
keystone_revocation_cache_time: 3600
keystone_revocation_expiration_buffer: 1800

## Fernet config vars
keystone_fernet_tokens_key_repository: "/etc/keystone/fernet-keys"
keystone_fernet_tokens_max_active_keys: 7
# Any of the following rotation times are valid:
#   reboot, yearly, annually, monthly, weekly, daily, hourly
keystone_fernet_rotation: daily
keystone_fernet_auto_rotation_script: /opt/keystone-fernet-rotate.sh

## Credentials config vars
keystone_credential_key_repository: /etc/keystone/credential-keys
# Any of the following rotation times are valid:
#   reboot, yearly, annually, monthly, weekly, daily, hourly
keystone_credential_rotation: weekly
keystone_credential_auto_rotation_script: /opt/keystone-credential-rotate.sh

keystone_assignment_driver: sql

keystone_resource_cache_time: 3600
```

(continues on next page)

(continued from previous page)

```
keystone_resource_driver: sql

keystone_bind_address: "{{ openstack_service_bind_address | default('0.0.0.0
↪') }}"

## Database info
keystone_db_setup_host: "{{ openstack_db_setup_host | default('localhost') }}"
keystone_db_setup_python_interpreter: "{{ openstack_db_setup_python_
↪interpreter | default((keystone_db_setup_host == 'localhost') |␣
↪ternary(ansible_playbook_python, ansible_facts['python']['executable'])) }}"
keystone_galera_address: "{{ galera_address | default('127.0.0.1') }}"
keystone_galera_user: keystone
keystone_galera_database: keystone
keystone_galera_port: "{{ galera_port | default('3306') }}"
keystone_database_connection_string: >-
  mysql+pymysql://{{ keystone_galera_user }}:{{ keystone_container_mysql_
↪password }}@{{ keystone_galera_address }}:{{keystone_galera_port}}/{{␣
↪keystone_galera_database }}?charset=utf8{% if keystone_galera_use_ssl |␣
↪bool %}&ssl_verify_cert=true{% if keystone_galera_ssl_ca_cert | length > 0
↪%}&ssl_ca={{ keystone_galera_ssl_ca_cert }}{% endif %}{% endif %}
## Database SSL
keystone_galera_use_ssl: "{{ galera_use_ssl | default(False) }}"
keystone_galera_ssl_ca_cert: "{{ galera_ssl_ca_cert | default('') }}"
# Database tuning
keystone_database_enabled: true
keystone_db_max_overflow: "{{ openstack_db_max_overflow | default('50') }}"
keystone_db_max_pool_size: "{{ openstack_db_max_pool_size | default('5') }}"
keystone_db_pool_timeout: "{{ openstack_db_pool_timeout | default('30') }}"
keystone_db_connection_recycle_time: "{{ openstack_db_connection_recycle_time␣
↪| default('600') }}"

## Oslo Messaging
keystone_messaging_enabled: true

# RPC
keystone_oslomsg_rpc_host_group: "{{ oslomsg_rpc_host_group | default(
↪'rabbitmq_all') }}"
keystone_oslomsg_rpc_setup_host: "{{ (keystone_oslomsg_rpc_host_group in␣
↪groups) | ternary(groups[keystone_oslomsg_rpc_host_group][0], 'localhost') }
↪}"
keystone_oslomsg_rpc_transport: "{{ oslomsg_rpc_transport | default('rabbit')␣
↪}}"
keystone_oslomsg_rpc_servers: "{{ oslomsg_rpc_servers | default('127.0.0.1') }
↪}"
keystone_oslomsg_rpc_port: "{{ oslomsg_rpc_port | default('5672') }}"
keystone_oslomsg_rpc_use_ssl: "{{ oslomsg_rpc_use_ssl | default(False) }}"
keystone_oslomsg_rpc_userid: keystone
keystone_oslomsg_rpc_vhost: /keystone
keystone_oslomsg_rpc_ssl_version: "{{ oslomsg_rpc_ssl_version | default(
↪'TLSv1_2') }}"
```

(continues on next page)

(continued from previous page)

```
keystone_oslomsg_rpc_ssl_ca_file: "{{ oslomsg_rpc_ssl_ca_file | default('') }}
↪"

# Notify
keystone_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default(
↪'rabbitmq_all') }}"
keystone_oslomsg_notify_setup_host: "{{ (keystone_oslomsg_notify_host_group␣
↪in groups) | ternary(groups[keystone_oslomsg_notify_host_group][0],
↪'localhost') }}"
keystone_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default(
↪'rabbit') }}"
keystone_oslomsg_notify_servers: "{{ oslomsg_notify_servers | default('127.0.
↪0.1') }}"
keystone_oslomsg_notify_port: "{{ oslomsg_notify_port | default('5672') }}"
keystone_oslomsg_notify_use_ssl: "{{ oslomsg_notify_use_ssl | default(False) }
↪}"
keystone_oslomsg_notify_userid: "{{ keystone_oslomsg_rpc_userid }}"
keystone_oslomsg_notify_password: "{{ keystone_oslomsg_rpc_password }}"
keystone_oslomsg_notify_vhost: "{{ keystone_oslomsg_rpc_vhost }}"
keystone_oslomsg_notify_ssl_version: "{{ oslomsg_notify_ssl_version | default(
↪'TLSv1_2') }}"
keystone_oslomsg_notify_ssl_ca_file: "{{ oslomsg_notify_ssl_ca_file | default(
↪'') }}"

## (Qdrouterd) info
# TODO(ansmith): Change structure when more backends will be supported
keystone_oslomsg_amqp1_enabled: "{{ keystone_oslomsg_rpc_transport == 'amqp' }
↪}"

## Role info
keystone_role_name: admin

## Admin info
keystone_admin_user_name: admin
keystone_admin_tenant_name: admin
keystone_admin_description: Admin Tenant

## Service Type and Data
keystone_service_setup: true
keystone_service_region: "{{ service_region | default('RegionOne') }}"
keystone_service_name: keystone
keystone_service_port: 5000
keystone_service_type: identity
keystone_service_description: "Keystone Identity Service"
keystone_service_tenant_name: service
keystone_service_project_description: "OpenStack Services"

keystone_service_proto: http
keystone_service_publicuri_proto: "{{ openstack_service_publicuri_proto |␣
↪default(keystone_service_proto) }}"
```

(continues on next page)

(continued from previous page)

```
keystone_service_adminuri_proto: "{{ openstack_service_adminuri_proto |␣
↪default(keystone_service_proto) }}"
keystone_service_internaluri_proto: "{{ openstack_service_internaluri_proto |␣
↪default(keystone_service_proto) }}"

keystone_service_internaluri_insecure: false
keystone_service_adminuri_insecure: false

keystone_service_publicuri: "{{ keystone_service_publicuri_proto }}://{{␣
↪external_lb_vip_address }}:{{ keystone_service_port }}"
keystone_service_internaluri: "{{ keystone_service_internaluri_proto }}://{{␣
↪internal_lb_vip_address }}:{{ keystone_service_port }}"
keystone_service_adminuri: "{{ keystone_service_adminuri_proto }}://{{␣
↪internal_lb_vip_address }}:{{ keystone_service_port }}"

## Set this value to override the "public_endpoint" keystone.conf variable
#keystone_public_endpoint: "{{ keystone_service_publicuri }}"

# Apache web server will handle all requests and will act as a
# reverse proxy to uWSGI. If internal TLS/SSL certificates are configured,
# they are implemented in this web server's configuration. Using a web server
# for endpoints is far better for scale and allows the use of additional
# modules to improve performance or security, leaving uWSGI to only have
# to be used for running the service.
#
keystone_web_server_bind_address: "{{ openstack_service_bind_address |␣
↪default('0.0.0.0') }}"

## security.txt
# When security risks in web services are discovered by independent security
# researchers who understand the severity of the risk, they often lack the
# channels to disclose them properly. As a result, security issues may be
# left unreported. security.txt defines a standard to help organizations
# define the process for security researchers to disclose security
# vulnerabilities securely. For more information see https://securitytxt.org/
# This content will be hosted at /security.txt and /.well-known/security.txt
keystone_security_txt_dir: "/var/www/html"
# keystone_security_txt_content: |
#   # Please see https://securitytxt.org/ for details of the specification of␣
↪this file

## Apache setup
keystone_apache_log_level: info
keystone_apache_custom_log_format: combined
keystone_apache_servertokens: "Prod"
keystone_apache_serversignature: "Off"

## Apache MPM tunables
keystone_httpd_mpm_backend: event
```

(continues on next page)

(continued from previous page)

```
keystone_httpd_mpm_server_limit: "{{ keystone_wsgi_processes }}"
keystone_httpd_mpm_start_servers: 2
keystone_httpd_mpm_min_spare_threads: 25
keystone_httpd_mpm_max_spare_threads: 75
keystone_httpd_mpm_thread_limit: 64
keystone_httpd_mpm_thread_child: 25
keystone_httpd_mpm_max_requests: "{{ keystone_httpd_mpm_server_limit | int *␣
↪keystone_httpd_mpm_thread_child | int }}"
keystone_httpd_mpm_max_conn_child: 0

## uWSGI setup
keystone_wsgi_threads: 1
## Cap the maximun number of processes when a user value is unspecified.
keystone_wsgi_processes_max: 16
keystone_wsgi_processes: "{{ [[ansible_facts['processor_vcpus']|default(1),␣
↪1] | max * 2, keystone_wsgi_processes_max] | min }}"
keystone_uwsgi_bind_address: "{{ openstack_service_bind_address | default('0.
↪0.0.0') }}"

keystone_uwsgi_ports:
  keystone-wsgi-public:
    http: 37358
    socket: 35358

keystone_uwsgi_ini_overrides: {}
keystone_default_uwsgi_overrides:
  uwsgi:
    socket: "127.0.0.1:{{ keystone_uwsgi_ports['keystone-wsgi-public']['socket
↪'] }}"

# set keystone_ssl to true to enable SSL configuration on the keystone␣
↪containers
keystone_ssl: "{{ openstack_backend_service_https | default(False) }}"

# The local address used for the keystone node
keystone_node_address: "{{ management_address | default('127.0.0.1') }}"

# Storage location for SSL certificate authority
keystone_pki_dir: "{{ openstack_pki_dir }}"

# Delegated host for operating the certificate authority
keystone_pki_setup_host: "{{ openstack_pki_setup_host | default('localhost') }
↪}"

keystone_pki_keys_path: "{{ keystone_pki_dir ~ '/certs/private/' }}"
keystone_pki_certs_path: "{{ keystone_pki_dir ~ '/certs/certs/' }}"
keystone_pki_intermediate_cert_name: "{{ openstack_pki_service_intermediate_
↪cert_name }}"
keystone_pki_intermediate_cert_path: "{{ keystone_pki_dir ~ '/roots/' ~␣
↪keystone_pki_intermediate_cert_name ~ '/certs/' ~ keystone_pki_intermediate_
↪cert_name ~ '.crt' }}"
```

(continues on next page)

(continued from previous page)

```
keystone_pki_regen_cert: ''

# By default, CA creation is controlled using the CA 'condition' field
keystone_pki_create_ca: True
# An optional private certificate authority for when Keystone is an IDP
keystone_idp_authority_name: "KeystoneIDPAuthority"
keystone_pki_authorities:
  - name: "{{ keystone_idp_authority_name }}"
    country: "GB"
    state_or_province_name: "England"
    organization_name: "Example Corporation"
    organizational_unit_name: "IT Security"
    cn: "Keystone IDP CA"
    provider: selfsigned
    basic_constraints: "CA:TRUE"
    key_usage:
      - digitalSignature
      - keyCertSign
    not_after: "+3650d"
    condition: "{{ (keystone_idp['certfile'] is defined) and _keystone_is_
↪first_play_host }}"

# By default, certificate creation is controlled using the certificates
↪'condition' field
keystone_pki_create_certificates: True
# Server certificate for Apache
keystone_pki_certificates:
  - name: "keystone_{{ ansible_facts['hostname'] }}"
    provider: ownca
    cn: "{{ ansible_facts['hostname'] }}"
    san: "{{ 'DNS:' ~ ansible_facts['hostname'] ~ ',IP:' ~ keystone_node_
↪address }}"
    signed_by: "{{ keystone_pki_intermediate_cert_name }}"
    condition: "{{ keystone_ssl }}"

# Set to the value of keystone_idp_authority_name to regenerate the IDP CA
keystone_pki_regen_ca: ''

# keystone destination files for Apache SSL certificates
keystone_ssl_cert: /etc/ssl/certs/keystone.pem
keystone_ssl_key: /etc/ssl/private/keystone.key
keystone_ssl_ca_cert: /etc/ssl/certs/keystone-ca.pem

# Installation details for SSL certificates
keystone_pki_install_certificates:
  # Apache certificates
  - src: "{{ keystone_user_ssl_cert | default(keystone_pki_certs_path ~
↪'keystone_' ~ ansible_facts['hostname'] ~ '.crt') }}"
    dest: "{{ keystone_ssl_cert }}"
```

(continues on next page)

(continued from previous page)

```
    owner: "{{ keystone_system_user_name }}"
    group: "{{ keystone_system_group_name }}"
    mode: "0644"
    condition: "{{ keystone_ssl }}"
  - src: "{{ keystone_user_ssl_key | default(keystone_pki_keys_path ~
↪'keystone_' ~ ansible_facts['hostname'] ~ '.key.pem') }}"
    dest: "{{ keystone_ssl_key }}"
    owner: "{{ keystone_system_user_name }}"
    group: "{{ keystone_system_group_name }}"
    mode: "0600"
    condition: "{{ keystone_ssl }}"
  - src: "{{ keystone_user_ssl_ca_cert | default(keystone_pki_intermediate_
↪cert_path) }}"
    dest: "{{ keystone_ssl_ca_cert }}"
    owner: "{{ keystone_system_user_name }}"
    group: "{{ keystone_system_group_name }}"
    mode: "0644"
    condition: "{{ keystone_ssl }}"
  # IDP certificates
  - src: "{{ keystone_pki_dir ~ '/roots/' ~ keystone_idp_authority_name ~ '/
↪certs/' ~ keystone_idp_authority_name ~ '.crt' }}"
    dest: "{{ keystone_idp['certfile'] | default('') }}"
    owner: "{{ keystone_system_user_name }}"
    group: "keystone_system_group_name"
    mode: "0640"
    condition: "{{ keystone_idp['certfile'] is defined | bool }}"
  - src: "{{ keystone_pki_dir ~ '/roots/' ~ keystone_idp_authority_name ~ '/
↪private/' ~ keystone_idp_authority_name ~ '.key.pem' }}"
    dest: "{{ keystone_idp['keyfile'] | default('') }}"
    owner: "{{ keystone_system_user_name }}"
    group: "{{ keystone_system_group_name }}"
    mode: "0640"
    condition: "{{ keystone_idp['keyfile'] is defined | bool }}"

keystone_ssl_protocol: "{{ ssl_protocol | default('ALL -SSLv2 -SSLv3 -TLSv1 -
↪TLSv1.1') }}"
# TLS v1.2 and below
keystone_ssl_cipher_suite_tls12: "{{ keystone_ssl_cipher_suite | default(ssl_
↪cipher_suite_tls12 | default(
↪'ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AESGCM:RSA+AES:!
↪aNULL:!MD5:!DSS')) }}"
# TLS v1.3
keystone_ssl_cipher_suite_tls13: "{{ ssl_cipher_suite_tls13 | default('TLS_
↪AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256') }}"

# Set these variables to deploy custom certificates
#keystone_user_ssl_cert: <path to cert on ansible deployment host>
#keystone_user_ssl_key: <path to cert on ansible deployment host>
#keystone_user_ssl_ca_cert: <path to cert on ansible deployment host>
```

(continues on next page)

(continued from previous page)

```
# Set to true when terminating SSL/TLS at a load balancer
keystone_external_ssl: "{{ (haproxy_ssl | default(True)) | bool }}"

# External SSL forwarding proto
keystone_secure_proxy_ssl_header: X-Forwarded-Proto

## Override memcached_servers
keystone_memcached_servers: "{{ memcached_servers }}"

## Caching
# This is a list of strings, each string contains a cache server's
# information (IP:port for example)
# The cache_servers default backend is memcached, so this variable
# should point to a list of memcached servers.
# If empty, caching is disabled.
keystone_cache_servers: "{{ keystone_memcached_servers.split(',') }}"

## LDAP Section
# Define Keystone LDAP domain configuration here.
# This may be used to add configuration for a LDAP identity back-end.
# See the http://docs.openstack.org/admin-guide/identity-integrate-with-ldap.
↪html
#
# Each top-level entry is a domain name. Each entry below that are key: value␣
↪pairs for
# the ldap section in the domain-specific configuration file.
#
# (EXAMPLE LAYOUT)
# keystone_ldap:
#   Users:
#     url: "ldap://127.0.0.1"
#     user: "root"
#     password: "secrete"
#     ...

keystone_ldap: {}
keystone_ldap_domain_config_dir: /etc/keystone/domains



## Policy vars
# Provide a list of access controls to update the default policy.json with.␣
↪These changes will be merged
# with the access controls in the default policy.json. E.g.
#keystone_policy_overrides:
#  identity:create_region: "rule:admin_required"
#  identity:update_region: "rule:admin_required"
```

(continues on next page)

(continued from previous page)

```
## Federation

# Enable the following section on the Keystone IdP
keystone_idp: {}
#keystone_idp:
#  certfile: "/etc/keystone/ssl/idp_signing_cert.pem"
#  keyfile: "/etc/keystone/ssl/idp_signing_key.pem"
#  self_signed_cert_subject: "/C=US/ST=Texas/L=San Antonio/O=IT/CN={{␣
↪external_lb_vip_address }}"
#  regen_cert: false
#  idp_entity_id: "{{ keystone_service_publicuri }}/v3//OS-FEDERATION/saml2/
↪idp"
#  idp_sso_endpoint: "{{ keystone_service_publicuri }}/v3/OS-FEDERATION/saml2/
↪sso"
#  idp_metadata_path: /etc/keystone/saml2_idp_metadata.xml
#  service_providers:
#    - id: "sp_1"
#      auth_url: https://example.com:5000/v3/OS-FEDERATION/identity_providers/
↪idp/protocols/saml2/auth
#      sp_url: https://example.com:5000/Shibboleth.sso/SAML2/ECP
#  # the following settings are optional
#  organization_name: example_company
#  organization_display_name: Example Corp.
#  organization_url: example.com
#  contact_company: example_company
#  contact_name: John
#  contact_surname: Smith
#  contact_email: jsmith@example.com
#  contact_telephone: 555-55-5555
#  contact_type: technical

# Enable the following section in order to install and configure
# Keystone as a Resource Service Provider (SP) and to configure
# trusts with specific Identity Providers (IdP).
keystone_sp: {}
#keystone_sp:
#  cert_duration_years: 5
#  apache_mod: shibboleth  #or mod_auth_openidc
#  cadf_notifications: false
#  cadf_notifications_opt_out:
#    - identity.authenticate.failed
#    - identity.authenticate.pending
#    - identity.authenticate.success
#  trusted_dashboard_list:
#    - "https://{{ external_lb_vip_address }}/auth/websso/"
#    - "https://{{ horizon_server_name }}/auth/websso/"
#  trusted_idp_list:
#    note that only one of these is supported at any one time for now
#    - name: "keystone-idp"
```

(continues on next page)

(continued from previous page)

```
#      domain_id: "default"
#      display_name: "Keystone IDP" # Optional, used in Horizon IDP dropdown
#      entity_ids:
#         - 'https://keystone-idp:5000/v3/OS-FEDERATION/saml2/idp'
#      metadata_uri: 'https://keystone-idp:5000/v3/OS-FEDERATION/saml2/metadata
↪'
#      metadata_file: 'metadata-keystone-idp.xml'
#      metadata_reload: 1800
#      federated_identities:
#        - domain: default
#          project: fedproject
#          group: fedgroup
#          role: _member_
#      protocols:
#        - name: saml2
#          mapping:
#            name: keystone-idp-mapping
#            rules:
#              - remote:
#                  - type: openstack_user
#                local:
#                  - group:
#                      name: fedgroup
#                      domain:
#                        name: Default
#                    user:
#                      name: '{0}'
#          attributes:
#            - name: openstack_user
#              id: openstack_user
#            - name: openstack_roles
#              id: openstack_roles
#            - name: openstack_project
#              id: openstack_project
#            - name: openstack_user_domain
#              id: openstack_user_domain
#            - name: openstack_project_domain
#              id: openstack_project_domain
#
#    - name: 'testshib-idp'
#      entity_ids:
#        - 'https://idp.testshib.org/idp/shibboleth'
#      metadata_uri: 'http://www.testshib.org/metadata/testshib-providers.xml'
#      metadata_file: 'metadata-testshib-idp.xml'
#      metadata_reload: 1800
#      federated_identities:
#        - domain: default
#          project: fedproject
#          group: fedgroup
```

(continues on next page)

(continued from previous page)

```
#           role: _member_
#       protocols:
#         - name: saml2
#           mapping:
#             name: testshib-idp-mapping
#             rules:
#               - remote:
#                   - type: eppn
#                 local:
#                   - group:
#                       name: fedgroup
#                       domain:
#                         name: Default
#                   - user:
#                       name: '{0}'
#
#     - name: 'adfs-idp'
#       entity_ids:
#        - 'http://adfs.contoso.com/adfs/services/trust'
#       metadata_uri: 'https://adfs.contoso.com/FederationMetadata/2007-06/
↪FederationMetadata.xml'
#       metadata_file: 'metadata-adfs-idp.xml'
#       metadata_reload: 1800
#       federated_identities:
#         - domain: default
#           project: fedproject
#           group: fedgroup
#           role: _member_
#       protocols:
#         - name: saml2
#           mapping:
#             name: adfs-idp-mapping
#             rules:
#               - remote:
#                   - type: upn
#                 local:
#                   - group:
#                       name: fedgroup
#                       domain:
#                         name: Default
#                   - user:
#                       name: '{0}'
#           attributes:
#             - name: 'http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn
↪'
#               id: upn
#
#     - name: "keycloak-oidc-idp"
#       oidc_provider_metadata_url: https://identity-provider/.well-known/
↪openid-configuration
```

(continues on next page)

(continued from previous page)

```
#      oidc_client_id: keystone
#      oidc_client_secret: secret
#      oidc_crypto_passphrase: random string
#      oidc_redirect_path: /oidc_redirect
#      oidc_oauth_introspection_endpoint: endpoint address (optional)
#      oidc_oauth_client_id: string (optional)
#      oidc_oauth_client_secret: secret (optional)
#      oidc_pkce_method: plain | S256 | referred_tb (optional)
#      oidc_outgoing_proxy: "proxy address" (optional setting)
#      oidc_auth_request_params: param=some+url+encoded+value&
↪param2=and+another+one (optional)
#      oidc_state_max_number_of_cookies: 5 false (optional)
#      oidc_default_url: https://example.com/callback (optional)
#      entity_ids:
#        - 'https://identity-provider/openid-endpoint/'
#      federated_identities:
#        - domain: default
#          project: fedproject
#          group: fedgroup
#          role: _member_
#      protocols:
#        - name: openid
#          mapping:
#            name: keycloak-oidc-idp-openid-mapping
#            rules:
#              - remote:
#                  - type: OIDC-email
#                local:
#                  - group:
#                      name: fedgroup
#                      domain:
#                        name: Default
#                    user:
#                      name: '{0}'

keystone_service_in_ldap: "{{ service_ldap_backend_enabled | default(False) }}
↪"

# Keystone notification settings
keystone_ceilometer_enabled: "{{ (groups['ceilometer_all'] is defined) and␣
↪(groups['ceilometer_all'] | length > 0) }}"

# Common pip packages
keystone_pip_packages:
  - "git+{{ keystone_git_repo }}@{{ keystone_git_install_branch }}
↪#egg=keystone"
  - ldappool
  - osprofiler
  - PyMySQL
```

(continues on next page)

(continued from previous page)

```
  - pymemcache
  - python-memcached
  - python-openstackclient
  - systemd-python
  - uWSGI
  - pyngus

# Specific pip packages provided by the user
keystone_user_pip_packages: []

# optional pip packages
keystone_optional_oslomsg_amqp1_pip_packages:
  - oslo.messaging[amqp1]

# NOTE(cloudnull): Tunable SSO callback file file-based overrides If defined,
#                  it'll be read from the deployment host, interpreted by the
#                  template engine and copied to the target host.
# keystone_sso_callback_file_path: "/etc/openstack_deploy/keystone/sso_
↪callback_template.html"

#: Tunable file-based overrides
# The contents of these files, if they exist, are read from the
# specified path on the deployment host, interpreted by the
# template engine and copied to the target host. If they do
# not exist then they will be generated on first playbook run.
shibboleth_cert_user_file_path: "/etc/openstack_deploy/keystone/sp-cert.pem"
shibboleth_key_user_file_path: "/etc/openstack_deploy/keystone/sp-key.pem"

#: Tunable var-based overrides
# The contents of these are templated over the default files.
keystone_keystone_conf_overrides: {}
keystone_keystone_default_conf_overrides: {}
keystone_policy_overrides: {}

keystone_required_secrets:
  - keystone_auth_admin_password
  - keystone_container_mysql_password
  - keystone_oslomsg_rpc_password
  - keystone_oslomsg_notify_password
  - keystone_rabbitmq_password

keystone_uwsgi_init_overrides: {}

## Service Name-Group Mapping
keystone_services:
  keystone-wsgi-public:
    group: keystone_all
    wsgi_app: True
    wsgi_path: "{{ keystone_bin }}/keystone-wsgi-public"
```

(continues on next page)

(continued from previous page)

```
    uwsgi_overrides: "{{ keystone_default_uwsgi_overrides | combine(keystone_
↪uwsgi_ini_overrides, recursive=True) }}"
    uwsgi_bind_address: "{{ keystone_uwsgi_bind_address }}"
    uwsgi_port: "{{ keystone_uwsgi_ports['keystone-wsgi-public']['http'] }}"

## Extra HTTP headers for Keystone
# Add any additional headers here that Keystone should return.
#
# Example:
#
#   keystone_extra_headers:
#     - parameter: "Access-Control-Expose-Headers"
#       value: "X-Subject-Token"
#     - parameter: "Access-Control-Allow-Headers"
#       value: "Content-Type, X-Auth-Token"
#     - parameter: "Access-Control-Allow-Origin"
#       value: "*"
keystone_extra_headers: []

# List of trusted IPs which can pass X-Forwarded-For
keystone_set_real_ip_from: []

# Toggle whether memcache should be flushed when doing
# database migrations. This is sometimes useful when
# doing upgrades, but should not usually be required.
# ref: https://bugs.launchpad.net/openstack-ansible/+bug/1793389
keystone_flush_memcache: no

# host which holds the ssh certificate authority
keystone_ssh_keypairs_setup_host: "{{ openstack_ssh_keypairs_setup_host |␣
↪default('localhost') }}"

# directory on the deploy host to create and store SSH keypairs
keystone_ssh_keypairs_dir: "{{ openstack_ssh_keypairs_dir | default('/etc/
↪openstack_deploy/ssh_keypairs') }}"

#Each keystone host needs a signed ssh certificate to log into the others
keystone_ssh_keypairs:
  - name: "keystone-{{ inventory_hostname }}"
    cert:
      signed_by: "{{ openstack_ssh_signing_key }}"
      principals: "{{ keystone_ssh_key_principals | default('keystone') }}"
      valid_from: "{{ keystone_ssh_key_valid_from | default('always') }}"
      valid_to: "{{ keystone_ssh_key_valid_to | default('forever') }}"

#Each keystone host needs the signed ssh certificate installing to the␣
↪keystone user
keystone_ssh_keypairs_install_keys:
  owner: "{{ keystone_system_user_name }}"
```

(continues on next page)

(continued from previous page)

```
  group: "{{ keystone_system_group_name }}"
  keys:
    - cert: "keystone-{{ inventory_hostname }}"
      dest: "{{ keystone_system_user_home }}/.ssh/id_rsa"

#Each compute host must trust the SSHD certificate authoritiy in the sshd␣
↪configuration
keystone_ssh_keypairs_install_ca: "{{ openstack_ssh_keypairs_authorities }}"

#Each compute host must allow SSH certificates with the appropriate principal␣
↪to log into the keystone user
keystone_ssh_keypairs_principals:
  - user: "{{ keystone_system_user_name }}"
    principals: "{{ keystone_ssh_key_principals | default(['keystone']) }}"
```

CHAPTER

FOUR

DEPENDENCIES

This role needs pip >= 7.1 installed on the target host.

To use this role, define the following variables:

```
# hostname or IP of load balancer providing external network
# access to Keystone
external_lb_vip_address: 10.100.100.102

# hostname or IP of load balancer providing internal network
# access to Keystone
internal_lb_vip_address: 10.100.100.102

# password used by the keystone service to interact with Galera
keystone_container_mysql_password: "YourPassword"

keystone_auth_admin_password: "SuperSecretePassword"
keystone_rabbitmq_password: "secrete"
keystone_container_mysql_password: "SuperSecrete"
```

This list is not exhaustive at present. See role internals for further details.

CHAPTER

FIVE

EXAMPLE PLAYBOOK

```yaml
---
- name: Installation and setup of Keystone
  hosts: keystone_all
  user: root
  roles:
    - { role: "os_keystone", tags: [ "os-keystone" ] }
  vars:
    external_lb_vip_address: 10.100.100.102
    internal_lb_vip_address: 10.100.100.102
    keystone_galera_address: 10.100.100.101
    keystone_galera_database: keystone
    keystone_venv_tag: "testing"
    keystone_developer_mode: true
    keystone_git_install_branch: master
    keystone_auth_admin_password: "SuperSecretePassword"
    keystone_oslomsg_rpc_password: "secrete"
    keystone_oslomsg_notify_password: "secrete"
    keystone_container_mysql_password: "SuperSecrete"
    keystone_oslomsg_rpc_transport: rabbit
    keystone_oslomsg_rpc_servers: 10.100.100.101
    keystone_oslomsg_rpc_port: 5671
    keystone_oslomsg_rpc_use_ssl: true
    keystone_oslomsg_rpc_userid: keystone
    keystone_oslomsg_rpc_vhost: /keystone
    keystone_oslomsg_notify_transport: rabbit
    keystone_oslomsg_notify_servers: 10.100.100.101
    keystone_oslomsg_notify_port: 5671
    keystone_oslomsg_notify_use_ssl: true
    keystone_oslomsg_notify_userid: keystone
    keystone_oslomsg_notify_vhost: /keystone
    galera_client_drop_config_file: false
    galera_root_user: root
  vars_prompt:
    - name: "galera_root_password"
      prompt: "What is galera_root_password?"
```

CHAPTER SIX

TAGS

This role supports two tags: `keystone-install` and `keystone-config`

The `keystone-install` tag can be used to install and upgrade.

The `keystone-config` tag can be used to maintain configuration of the service.