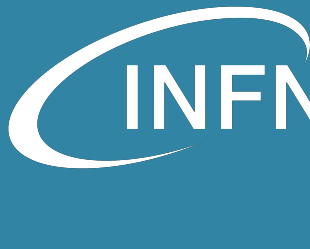
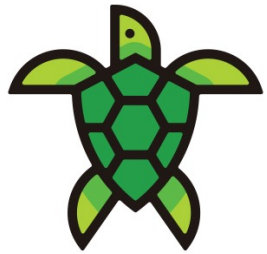


Keystone



«OpenStack Administration 101» , 30 Nov. – 3 Dec. 2021

Doina Cristina Duma & Stefano Stalio

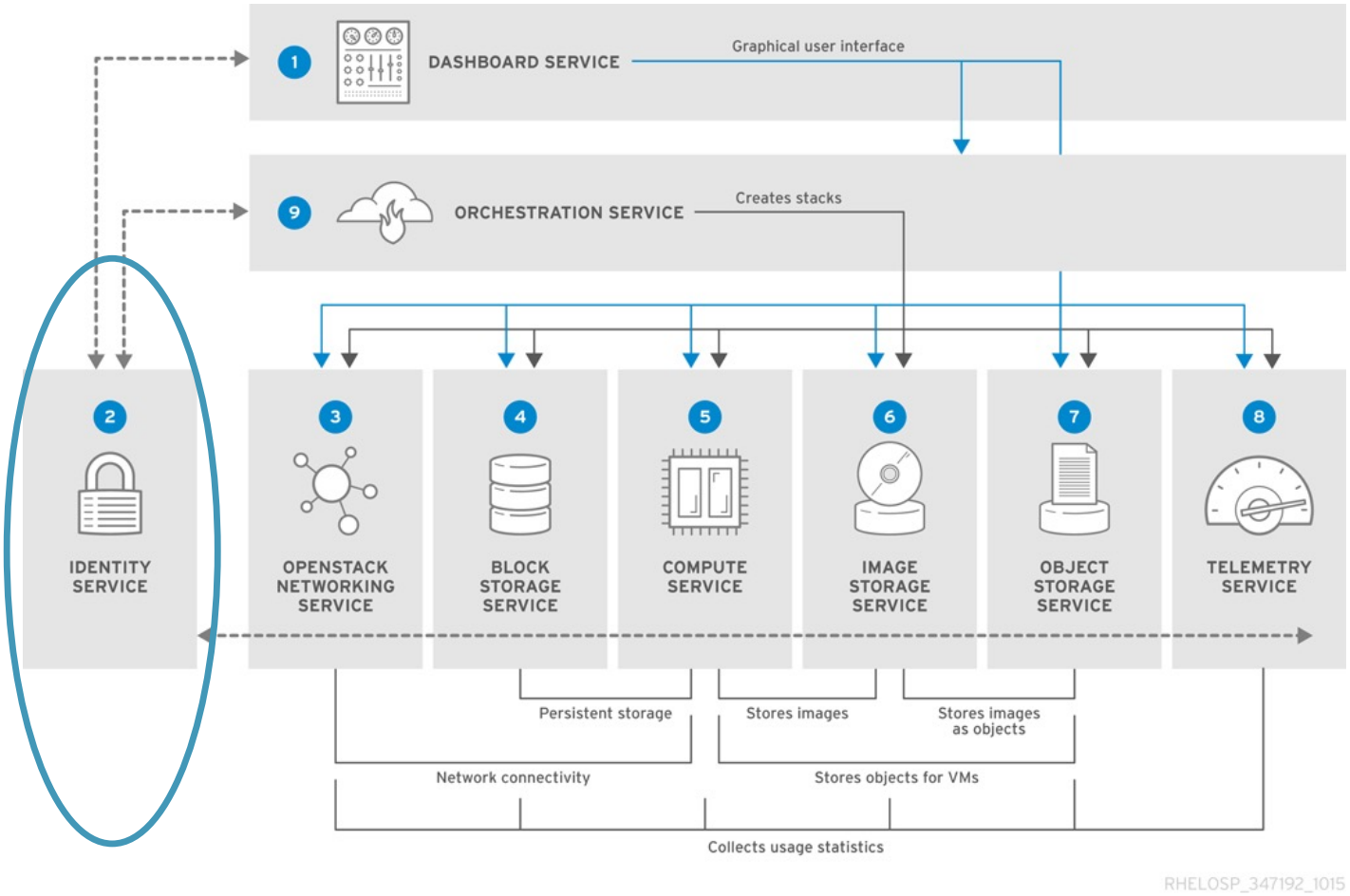
30/11/2021

Overview

- What is the Openstack Identity Service
- Keystone Services
- Understand Keystone Configurations Commands
- Services and their endpoints
- (Hints) on installation and configuration
- Working with OpenStack Project, users, Domains and Groups (hands-on)

INFN

High-level Overview of Core Services



What is the OpenStack Identity Service?

The OpenStack Identity Service, also known as Keystone, provides – a set of services developed for the purposes of user authentication and authorization by OpenStack cloud. The service allows to securely check the client's identity and assign a unique access code (token) trusted by internal services.

- The OpenStack Identity service provides **a single point of integration** for managing authentication, authorization, and a catalog of services.
- It is the first service a user interacts with.
  - Once authenticated, an end user can use its identity to access other OpenStack services
- Other OpenStack services leverage the Identity service to **ensure users are who they say they are** and **discover where other services are** within the deployment

Overview of Keystone Capabilities (1)

Identity

- Identity refers to the identification of who is trying to access cloud resources.
  - In OpenStack Keystone, identity is typically represented as a user.
  - In simple deployments, the identity of a user can be stored in Keystone's own database.
  - In production or enterprise environments, an external Identity Provider is commonly used.
- Authentication
  - Authentication is the process of validating a user's identity.
  - In many cases, authentication is initially performed by a user performing a login with their user identity and a password. Keystone is capable of performing all the authentication steps itself.
  - Keystone is pluggable such that it easily integrates with an existing hardened production authentication service, such as LDAP or Active Directory.
  - While a user identity is **typically initially authenticated with a password**, it is very common as part of this initial authentication to **create a token** for subsequent authentications.
    - Tokens also have a limited lifespan and expire so that their usefulness is limited if they are stolen.
    - OpenStack relies heavily on tokens for authentication and other purposes — and **Keystone is the one and only OpenStack service that can issue them.**
    - Currently, Keystone uses a form of token called a **bearer token**. =>whomever has obtained ownership of the token, is capable of using it to authenticate and access resources.

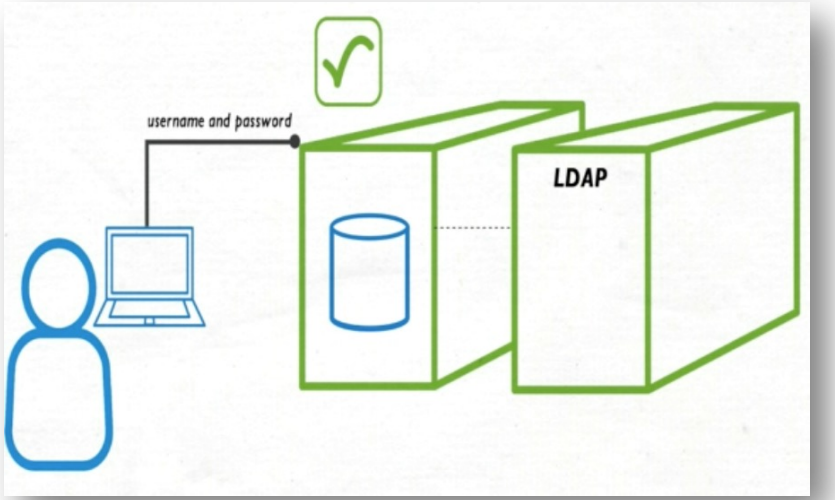
Overview of Keystone Capabilities (2)

Access Management (Authorization)

- Once a user identity has been authenticated and a token has been created and allocated – it starts the Access Management.
- Access Management, also referred to as Authorization, is the process of determining what resources a user is permitted to access.
  - Cloud environments such as OpenStack provide users with access to large amounts of resources. => there needs to be a mechanism for determining which users are allowed to create new instances of a particular virtual machine, etc
  - Keystone maps Users to Projects or Domains by associating a Role for the User for that Project or Domain.
    - Other OpenStack projects (Nova, Cinder, and Neutron) examine the User's Project and Role associations and evaluate this information using a policy engine. => examines this information (the Role value) and makes a determination about what actions the user is allowed to perform.

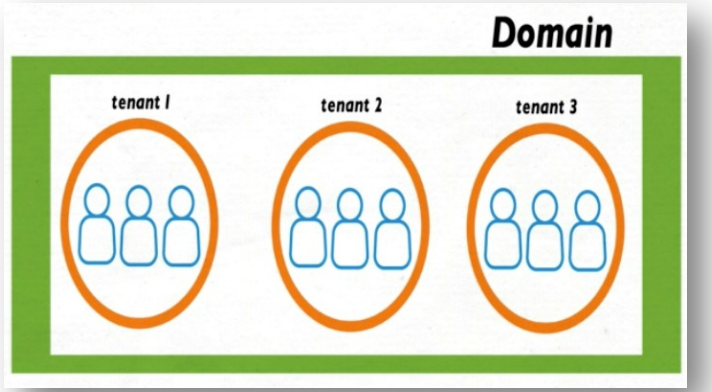
Keystone Services (1)

- Authentication (or Identity) Service:
  - Allows for credential validation of username and password pairs when cloud users log into the cloud environment.
  - Keystone can use a Database to store these username/password pairs or it can use an external source such as an LDAP server.
  - Concepts:
    - Users - represent an individual API consumer
      - A **digital representation** of a person, system, or service that uses OS services
      - A user itself must be owned by a **specific domain**
    - Groups
      - container representing **a collection of users**.
      - A group itself must be owned by a **specific domain**



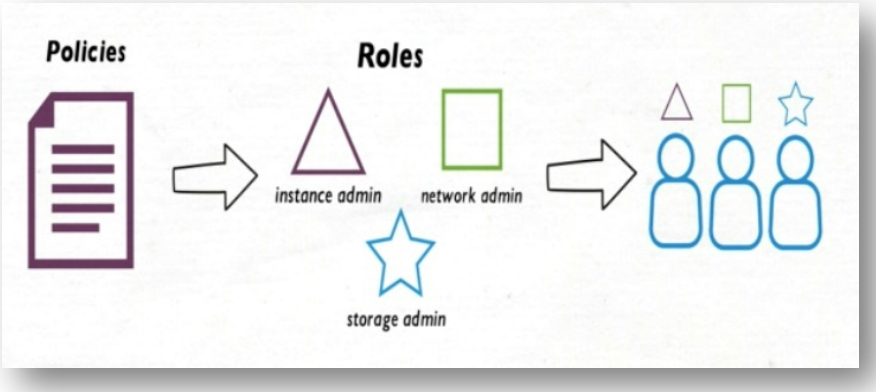
Keystone Services (2)

- Resource Service:
  - Resource Service manages all of the **data relative to projects/tenants and domains.**
  - Concepts:
    - Projects
      - base unit of ownership in OpenStack, in that all resources in OpenStack should be owned by a specific project
      - A project itself must be owned by a **specific domain**
    - Domains
      - high-level container for projects, users and groups
      - Keystone provides a default domain, aptly named '*Default*'.
      - Define **administrative boundaries for managing Identity entities**.
      - Users can be granted the administrator role for a domain. A domain administrator can create projects, users, and groups in a domain and assign roles to users and groups in a domain.
    - Projects are the entities that are granted access to the cloud resources and Domains are entities that are used to manage groups of projects and users.



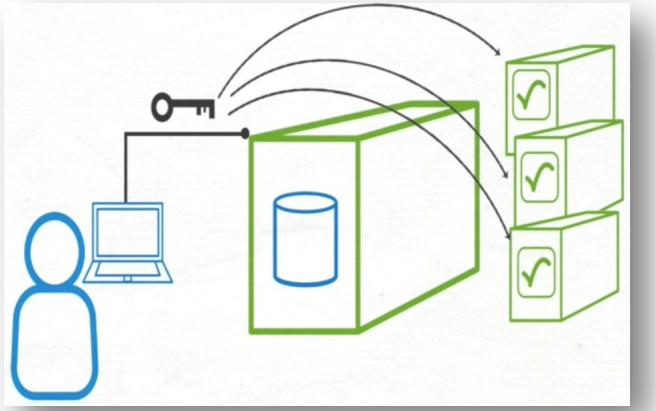
Keystone Services (3)

- Assignment Service:
  - manages all of the data relating to roles and the role assignments.
  - Roles are assigned to users and are what grant or restrict access to specific cloud resources.
  - Concepts:
    - Roles
      - Dictate the level of authorization the end user can obtain.
      - Roles can be granted at either the domain or project level.
      - A role can be assigned at the individual user or group level. Role names are unique within the owning domain.
    - Role Assignments
      - A 3-tuple that has a Role, a Resource and an Identity
  -



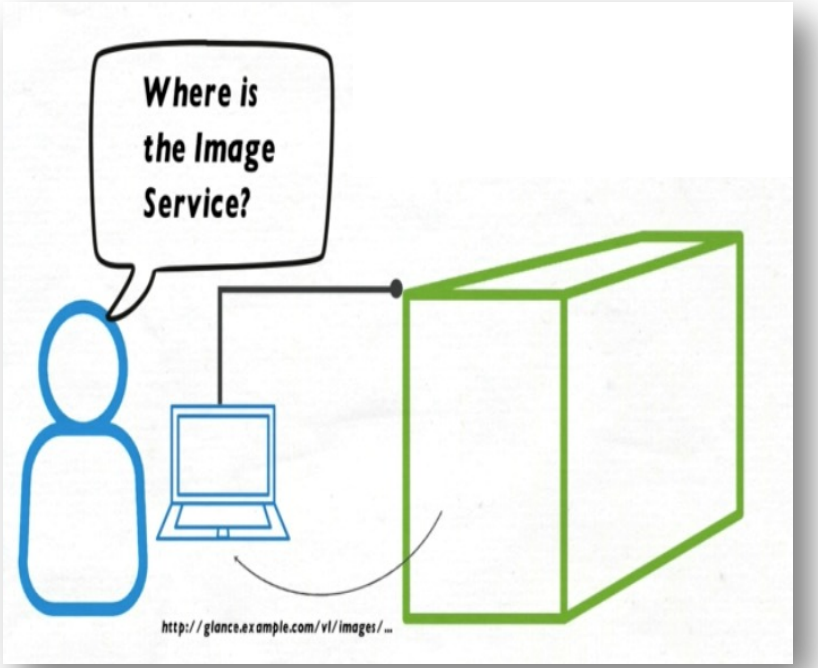
Keystone Services (4)

- The Authorization (or Token) Service:
  - Works hand in hand with the Authentication Service by creating **authentication tokens** for authenticated users and services that allow them to gain access to other OpenStack services.
  - *Concepts*:
    - Token
      - An **alpha-numeric text string** that enables access to OpenStack APIs and resources.
      - A token may be **revoked at any time** and is **valid for a finite duration**. While OpenStack Identity supports token-based authentication in this release, it intends to support additional protocols in the future

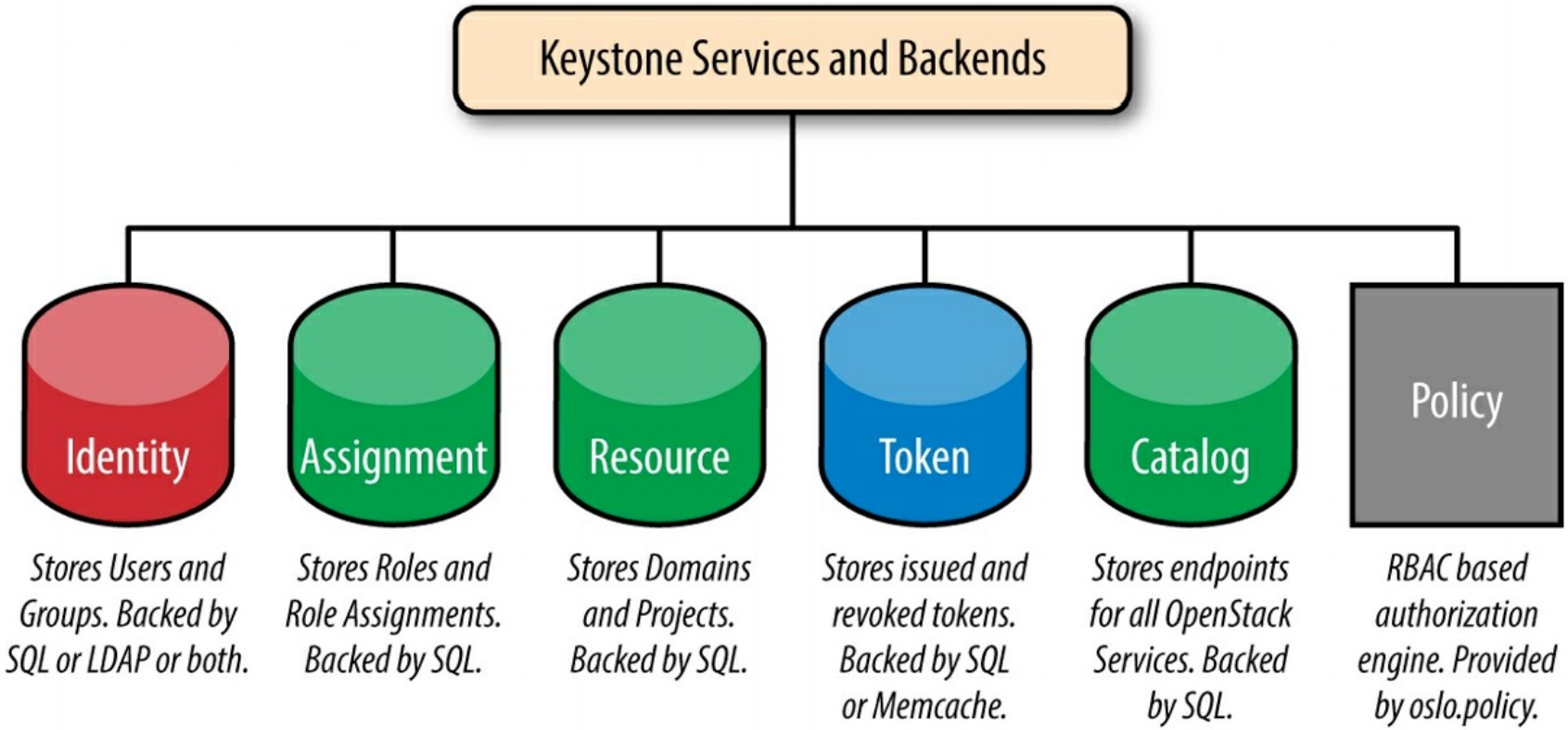


Keystone Services (2)

- Service Catalog.: Endpoint registry & discovery
  - It provides a central index of cloud services and their endpoints.
  - This can **simplify the configuration** of the other cloud services because instead of having to configure each cloud service manually to know how to communicate with every other cloud service, you can just **configure the service to use the Service Catalog to look up the endpoint of the service** that they need to communicate with.
  - Concepts:
    - Service
      - An **OS service** that provides endpoints through which users can access resources and perform operations
    - Endpoint
      - A **network-accessible address**, (URL), through which you can access a service.



Keystone Services and their backends



More on Service Catalogue

- Users and services can locate other services by using the service catalog, which is managed by the Identity service.
  - a service catalog is **a collection of available services in an OpenStack deployment**.
  - Each service can have **one or many endpoints** and each endpoint can be one of three types: admin, internal, or public.
  - In a production environment, different endpoint types might reside on separate networks exposed to different types of users for security reasons.
    - The public API network might be visible from the Internet so customers can manage their clouds.
    - The admin API network might be restricted to operators within the organization that manages cloud infrastructure.
    - The internal API network might be restricted to the hosts that contain OpenStack services.
  - Also, OpenStack supports multiple regions for scalability.
    - Together, regions, services, and endpoints created within the Identity service comprise the service catalog for a deployment.
    - Each OpenStack service needs a service entry with corresponding endpoints stored in the Identity service. This can all be done after the Identity service has been installed and configured.

Service Catalog Management Commands

- Syntax: `openstack service` ***MODE OPTIONS***

Mode	Description
`create`	-define a new service
`delete`	-delete an existing service
`list`	-display existing services

- Syntax: `openstack endpoint` ***MODE OPTIONS***

Mode	Description
`create`	-define a new endpoint for a service
`delete`	-delete an existing service endpoint
`list`	-display existing configured service endpoints

Services and their endpoints

```
[root@oa101-dcd-ctrl ~(keystone_admin)]# openstack service list
+----------------------------------+-----------+-----------------+
| ID                               | Name      | Type            |
+----------------------------------+-----------+-----------------+
| 0874f00a5d8a4b8d834d0794934749b9 | magnum    | container-infra |
| 1d0fb6045bf54e698f085db849d1f7f8 | nova      | compute         |
| 5cc118d01f0042cd9b860770dcd9c10f | cinderv3  | volumev3        |
| 63bad06eb95a49f288901ca61a272374 | placement | placement       |
| 7785bfa25f814661b1b095e71cd9dec0 | neutron   | network         |
| a3a3dc29a47a44ffaa97ad739c824591 | heat      | orchestration   |
| bb00149287d1426aa89306535d967cfe | glance    | image           |
| bcbf689138684a19943ee167a26e5756 | swift     | object-store    |
| c1f01d27248a4c2b92accd4f9039678f | heat-cfn  | cloudformation  |
| c81fbfa9b5d648ac86edf357a91339a3 | keystone  | identity        |
+----------------------------------+-----------+-----------------+
```

Services and their endpoints

Services

Service endpoints

```
[root@oa101-dcd-ctrl ~(keystone_admin)]# openstack endpoint list --sort-column "Service Name"
+----------------------------------+-----------+--------------+----------------+---------+-----------+-----------------------------------------------+
| ID                               | Region    | Service Name | Service Type   | Enabled | Interface | URL                                           |
+----------------------------------+-----------+--------------+----------------+---------+-----------+-----------------------------------------------+
| 361e5861027541f39ad5afceff48eb97 | RegionOne | cinderv3     | volumev3       | True    | admin     | http://10.10.0.5:8776/v3/%(tenant_id)s        |
| 50900871057f4f9bb9464608c7341234 | RegionOne | cinderv3     | volumev3       | True    | internal  | http://10.10.0.5:8776/v3/%(tenant_id)s        |
| 9f98bf387474430f87bfb3fd0b8363fd | RegionOne | cinderv3     | volumev3       | True    | public    | http://10.10.0.5:8776/v3/%(tenant_id)s        |
| 590ff98f728444ab90c9a78057d8341b | RegionOne | glance       | image          | True    | admin     | http://10.10.0.5:9292                         |
| 5abc51033def4d0388cffb284b4a8a2e | RegionOne | glance       | image          | True    | public    | http://10.10.0.5:9292                         |
| bd704e0494cd470f9c822905539fc431 | RegionOne | glance       | image          | True    | internal  | http://10.10.0.5:9292                         |
| 2c32c9727c2245b0a47c420ec7bc4633 | RegionOne | heat         | orchestration  | True    | public    | http://10.10.0.5:8004/v1/%(tenant_id)s        |
| 77e8b458d3434de1be01edef3e890587 | RegionOne | heat         | orchestration  | True    | admin     | http://10.10.0.5:8004/v1/%(tenant_id)s        |
| e9a6a2b0935e4f349c4f0194ab8781ef | RegionOne | heat         | orchestration  | True    | internal  | http://10.10.0.5:8004/v1/%(tenant_id)s        |
| 59f0c289a6b44287bd1893ccba035aaa | RegionOne | heat-cfn     | cloudformation | True    | admin     | http://10.10.0.5:8000/v1                      |
| 69f726492caf44df867655d027a1403c | RegionOne | heat-cfn     | cloudformation | True    | internal  | http://10.10.0.5:8000/v1                      |
| 70061f6508a54c47b78d53c469207137 | RegionOne | heat-cfn     | cloudformation | True    | public    | http://10.10.0.5:8000/v1                      |
| 27f60d4787a345d2bbec2fd8308d1468 | RegionOne | keystone     | identity       | True    | public    | http://10.10.0.5:5000                         |
| 563ee1bc533f42e3b4ce167e8e95339b | RegionOne | keystone     | identity       | True    | admin     | http://10.10.0.5:5000                         |
| 80f1e60140894176b625fb83b76bd350 | RegionOne | keystone     | identity       | True    | internal  | http://10.10.0.5:5000                         |
| 6916f78056cc4e8c88fe6603508ebfcd | RegionOne | magnum       | container-infra | True   | public    | http://10.10.0.5:9511/v1                      |
| 77182eec57d040c8b9ce8d97402b2b3e | RegionOne | magnum       | container-infra | True   | internal  | http://10.10.0.5:9511/v1                      |
| 7ec25bc81cb546c48403cabec14753ec | RegionOne | magnum       | container-infra | True   | admin     | http://10.10.0.5:9511/v1                      |
| 24e87c17c0e6414cae310a9be98d646e | RegionOne | neutron      | network        | True    | internal  | http://10.10.0.5:9696                         |
| 7c90d04a3d8844afbc86d06ac69d09c8 | RegionOne | neutron      | network        | True    | public    | http://10.10.0.5:9696                         |
| d12b8017901b45d989645b29ecd5de1f | RegionOne | neutron      | network        | True    | admin     | http://10.10.0.5:9696                         |
| a7281371b4074365aa20abeb4e56554c | RegionOne | nova         | compute        | True    | internal  | http://10.10.0.5:8774/v2.1/%(tenant_id)s      |
| b2caebb966d34c22a8a2e6ed456c8bed | RegionOne | nova         | compute        | True    | public    | http://10.10.0.5:8774/v2.1/%(tenant_id)s      |
| f805bcfabeda473f97862aa6728e8ce8 | RegionOne | nova         | compute        | True    | admin     | http://10.10.0.5:8774/v2.1/%(tenant_id)s      |
| 0a0eec5f71564c0d8a0486b10bbe4b51 | RegionOne | placement    | placement      | True    | admin     | http://10.10.0.5:8778/placement               |
| 75d9070192d44f79ae7bfb3b80fb0443 | RegionOne | placement    | placement      | True    | public    | http://10.10.0.5:8778/placement               |
| b384323fce9f4e588ac5ad9587d58722 | RegionOne | placement    | placement      | True    | internal  | http://10.10.0.5:8778/placement               |
| 3160d12431c1494ba0956aeae7272ede | RegionOne | swift        | object-store   | True    | admin     | http://10.10.0.5:8080/v1/AUTH_%(tenant_id)s   |
| 340de3981cab4b7dac3251142f26f9ea | RegionOne | swift        | object-store   | True    | internal  | http://10.10.0.5:8080/v1/AUTH_%(tenant_id)s   |
| 4a0f1c131b9f4e1db0d31e2903e67194 | RegionOne | swift        | object-store   | True    | public    | http://10.10.0.5:8080/v1/AUTH_%(tenant_id)s   |
+----------------------------------+-----------+--------------+----------------+---------+-----------+-----------------------------------------------+
```

Services and their endpoints

Services

Service endpoints

```
openstack endpoint list --sort-column "Service Name" --sort-column Region
```

ID	Region	Service Name	Service Type	Enabled	Interface	URL
387265b392ab42a9b5974d1c7f1b37de	sdds	cinderv2	volumev2	True	internal	https://cloud-api-int.cloud.cnaf.infn.it:8776/v2/%(project_id)s
89a736a8e8fe4c9d81aec57236ef8c3c	sdds	cinderv2	volumev2	True	admin	https://cloud-api-int.cloud.cnaf.infn.it:8776/v2/%(project_id)s
c11659e837ee46d795abfb8fc94d5c4e	sdds	cinderv2	volumev2	True	public	https://cloud-api-pub.cloud.cnaf.infn.it:8776/v2/%(project_id)s
5fd3b28cbf7949899f29ccbdaffbf550	tier1	cinderv2	volumev2	True	internal	https://cloud-api-int.cr.cnaf.infn.it:8776/v2/%(project_id)s
7123b07f427d4fde908292453d6032b7	tier1	cinderv2	volumev2	True	public	https://cloud-api-pub.cr.cnaf.infn.it:8776/v2/%(project_id)s
be6bc6c423ab43cf9f25aa4a75879256	tier1	cinderv2	volumev2	True	admin	https://cloud-api-int.cr.cnaf.infn.it:8776/v2/%(project_id)s
4eb72789396d40e895788ef657917451	sdds	cinderv3	volumev3	True	public	https://cloud-api-pub.cloud.cnaf.infn.it:8776/v3/%(project_id)s
55434d471f414ac485f4e4faafd4c969	sdds	cinderv3	volumev3	True	internal	https://cloud-api-int.cloud.cnaf.infn.it:8776/v3/%(project_id)s
d563d3f827224e3b8e78192c3c62551e	sdds	cinderv3	volumev3	True	admin	https://cloud-api-int.cloud.cnaf.infn.it:8776/v3/%(project_id)s
3dd214757415472f84eaac06bac73c54	tier1	cinderv3	volumev3	True	public	https://cloud-api-pub.cr.cnaf.infn.it:8776/v3/%(project_id)s
c5a1c095a6034d07aa318f5089aec8f7	tier1	cinderv3	volumev3	True	internal	https://cloud-api-int.cr.cnaf.infn.it:8776/v3/%(project_id)s
d7c74058e3504c868c004a290b0a3591	tier1	cinderv3	volumev3	True	admin	https://cloud-api-int.cr.cnaf.infn.it:8776/v3/%(project_id)s
3f5a19aa76fc4f46b6488297d134373f	sdds	glance	image	True	admin	https://cloud-api-int.cr.cnaf.infn.it:9292
b1f2c2bf8c72446494dd321c8eab4ff5	sdds	glance	image	True	internal	https://cloud-api-int.cr.cnaf.infn.it:9292
ff9e09cc35474e4da705197af81344dc	sdds	glance	image	True	public	https://cloud-api-pub.cr.cnaf.infn.it:9292
0c77a5327dbb463e9dab8ad2f9676284	tier1	glance	image	True	public	https://cloud-api-pub.cr.cnaf.infn.it:9292
375c096e2ed94a91b79ea1567d2b7946	tier1	glance	image	True	internal	https://cloud-api-int.cr.cnaf.infn.it:9292
3c5ed313ce7e41c29ad2e8643744f0b4	tier1	glance	image	True	admin	https://cloud-api-int.cr.cnaf.infn.it:9292
25da83beb3d440799441b506441eeb18	sdds	heat	orchestration	True	internal	https://cloud-api-int.cloud.cnaf.infn.it:8004/v1/%(tenant_id)s
2aebbaa50e5448d095925070ada219ba	sdds	heat	orchestration	True	public	https://cloud-api-pub.cloud.cnaf.infn.it:8004/v1/%(tenant_id)s
68da3dd53d3a4272b0eb52f26565580a	sdds	heat	orchestration	True	admin	https://ccloud-api-int.cloud.cnaf.infn.it:8004/v1/%(tenant_id)s
11b53b4ea16340b78b01ef2cd918692e	tier1	heat	orchestration	True	public	https://cloud-api-pub.cr.cnaf.infn.it:8004/v1/%(tenant_id)s
a458aadea13145b08f58d9c5300f5302	tier1	heat	orchestration	True	internal	https://cloud-api-int.cr.cnaf.infn.it:8004/v1/%(tenant_id)s
b844eee0cc2843a1894c415280dee765	tier1	heat	orchestration	True	admin	https://cloud-api-int.cr.cnaf.infn.it:8004/v1/%(tenant_id)s
2cec71e581d34f1193a84f20cc300a37	sdds	heat-cfn	cloudformation	True	internal	https://cloud-api-int.cloud.cnaf.infn.it:8000/v1
bca1866dc68a4e67a83717b52a5d927d	sdds	heat-cfn	cloudformation	True	admin	https://cloud-api-int.cloud.cnaf.infn.it:8000/v1
e4c55cfbb31b4a6cb5820270e9c3c19b	sdds	heat-cfn	cloudformation	True	public	https://cloud-api-pub.cloud.cnaf.infn.it:8000/v1
402315165608464aa989f457a7d801b7	tier1	heat-cfn	cloudformation	True	internal	https://cloud-api-int.cr.cnaf.infn.it:8000/v1
7f8f15b7705843b09d702ea4e503e639	tier1	heat-cfn	cloudformation	True	admin	https://cloud-api-int.cr.cnaf.infn.it:8000/v1
afe863a19ee1469f8a06651a5d35ae95	tier1	heat-cfn	cloudformation	True	public	https://cloud-api-pub.cr.cnaf.infn.it:8000/v1
5d47af1b9b7f49b18a898e197d337593	sdds	keystone	identity	True	admin	https://cloud-api-int.cr.cnaf.infn.it:5000/v3/
87de7a060b114b73a3055cc0da8cbc8c	sdds	keystone	identity	True	internal	https://cloud-api-int.cr.cnaf.infn.it:5000/v3/
d4d4e92dd4944ad68f9aa1663846e8b6	sdds	keystone	identity	True	public	https://cloud-api-pub.cr.cnaf.infn.it:5000/v3/
2138f3e119514048923747d3f7817e90	tier1	keystone	identity	True	internal	https://cloud-api-int.cr.cnaf.infn.it:5000/v3/
29bcee65da534c38b10a799bf3078590	tier1	keystone	identity	True	admin	https://cloud-api-int.cr.cnaf.infn.it:5000/v3/
88a83a09a22849bf8f6599d992daaf6c	tier1	keystone	identity	True	public	https://cloud-api-pub.cr.cnaf.infn.it:5000/v3/

Keystone components

- The Identity service contains these components:
  - Server - a centralized server provides authentication and authorization services using a RESTful interface.
  - Drivers - or a service backends are integrated to the centralized server.
    - used for accessing identity information in repositories external to OpenStack, and may already exist in the infrastructure where OpenStack is deployed (for example, SQL databases or LDAP servers).
  - Modules - middleware modules run in the address space of the OpenStack component that is using the Identity service.
    - intercept service requests, extract user credentials, and send them to the centralized server for authorization. The integration between the middleware modules and OpenStack components uses the Python Web Server Gateway Interface.

Install & Config (hints) - Prerequisits

Before you install and configure the Identity service, you must create a database.

1. Use the database access client to connect to the database server as the **root** user:

```
$ mysql -u root -p
```

2. Create the **keystone** database:

```
MariaDB [(none)]> CREATE DATABASE keystone;
```

3. Grant proper access to the **keystone** database:

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' \
IDENTIFIED BY 'KEYSTONE_DBPASS';
MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' \
IDENTIFIED BY 'KEYSTONE_DBPASS';
```

   Replace **KEYSTONE_DBPASS** with a suitable password.

4. Exit the database access client.

Install & Config (hints)

1. Run the following command to install the packages:

```
# yum install openstack-keystone httpd mod_wsgi
```

> **Note**
>
> For RHEL8/Centos8 and above install package python3-mod_wsgi.

2. Edit the **/etc/keystone/keystone.conf** file and complete the following actions:
   - In the **[database]** section, configure database access:

```
[database]
# ...
connection = mysql+pymysql://keystone:KEYSTONE_DBPASS@controller/keystone
```

Replace **KEYSTONE_DBPASS** with the password you chose for the database.

```
[token]
# ...
provider = fernet
```

Install & Config (hints)

3. Populate the Identity service database:

```
# su -s /bin/sh -c "keystone-manage db_sync" keystone
```

4. Initialize Fernet key repositories:

```
# keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone
# keystone-manage credential_setup --keystone-user keystone --keystone-group keystone
```

5. Bootstrap the Identity service:

```
# keystone-manage bootstrap --bootstrap-password ADMIN_PASS \
  --bootstrap-admin-url http://controller:5000/v3/ \
  --bootstrap-internal-url http://controller:5000/v3/ \
  --bootstrap-public-url http://controller:5000/v3/ \
  --bootstrap-region-id RegionOne
```

Replace **ADMIN_PASS** with a suitable password for an administrative user.

Install & Config (hints)

Configure the Apache HTTP server

1. Edit the **/etc/httpd/conf/httpd.conf** file and configure the **ServerName** option to reference the controller node:

```
ServerName controller
```

The **ServerName** entry will need to be added if it does not already exist.

2. Create a link to the **/usr/share/keystone/wsgi-keystone.conf** file:

```
# ln -s /usr/share/keystone/wsgi-keystone.conf /etc/httpd/conf.d/
```

Finalize the installation

1. Start the Apache HTTP service and configure it to start when the system boots:

```
# systemctl enable httpd.service
# systemctl start httpd.service
```

2. Configure the administrative account by setting the proper environmental variables:

```
$ export OS_USERNAME=admin
$ export OS_PASSWORD=ADMIN_PASS
$ export OS_PROJECT_NAME=admin
$ export OS_USER_DOMAIN_NAME=Default
$ export OS_PROJECT_DOMAIN_NAME=Default
$ export OS_AUTH_URL=http://controller:5000/v3
$ export OS_IDENTITY_API_VERSION=3
```

These values shown here are the default ones created from **keystone-manage bootstrap**.

Troubleshooting the Identity Service

- To troubleshoot the Identity service: review the logs in the /var/log/keystone/keystone.log file.
- Use the /etc/keystone/logging.conf file to configure the location of log files.

Loging

- The name of the **file specifying the logging configuration** is set using the *log_config_append* option in the *[DEFAULT]* section of the /etc/keystone/keystone.conf file
- To route **logging through syslog**, set ***use_syslog=true*** in the [DEFAULT] section.

- Domain-specific configuration
  - Keyston supports domain-specific Identity drivers
  - By default disabled
  - Stored in domain-specific configuration files, or in the Identity SQL database using API REST calls.
  - **for domain-specific configuration files**
    - Add in /etc/keystone/keystone.conf file

```
[identity]
domain_specific_drivers_enabled = True
domain_config_dir = /etc/keystone/domains
```

- **Storing configuration options in SQL database**
  - Set in the /etc/keystone/keystone.conf

```
[identity]
domain_specific_drivers_enabled = True
domain_configurations_from_database = True
```

Keystone's Primary Benefits

- **Single Authentication and Access Management interface** for other OpenStack services.
- Keystone handles the complex tasks of **integrating with external Authentication systems** and also provides **uniform Access Management** for all the other OpenStack services, such as Nova, Glance, Cinder, Neutron, etc., and thus Keystone **isolates all the other services from knowing how to talk to different identity and authorization providers**.
- Keystone provides a **registry of containers ("Projects")** that other OpenStack services can use to segregate resources (e.g., servers, images, etc.).
- Keystone provides a **registry of Domains** that are used to define separate namespaces for users, groups, and projects to allow segregation between customers.
- A **registry of Roles** that will be used for authorization between Keystone and the policy files of each of the OpenStack services.
- An **assignment store** allowing users and groups to be assigned roles on projects and domains.
- **A catalog storing OpenStack services**, endpoints, and regions, allowing clients to discover the service or endpoint they need.

Conclusions

- The reason Keystone is so important is that it simplifies interaction between and with all of the cloud services.
  - Without the OpenStack Identity Service, every user would have to be separately granted access to every cloud service, greatly increasing administration overhead and creating more opportunities for errors and misconfiguration.
- In summary, the OpenStack Identity Service, or Keystone, provides a range of services that allow users and cloud services to interact with a minimized amount of configuration and a high degree of security.

References

- https://docs.openstack.org/keystone/latest/admin/identity-concepts.html

Backup Slide

Tokens

Tokens : UUID

Pros :

- Simplest and Most Light Weight
- The UUID token is simply a randomly generated UUID 32-character string (Version 4 UUID) getuuid
- The token is extremely small and easy to use when accessing Keystone through a cURL command.

Cons :

-Server side validation (Disadvantage with this token format is that Keystone can become a bottleneck due to the tremendous amount of communication that occurs when Keystone is needed to validate the token.)

- Revoked tokens are not removed from the database. Need to manually flush the database. "keystone-manage token_flush"

Token

Token : PKI/PKIz

These are Cryptographically Encrypted Signed Document using X509 Standards.
Heavy weight as the contain contains the entire validation response that would be received from Keystone.

- Expiry Date
- user identification
- Role information
- service catalog
- other information like region

Pros :

- Client side validation.

Cons :

- Complex to setup (Need Cerificates issued from CA)
- Extremely Large (Size can break the web performance)
- Persisted in database. (Need to manually flush the database.)

Tokens

Token : Fernet Token

Fernet Token :

The newest Keystone token format is the Fernet token format. The Fernet token attempts to improve on previous token formats in a variety of ways.

Pros :

- Small footprint, 255 characters. (larger than UUID tokens, but significantly smaller than PKI)

- Not stored in persistant backend.

Cons :

- Service side validation
- Fernet tokens use symmetric keys to sign the token, nnd these keys need to be distributed to the various OpenStack regions.

https://docs.openstack.org/keystone/wallaby/admin/tokens-overview.html