Container Node:

Identity sevice :

توکنها از یک درخواست به درخواست دیگر تغییر نمی کنند ولی بعد از یک مدت منقضی می شوند . توکن با استفاده از هدرهای X-Auth-Token یا X-Storage-Token به سمت سویفت ارسال می شوند . ساختار هر دو هدر یکسان است و هر دو برابر یک رشته ساده حاوی توکن می باشند .

بعد از اینکه سویفت توکن را تحویل گرفت ، آنرا چک می نماید که مورد تایید باشد . اگر این توکن تایید شد ، یک زمان انضا به آن داده می شود . همچنین برای آنکه هر بار نخواهد آنرا چک نماید ، توکن را برای یک مدت زمان ( که قابل تنظیم است ) آنرا کش میکند که البته این زمان از زمان انقضا توکن بیشتر نیست .

سویفت دو نوع سیستم اهراز هویت دارد :

1. Temp auth
2. Keystone

همچنین می توان یک سیستم اهراز هویت اختصاصی برای سویفت نوشت ( extending auth )

Temp auth

بیشتر در محیط های تستی بکار می رود و به کار بردن آن در محیط production توصیه نمی شود . البته می توان از آن برای توسعه سیستم اهراز هویت اختصاصی الگو گرفت . در این مدل ، کاربرها به دو گونه ادمین و غیر ادمین تقسیم می شوند که کاربرهای ادمین می توانند هر کاری انجام دهند و کاربرهای غیر ادمین فقط می توانند عمل خواندن (read ) انجام دهند . گرچه این کاربران غیر ادمین برخی فراداده های با حق امتیاز ویژه ( privileged metadata ) مانند X-Container-Sync-Key برای کاربر غیر ادمین قابل دسترسی نیستند .

کاربرهای عضو گروه ویژه .reseller\_admin می توانند بر روی همه ی کاربرها عملیات انجام دهند . دیگر کاربرها با تعریف ACL ها می توانند بر روی یک اکانت یا ظرف ( container ) عملیات مورد نظر را انجام دهند . مدل temp auth دو نوع ACL را پشتیبانی می نماید :

* Per container ACLs : بر اساس فراداده های X-Container-Readو X-Container-Write فراداده مربوط به container ها
* Per account ACLs : بر اساس فراداده X-Account-Access-Control مربوط به اکانتها

میان افزار TempAuth مسئول ساختن توکنها می باشد . کاربر یک درخواست شامل نام کاربری و پسورد خود ارسال می نماید . سپس TempAuth در پاسخ به وی یک توکن می دهد که در درخواستهای بعدی ( کار بر روی اکانتها ، container ها و object ها ) به کار می رود .

Keystone Auth

در این محیط ، keystone مسئول ساخت و بررسی توکن ها بوده و میان افزار keystone مسئول اجرای سیستم اهراز هویت می باشد . این مدل از ACLها به ازای هر container پشتیبانی می کند که بر روی فراداده های X-Container-Read و X-Container-Write مربوط به container ها اعمال می شود . در این مدل ACL های سطح اکانت پشتیبانی نمی شوند .

برای استفاده از میان افزار keystoneauth ، میان افزار auth\_token از میان افزارهای keystone باید تنظیم گردد . میان افزار authtoken کار تایید توکن و همچنین بازیابی اطلاعات اهراز هویت از آنرا بر عهده دارد . میان افزار keystone auth نیز کار بررسی مجوز ها ( authorization ) و همچنین نگاشت keystone role ها به swift ACL ها را بر عهده دارد .

تنظیم سویفت برای به کار گیری keystone :

mysql

CREATE DATABASE keystone;

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'localhost' IDENTIFIED BY 'KEYSTONE\_DBPASS'; => KEYSTONE\_DBPASS=salamsalam

GRANT ALL PRIVILEGES ON keystone.\* TO 'keystone'@'%' IDENTIFIED BY 'KEYSTONE\_DBPASS'; => KEYSTONE\_DBPASS=salamsalam

apt-get install keystone

vim.tiny /etc/keystone/keystone.conf

[database]

# ...

connection = mysql+pymysql://keystone:salamsalam@192.168.116.136:3306/keyston=> KEYSTONE\_DBPASS=salamsalam

[token]

# ...

provider = fernet

Populate the Identity service database:

su -s /bin/sh -c "keystone-manage db\_sync" keystone

keystone-manage fernet\_setup --keystone-user keystone --keystone-group keystone

keystone-manage credential\_setup --keystone-user keystone --keystone-group keystone

keystone-manage bootstrap --bootstrap-password salamsalam --bootstrap-admin-url http://swift-proxy:35357/v3/ --bootstrap-internal-url http://swift-proxy:5000/v3/ --bootstrap-public-url http://swift-proxy:5000/v3/ --bootstrap-region-id RegionOne