广西科学院学报
Journal of Guangxi Academy of Sciences

2015,31(1):73~76
Vol. 31, No. 1 February 2015

网络优先数字出版时间:2015-01-14
网络优先数字出版地址:http://www.cnki.net/kcms/detail/45.1075.N.20150114.1023.001.html

OpenStack 核心存储件 Swift 与 Keystone 的集群整合方法*

Integration Method for Swift and Keystone Cluster Components of OpenStack

陈 慧[1],李陶深[1,2]**,岑 霄[1]
CHEN Hui[1], LI Tao-shen[1,2], CEN Xiao[1]

(1.广西大学计算机与电子信息学院,广西南宁 530004;2.广西高校并行分布式计算技术重点实验室,广西南宁 530004)
(1. School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, 530004, China; 2. Guangxi Colleges and Universities Key Laboratory of Parallel and Distributed Computing Technology, Nanning, Guangxi, 530004, China)

摘要:【目的】针对校园网私有云存储系统建设和应用开发需求,研究 OpenStack 核心组件 Swift 和 Keystone 的整合方法。**【方法】**通过 Swift 的 IP 地址获取对象访问令牌(Token),再利用该令牌保证 Swift 能够通过 Keystone 提供的各种插件,实现对用户的身份验证;又通过修改配置文件 proxy-server.conf 的参数,使验证通过的租户可以利用 Keystone 服务的 Token 来调用 Swift 的服务。**【结果】**利用这一集成技术,开发人员可以有效地使用 Swift 组件实现用户身份的验证和调用 Swift 的服务。**【结论】**实际应用表明,该方法有效、可行。

关键词:云平台 OpenStack Swift Keystone 整合
中图分类号:TP31 **文献标识码:**A **文章编号:**1002-7378(2015)01-0073-04

Abstract:【Objective】Based on the needs for construction and application development of private cloud storage system in campus network, this paper researches the integration method for the Swift and Keystone component of the OpenStack. **【Methods】**Through the IP address of the Swift component, an object access token can be obtained. Then, using this token, Swift can achieve the identity authentication of users with the help of different plugins provided by Keystone. By modifying the parameters of the configuration file (proxy-server.conf), the authentication user can use the token provided by Keystone service to invoke the Swift service. **【Results】**Using this integrating technique, the programmers can effectively realize the identity authentication of users and invoke the Swift service. **【Conclusion】**The application results shows that this method is effective and feasible.

Key words: cloud platform, OpenStack, Swift, Keystone, integration

收稿日期:2014-11-10
修回日期:2014-11-21
作者简介:陈 慧(1991-),女,主要从事云计算与云存储技术研究。
*国家自然科学基金项目(61363067)和广西青年基金项目(2013GXNSFBA019281)资助。
**通讯作者:李陶深(1957-),男,教授,博士,主要从事云计算、分布式数据库、无线 mesh 网络等研究。

0 引言

【研究意义】私有云存储是针对特别组织或用户

定制的云存储系统[1]。近年来，随着移动网络的发展，校园网私有云存储系统的应用已成为学校数字化校园的一个重要组成部分。与公有云存储不同，私有云存储一般运行在内部网络的数据中心上，具有较高的安全性，但是私有云存储也存在着可扩展性相对较差的缺点。私有云存储的实现可以应用OpenStack软件，该软件是由NASA和Rackspace联合开发的一个开源项目，旨在提供一个完全开放的、可大规模部署的云计算平台[2]。OpenStack架构一直努力使每个项目尽可能的独立，这使得用户可以选择只部署一个功能子集，并将它与提供类似或互补功能的其他系统和技术相集成。但是在实际的运用过程中，私有云存储系统需要使用OpenStack提供的大部分功能才可以正常运作，需要将各个组件有机地集成起来。**【前人研究进展】**OpenStack项目目前已经获得业界精英企业如戴尔、惠普、英特尔、思科等重要公司和开源组织的支持，成为现在最流行的云计算操作系统软件之一[3]。OpenStack平台提供了可以在标准硬件上配置虚拟机（VM）的软件，而且还提供一个分布式对象存储和广泛的可选功能，包括网络控制器、身份验证管理器、管理仪表板和块存储。其中，Swift组件[4,5]用于分布式对象存储服务，Keystone组件[5]用于处理授权用户的管理。Swift采用层次数据模型，设有账户、容器、对象等三层逻辑结构，每层节点数均没有限制，可以任意扩展。这里的账户和个人账户不是一个概念，可理解为租户，用来做顶层的隔离机制，可以被多个个人账户所共同使用；容器代表封装一组对象，类似文件夹或目录；叶子节点代表对象，由元数据和内容两部分组成。Swift为账户、容器和对象分别定义了环，主要目的是将虚拟节点（分区）映射到一组物理存储设备上，并提供一定的冗余度。**【本研究切入点】**OpenStack核心组件Swift和Keystone在校园网私有云存储系统的整合鲜有报道。**【拟解决的关键问题】**在校园网私有云存储系统的设计与实现中，采用OpenStack平台来搭建完成，利用Swift组件实现对静态的教学资源和相关数据的存储管理，通过Keystone组件实现对用户身份的验证。

1 Keystone的认证

在访问Swift服务之前，需要先通过认证服务获取访问令牌，然后在发送的请求中加入头部信息X-Auth-Token。Swift组件提供的认证服务（Authentication Server）主要用于验证访问用户的身份信息，获得一个对象访问令牌（Token），并确保其在一定的时间内会一直有效。同时也用于验证访问令牌的有效性，并把令牌缓存下来直至过期时间。

在OpenStack平台框架中，Keystone组件的主要职责是包括用户身份的认证和对用户信息的管理等功能，它提供的API调用接口是身份认证API。实际上，Keystone就相当于一个服务总线或者整个OpenStack平台框架的注册表，通过认证服务组件其他服务可以注册服务的退出点。而且任意的两个服务间的相互通信和调用都必须经过Keystone服务组件的身份验证功能，从而才能获得目标服务的Endpoint，并且根据它找到目标服务调用。

在校园网私有云存储系统的设计实现过程中，为了访问服务器，用户需要把自己的证书提交到Keystone并获得一个令牌，然后利用这个令牌保证用户和租户之间的通话。为此，需要将Swift组件和Keystone组件进行有机的整合，使得Swift能够通过Keystone提供的各种插件，实现对用户的身份验证。具体的实现方法：首先，通过Swift的IP地址获取对象访问令牌（Token）；Swift端认证成功之后，以json的格式返回认证解码以及Swift的公共接口，从而完成Swift与Keystone的整合。具体的实现代码如下：

```
//获取 token
$url_base = curl_init('http://192.168.199.128:35357/v2.0/');//swift 的 ip 地址
$postfix = 'tokens';
$auth_user = 'admin';
$auth_pwd = 'admin';
$tenant_name = 'admin';
//set the final auth url.
$auth_url = $url_base.$postfix;
$json = '{"auth":{"passwordCredentials":{"username":"'.$auth_user.'","password":"'.$auth_pwd.'"},"tenantName":"'.$tenant_name.'"}}';
$response = \Httpful\Request::post($auth_url)
->sendsJson()
->body($json)
->send();
//从 json 中取得 token 和 swift url
$result = json_decode($response);
```

```
//get the token
$token = $result->access->token->id;
//get a service list
$service_catalog = $result->access->serviceCatalog;
echo "<p></p>";
//获取 swift 的公共接口
$swift_url = $service_catalog[4]->endpoints[0]->publicURL;
echo "<br>";
```

2　Swift 中配置文件 proxy-server.conf 的修改

除了认证服务外,Swift 组件还提供了代理服务(Proxy Server)、缓存服务(Cache Server)、账户服务(Account Server)、容器服务(Container Server)、对象服务(Object Server)、复制服务(Replicator)、更新服务(Updater)、审计服务(Auditor)、账户清理服务等服务功能。

Swift 组件提供的代理服务是通过 Proxy Server 向外提供基于 HTTP 的 REST 服务接口,实现对账户、容器和对象进行 CRUD 等操作。在 Swift 组件中,它的代理服务对外提供对象服务 API,根据环的信息来查找服务地址并转发用户请求至相应的账户、容器或者对象服务。因此,在具体实现 Swift 与 Keystone 的整合时,需要根据 Keystone 提供的认证授权、系统管理服务的 IP 地址、端口地址等,对配置文件 proxy-server.conf 的一些参数进行修改,使得通过了 Keystone 验证后的用户可以使用 Keystone 服务的 Token,调用 Swift 的服务。在校园网私有云存储系统的设计实现中,修改了 Swift 的配置文件 proxy-server.conf,在其中添加 authtoken 与 keystoneauth 组件,并将 pipeline 中的 tempauth 改为 authtoken 与 keystoneauth,表示采用 Keystone 而不是 TempAuth 来完成用户身份认证和权限控制。authtoken 是 python-keystoneclient 中的组件,用于访问 Keystone;keystoneauth 是 Swift 中的组件,用于一些附加的条件设置。

修改后的配置文件 proxy-server.conf 的内容如下:

```
[DEFAULT]
bind_port = 8080
user = root
workers = 8
log_facility = LOG_LOCAL1
[pipeline:main]
pipeline = healthcheck cache authtoken keystoneauth proxy-logging proxy-server
[app:proxy-server]
use = egg:swift#proxy
allow_account_management = true
account_autocreate = true
[filter:tempauth]
use = egg:swift#tempauth
user_admin_admin = admin .admin .reseller_admin
user_test_tester = testing .admin
user_test2_tester2 = testing2 .admin
user_test_tester3 = testing3
reseller_prefix = AUTH
token_life = 86400

[filter:authtoken]
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
# 以下各项是根据 Keystone 配置文件中的参数及其所在 PC 来设置的。
auth_host = 192.168.3.67
# Keystone 提供的认证授权、系统管理服务的 IP 地址,通常为内网。
auth_port = 35357
# Keystone 提供的认证授权、系统管理服务监听的端口,通常为内网。
auth_protocol = http
# 访问 Keystone 所使用的协议,http 或 https。
service_host = 192.168.3.67
# Keystone 提供的认证授权服务的 IP 地址,通常为公网(外网),也可以是内网。
service_port = 5000
# Keystone 提供的认证授权服务监听的端口,通常为公网(外网),也可以是内网。
admin_token = ADMIN
# admin_token 参数是用来访问 Keystone 服务的,即 Keystone 服务的 Token。可以使用该 Token 访问 Keystone 服务、查看信息、创建其他服务等。
```

```
[filter:keystoneauth]
use = egg:swift#keystoneauth
operator_roles = adminRole,swiftoperator
# 允许访问并使用Swift的角色。
reseller_prefix = AUTH_
# account的命名前缀,注意此处必须加"_"。
# 例如 http://192.168.3.52:8080/v1/AUTH_54d3db64adfc4731b5222cac974f8bc5

[filter:healthcheck]
use = egg:swift#healthcheck

[filter:cache]
use = egg:swift#memcache
memcache_servers = 192.168.3.52:11211,192.168.3.53:11211

[filter:proxy-logging]
use = egg:swift#proxy_logging
```

至此,Openstack已配置完毕。使用以上的配置文件proxy-server.conf,被Keystone验证通过的用户就可以调用Swift的服务。

3 结束语

本文针对校园网私有云存储系统建设的需求,阐述如何实现Openstack开源平台的Swift组件与Keystone组件有效整合的方法。利用这一集成技术,校园网私有云存储系统不需要使用镜像,只是通过安装Swift组件就能够有效地管理存储对象,有效地运用Keystone来实现用户身份的验证,以及实现Openstack环境的配置。这为Openstack开源平台的其它功能组件的有效集成提供了借鉴方法。

参考文献:

[1] Yang S,Jiang L,Zhu S D,et al. Research and Application of Private Cloud Storage Platform in High Schools Based on Seafile[C]. The 6th International Conference on Intelligent Networks and Intelligent Systems,Shenyang,China,November 1-3,2013,25-28.

[2] Wen X L,Gu G Q,Li Q C,et al. Comparison of Open-source Cloud Management Platforms : OpenStack and OpenNebula[C]. The 9th International Conference on Fuzzy Systems and Knowledge Discovery,Chongqing,China,May 29-30,2012,2457-2461.

[3] Younge A J,Laszewski G,Wang L,et al. Efficient Resource Management for Cloud Computing Environments[C]. Proceedings of the 2010 International Conference on Green Computing,Chicago,IL. USA,August 15-18,2010,359-364.

[4] Michael M,Moreira J E,Shiloach D,et al. Scale-up xs-cale-out: Acase Study Using Nutch/Lucene[C]. The 21th International Parallel and Distributed Processing Symposium,Long Beach,California,USA,March 26-30,2007,1-8.

[5] 李辉. 基于OpenStack的私有云计算平台的研究和实现[D]. 南昌:江西师范大学,2013.
Li H. Research and Implementation of the Private Cloud Platform Based on OpenStack[D]. Nanchang: Jiangxi Normal University,2013.

(责任编辑:尹 闯)

(上接第72页 Continue from page 72)

[6] 吴彬,李俊娥. 无线传感器网络在室内定位中的应用研究[J]. 计算机科学,2013,40(5):115-117.
Wu B,Li J E. Application of wireless sensor network in indoor localization[J]. Computer Science,2013,40(5):115-117.

[7] 朱明强,侯建军,刘颖,等. 一种基于卡尔曼数据平滑的分段曲线拟合室内定位算法[J]. 北京交通大学学报,2012,36(5):79-80.
Zhu M Q,Hou J J,Liu Y,et al. An indoor locating algorithm based on kalman smoothing filter and piecewise curve fitting [J]. Journal of Beijing Jiaotong University,2012,36(5):79-80.

[8] 刘雪兰,王宜怀,陆全华,等. 无线传感器网络RSSI定位算法改进[J]. 计算机应用软件,2013,30(11):87-89,141.
Liu X L,Wang Y H,Lu Q H,et al. Improvement for RSSI-based localization algorithm in wireless sensor networks[J]. Computer Applications and Software,2013,30(11):87-89,141.

[9] 张苍松. 基于RSSI的室内定位优化技术[D]. 西安:西北工业大学,2014.
Zhang C S. An Improved Algorithm of Indoor Positioning Which Based on RSSI [D]. Xi'an: Northwest University,2014.

[10] 朱剑,赵海,孙佩刚,等. 基于RSSI均值的等边三角形定位算法[J]. 东北大学学报:自然科学版,2007,28(8):1094-1097.
Zhu J,Zhao H,Sun P G,et al. Equilateral triangle localization algorithm based on average RSSI [J]. Journal of Northeastern University: Natural Science,2007,28(8):1094-1097.

(责任编辑:米慧芝)