



OpenStack故障处理



前言

- OpenStack服务众多，运维人员有必要了解和掌握OpenStack故障处理知识，更好的运维OpenStack。
- 本章节分为两个部分：理论和实验
 - 理论部分主要讲解OpenStack故障处理基础知识、工具、典型排错方法及与故障处理相关的其他OpenStack服务。
 - 实验部分重点练习OpenStack故障处理基本流程、常用命令及典型故障处理案例，帮助学员理论联系实际，真正掌握OpenStack故障处理能力。



目标

- 学完本课程后，您将能够：
 - 描述OpenStack故障处理基本流程
 - 使用OpenStack故障处理常用工具
 - 处理OpenStack典型故障
 - 描述OpenStack与故障处理相关的服务
 - 具备OpenStack日常故障处理能力



目录

- 1. OpenStack故障处理基础**
2. OpenStack故障处理工具
3. OpenStack典型故障处理
4. OpenStack动手实验：故障处理
5. OpenStack故障处理相关项目



OpenStack故障处理一般方法

- OpenStack发生故障时，可以通过以下方法进行故障诊断和处理：

验证OpenStack服务
状态

检查OpenStack服务
日志记录

为OpenStack服务启
用调试模式

检查OpenStack服务
的配置文件

这些方法之间没有严格的先后顺序，请根据实际情况选择不同故障处理方法。



验证OpenStack服务状态

- 要确保OpenStack服务已启动并运行，请验证每个控制器节点上的服务状态。某些OpenStack服务需要在非控制器节点上进行额外验证。

方法一：

- 使用SERVICE_NAME service-list可以快速验证OpenStack服务状态：
例如nova service-list

方法二：

- 如果服务不支持service-list命令，可以使用如下方法验证：
先ps -aux | grep SERVICE_NAME查找出服务
例如ps -aux | grep nova
再service SERVICE_NAME status验证服务状态
例如 service nova-api status



验证OpenStack服务状态一览表

服务	控制节点验证	非控制节点验证
Nova	nova service-list	service nova-compute status
Cinder	service cinder-api status service cinder-scheduler status	service cinder-volume status service cinder-backup status
Glance	service glance-api status service glance-registry status	
Neutron	service neutron-server status neutron agent-list	service neutron-openvswitch-agent status
Heat	heat service-list	
Horizon	service apache2 status netstat -nltp egrep '80 443'	
Keystone	service apache2 status netstat -nltp egrep '5000 35357'	
Swift	swift stat	

如果服务状态异常，可以尝试重启服务，例如service nova-compute restart



检查OpenStack服务日志记录

- OpenStack的日志系统非常完善，大多数的故障都能从日志中找到原因。
- OpenStack日志路径通常在/var/log/SERVICE_NAME下。

节点类型	服务	日志路径
控制节点	nova-*	/var/log/nova
控制节点	glance-*	/var/log/glance
控制节点	cinder-*	/var/log/cinder
控制节点	keystone-*	/var/log/keystone
控制节点	neutron-*	/var/log/neutron
控制节点	horizon	/var/log/apache2/
所有节点	misc (swift, dnsmasq)	/var/log/syslog
计算节点	libvirt	/var/log/libvirt/libvirtd.log
计算节点	VM实例Console（启动消息）	/var/lib/nova/instances/INSTANCE-ID/console.log
块存储节点	cinder-volume	/var/log/cinder/cinder-volume.log



检查OpenStack服务日志记录 - 日志解读

- OpenStack 的日志格式都是统一的：

时间戳 + 日志等级 + 代码模块 + Request ID + 日志内容 + 源代码位置

2019-07-10 20:46:49.566 DEBUG [nova.virt.libvirt.config](#) [req-5c973fff-e9ba-4317-bfd9-76678cc96584 None None] Generated XML ('<[cpu](#)>...</[cpu](#)>') [to_xml](#)
[/opt/stack/nova/nova/virt/libvirt/config.py:82](#)

- 代码模块是nova.virt.libvirt.config
- 日志内容是生成XML文件
- 源代码文件是/opt/stack/nova/nova/virt/libvirt/config.py 的 82 行，方法是to_xml



为OpenStack服务启用调试模式

- 如果需要获取更多日志信息，可以为OpenStack服务启用调试模式。
- 开启调试模式，登录每个控制节点，将OpenStack服务配置文件的DEFAULT部分配置为 “debug=true” 。
 - 例如Nova，配置/etc/nova/nova.conf的DEFAULT部分

```
debug = true
```

注意：

处理故障后，请及时关闭OpenStack服务的调试模式，否则会影响性能。



检查OpenStack服务的配置文件

- **警告：**
 - 一般情况下，**避免修改**OpenStack服务的配置文件，可能严重影响OpenStack。
 - 修改配置文件前，务必确保**提前备份配置文件**，以便随时还原。

核心服务	配置文件路径	
Nova	/etc/nova/api_paste.ini	/etc/nova/nova.conf
Cinder	/etc/cinder/api_paste.ini	/etc/cinder/cinder.conf
Glance	/etc/glance/glance-api.conf /etc/glance/glance-cache.conf /etc/glance/glance-registry.conf	/etc/glance/glance-scrubber.conf /etc/glance/glance-registry-paste.ini /etc/glance/glance-api-paste.ini
Neutron	/etc/neutron/api-paste.ini /etc/neutron/neutron.conf /etc/neutron/dhcp_agent.ini	/etc/neutron/metadata_agent.ini /etc/neutron/plugins/evs/evs_plugin.ini /etc/neutron/l3_agent.ini
Heat	/etc/heat/api_paste.ini	/etc/heat/heat.conf
Horizon	/etc/openstack_dashboard/local_settings.py	/etc/apache2/apache2.conf
Keystone	/etc/keystone/keystone-paste.ini	/etc/keystone/keystone.conf



重要提示：OpenStack外部故障

计算、存储、网络节点故障

- 磁盘空间不足
- 交换分区空间不足
- 内存空间不足
- CPU负载过高
- 文件系统故障
- 物理节点故障

网络连接故障

- IP冲突
- 交换机配置错误
- 网线故障

其他故障

- 时间同步
- DNS
- 防火墙



目录

1. OpenStack故障处理基础
- 2. OpenStack故障处理工具**
3. OpenStack典型故障处理
4. OpenStack动手实验：故障处理
5. OpenStack故障处理相关项目



OpenStack故障处理常用工具 - 网络

- 网络故障处理常用工具

```
$ ip address  
$ ip netns list  
$ ip netns exec  
$ route -n
```

```
$ sudo ovs-vsctl show  
$ sudo brctl show  
$ sudo ovs-tcpdump -i br-int  
$ sudo tcpdump -i <tap-dev>
```

- 如果运行命令时提示 “Command not found” ， 可以先安装工具后再使用：
 - Ubuntu : `sudo apt install`
 - CentOS : `sudo yum install`



OpenStack故障处理常用工具 - 计算和存储

- 计算和存储故障处理常用工具

```
$ lsb_release -a  
$ uname -a  
$ df -h  
$ free -m
```

```
$ top  
$ iostat  
$ dmesg  
$ qemu-nbd  
$ kpartx
```



OpenStack故障处理常用工具 - Inav日志查看

- 基于消息的时间戳，Inav能把多个日志文件合并到一个视图，并将警告和错误信息以不同颜色高亮显示，故障处理更加直观。
 - 例如，合并查看Nova和Glance的日志，追踪虚拟机请求启动镜像的详情情况：

```
$ Inav /var/log/nova/nova-api.log /var/log/glance/glance-api.log
```

```
Tue Aug 13 /var/log/glance/glance-api.log: openstack log LOG
- [13/Aug/2019 02:01:21] "GET /v2/images/87ccf4fd-c261-4f29-a973-f8c68ff3b44c
- [13/Aug/2019 02:01:21] "GET /v2/images/87ccf4fd-c261-4f29-a973-f8c68ff3b44c/
- [13/Aug/2019 02:01:21] "GET /v2/images/87ccf4fd-c261-4f29-a973-f8c68ff3b44c
- [13/Aug/2019 02:01:21] "GET /v2/schemas/image HTTP/1.1" 200 5571 0.002721
- [13/Aug/2019 02:01:22] "GET /v2/images/87ccf4fd-c261-4f29-a973-f8c68ff3b44c
- [13/Aug/2019 02:01:22] "GET /v2/schemas/image HTTP/1.1" 200 5571 0.002805
- [13/Aug/2019 02:01:23] "GET /v2/images/87ccf4fd-c261-4f29-a973-f8c68ff3b44c
- [13/Aug/2019 02:01:23] "GET /v2/schemas/image HTTP/1.1" 200 5571 0.002834
default default] Creating event network-changed:ad65891c-c59a-454c-b39d-d876fb
.2.0.11 "POST /v2.1/os-server-external-events HTTP/1.1" status: 200 len: 578 t
default default] Creating event network-vif-plugged:ad65891c-c59a-454c-b39d-d8
.2.0.11 "POST /v2.1/os-server-external-events HTTP/1.1" status: 200 len: 582 t
uses the service catalog via common keystoneauth1 Adapter configuration optio
200 len: 151 time: 0.9957917
.2.0.11 "GET /v2.1 HTTP/1.1" status: 302 len: 313 time: 0.2753255
.2.0.11 "GET /v2.1/ HTTP/1.1" status: 200 len: 784 time: 0.0067503
.2.0.11 "GET /v2.1/servers/detail?all_tenants=True&limit=21&sort_dir=desc HTTP
.2.0.11 "GET /v2.1/flavors/detail HTTP/1.1" status: 200 len: 860 time: 0.02975
.2.0.11 "GET /v2.1/extensions HTTP/1.1" status: 200 len: 23101 time: 0.0048733
```

glance-api.log

nova-api.log



OpenStack故障处理常用工具 - 服务和日志查看

- 服务和日志查看常用工具

```
$ grep -aux | grep nova  
$ grep REQ_ID /var/log/nova/*  
$ less /var/log/nova/nova-api.log  
$ less /var/log/nova/*  
$ zgrep * VOLUME_ID  
$ zgrep VOLUME_ID * | grep ERROR
```

```
$ journalctl --unit nova-api.service  
$ journalctl -u nova*  
$ journalctl -u nova* | grep <id>  
$ journalctl -a # colors  
$ journalctl --since -1hour # limit history
```



OpenStack故障处理常用工具 - 命令行debug

- 执行OpenStack命令时，使用“debug”选项可以追踪命令详细执行过程。

```
$ openstack server create --flavor m1.nano --image cirros-0.4.0-x86_64-disk --network private test --debug
...
auth_config_hook(): {'auth_type': 'password', 'beta_command': False, u'image_status_code_retries': '5'
defaults: {u'auth_type': 'password', u'status': u'active', u'image_status_code_retries': 5, 'api_time
cloud cfg: {'auth_type': 'password', 'beta_command': False, u'image_status_code_retries': '5', u'int
...
command: server create -> openstackclient.compute.v2.server.CreateServer (auth=True)
...
Using parameters {'username': 'demo', 'project_name': 'demo', 'user_domain_id': 'default', 'auth_url'
Get auth_ref
REQ: curl -g -i -X GET http://192.168.10.15/identity -H "Accept: application/json" -H "User-Agent: op
Starting new HTTP connection (1): 192.168.10.15:80
http://192.168.10.15:80 "GET /identity HTTP/1.1" 300 272
RESP: [300] Connection: close Content-Length: 272 Content-Type: application/json Date: Mon, 29 Apr 20
RESP BODY: {"versions": {"values": [{"status": "stable", "updated": "2019-01-22T00:00:00Z", "media-ty
...
http://192.168.10.15:80 "POST /identity/v3/auth/tokens HTTP/1.1" 201 3253
{"token": {"is_domain": false, "methods": ["password"], "roles": [{"id": "9ae9e8b27dcb419598a8952f4d8
Instantiating image api: <class 'openstackclient.api.image_v2.APIv2'>
curl -g -i -X GET -H 'Accept-Encoding: gzip, deflate' -H 'Accept: */*' -H 'User-Agent: python-glancec
...
REQ: curl -g -i -X GET http://192.168.10.15/compute/v2.1/flavors/m1.nano -H "Accept: application/json
Resetting dropped connection: 192.168.10.15
http://192.168.10.15:80 "GET /compute/v2.1/flavors/m1.nano HTTP/1.1" 404 80
RESP: [404] Connection: close Content-Length: 80 Content-Type: application/json; charset=UTF-8 Date:
RESP BODY: {"itemNotFound": {"message": "Flavor m1.nano could not be found.", "code": 404}}
```

--debug

Set environment

Parse arguments

Request auth(n|z)

Request image

Request flavor



实验时间：OpenStack故障处理常用工具

- 请登录OpenStack实验环境，练习OpenStack故障处理常用工具。
 - 请将命令都 --help 一次，初步了解每个命令的作用，下节内容会详细介绍用法。
 - 例如tcpdump -help

```
osbash@controller2:~$ tcpdump -help
tcpdump version 4.9.2
libpcap version 1.8.1
OpenSSL 1.1.1 11 Sep 2018
Usage: tcpdump [-aAbdDefhHIJKlLnNOpqStuUvxxX#] [-B size] [-c count]
               [-C file_size] [-E algo:secret] [-F file] [-G seconds]
               [-i interface] [-j tstamptype] [-M secret] [--number]
               [-Q in|out|inout]
               [-r file] [-s snaplen] [--time-stamp-precision precision]
               [--immediate-mode] [-T type] [--version] [-V file]
               [-w file] [-W filecount] [-y datalinktype] [-z postrotat
e-command ]
               [-Z user] [ expression ]
```



目录

1. OpenStack故障处理基础
2. OpenStack故障处理工具
- 3. OpenStack典型故障处理**
4. OpenStack动手实验：故障处理
5. OpenStack故障处理相关项目



OpenStack典型故障处理

Keystone故障处理案例

- 服务连接超时
- 认证失败

Neutron故障处理案例

- 无法ping通虚拟机

Nova故障处理案例

- No valid host was found



OpenStack常见故障代码

- OpenStack出现故障时，一般会有故障代码，提示是什么类型故障。

40X错误：

- 指一个服务已经正常运行，但用户执行的命令或调度任务出错：
 - a) 例如，401是身份认证失败，403是权限被拒绝

50X错误：

- 指一个服务没有正常运行或不能访问，建议检查该服务是否运行和访问正常：
 - a) 例如，500是内部错误，503是服务不可用



Keystone故障案例 - 服务连接超时

- 故障现象：服务连接超时，日志中提示“HTTP 500”错误。
- 故障原因：Keystone Catalog配置错误，导致访问不到服务，连接超时。
- 故障处理过程：

- 1、检查端点配置

```
$ openstack catalog list
```

- 2、修改端点信息

```
$ openstack endpoint set --service compute --interface admin --url URL
```



Keystone故障案例 - 认证失败

- 故障现象：身份认证失败，日志中提示“HTTP 401” 错误。
- 故障原因：Keystone用户丢失、密码输入错误、用户被禁用等。
- 故障处理过程：

- 1、检查用户是否存在、名称是否正确

```
$ openstack user list
```

- 2、检查用户详细信息

```
$ openstack user show USER_ID
```

- 3、重置用户密码或重新启用用户

```
$ openstack user set --password NEW_PASSWORD USER_ID
```

```
$ openstack user set --enable USER_ID
```




Neutron网络故障处理准备

- 在开始排除OpenStack网络故障前，建议提前准备：

- 检查Neutron相关服务是否正常工作

```
$ openstack network agent list
```

- 检查Neutron日志

日志类型	日志路径	所在节点
vSwitch代理日志	/var/log/neutron/openvswitch-agent.log	计算节点
L3代理日志	/var/log/neutron/l3-agent.log	运行L3代理的网络节点
DHCP代理日志	/var/log/neutron/dhcp-agent.log	运行DHCP代理的网络节点
DNSMASQ日志	/var/log/messages	运行DHCP代理的同一节点

```
$ sudo grep -E -R -i "trace|error" /var/log/neutron/
```

```
$ sudo lnav /var/log/neutron/
```



Neutron故障案例 - 无法ping通虚拟机

- 故障现象：无法ping通虚拟机固定IP或浮动IP
- 故障分析：固定IP一般由DHCP分配，浮动IP一般由虚拟路由器管理，需逐步排错
- 故障处理过程：
 - 1、检查虚拟机的安全组规则是否放行ICMP，未放行的话，添加规则放行ICMP

```
$ openstack server show INSTANCE_ID | grep security_groups
```

```
osbash@controller2:~$ openstack server show b6944647-4f1c-4933-b41c-7f34302b128  
6 | grep security_groups  
| security_groups | name='sg01'
```

安全组名称

```
$ openstack security group rule list SECURITY_GROUP_NAME | grep icmp
```

```
osbash@controller2:~$ openstack security group rule list sg01 | grep icmp
```

无回显，没有icmp规则

```
$ openstack security group rule create --protocol icmp SECURITY_GROUP_NAME
```

```
osbash@controller2:~$ openstack security group rule list sg01 | grep icmp  
| 2c41a291-65ad-404d-8f26-d10d36fd653d | icmp | 0.0.0.0/0 |
```

添加icmp规则后，有回显



Neutron故障案例 - 无法ping通虚拟机

- 2、检查虚拟机Console日志，是否为虚拟机分配了IP地址

```
$ openstack console log show INSTANCE_ID
```

```
osbash@controller2:~$ openstack console log show b6944647-4f1c-4933-b41c-7f34302b1286
ip-route:default via 203.█.113.1 dev eth0
ip-route:169.254.169.254 via 203.█.113.101 dev eth0
ip-route:203.█.113.0/24 dev eth0 src 203.█.113.110
ip-route6:fe80::/64 dev eth0 metric 256
ip-route6:unreachable default dev lo metric -1 error -101
ip-route6:ff00::/8 dev eth0 metric 256
ip-route6:unreachable default dev lo metric -1 error -101
```

IP地址分配

截图显示虚拟机实例分配到的IP地址为203.X.113.110



Neutron故障案例 - 无法ping通虚拟机

- 3、如果虚拟机实例没有被分配IP地址，检查子网是否启用了DHCP

- 3.1 查找虚拟机网络信息

```
$ openstack server show INSTANCE_ID | grep addresses
```

```
osbash@controller2:~$ openstack server show b6944647-4f1c-4933-b41c-7f34302b1286 | grep addresses  
| addresses | provider=203...113.110
```

Network名称

- 3.2 检查子网是否启用DHCP

```
$ openstack network show NETWORK_ID | grep subnets
```

```
osbash@controller2:~$ openstack network show provider | grep subnets  
| subnets | 8829c6df-6aee-4220-a96a-c5337656f233
```

Subnet id

```
$ openstack subnet show SUBNET_ID | grep enable_dhcp
```

```
osbash@controller2:~$ openstack subnet show 8829c6df-6aee-4220-a96a-c5337656f233 | grep enable_dhcp  
| enable_dhcp | True
```

True为开启DHCP

- 3.3 重启DHCP服务

```
$ service neutron-dhcp-agent restart
```



Neutron故障案例 - 无法ping通虚拟机

- 4、如果启用DHCP后，还是不能ping通虚拟机固定IP，需检查DHCP命名空间

- 4.1 找到实例的网络ID

```
$ nova interface-list INSTANCE_ID
```

```
osbash@controller2:~$ nova interface-list b6944647-4f1c-4933-b41c-7f34302b1286
```

Port	State	Port ID	Net ID
ACTIVE		ad65891c-c59a-454c-b39d-d876fb41aa45	2f0875eb-d43d-4813-8866-a0b694be59d3

Network ID

- 4.2 找到网络的dhcp-agent位置

```
$ neutron dhcp-agent-list-hosting-net NETWORK_ID
```

```
osbash@controller2:~$ neutron dhcp-agent-list-hosting-net 2f0875eb-d43d-4813-8866-a0b694be59d3
```

neutron CLI is deprecated and will be removed in the future. Use openstack CLI instead.

id	host	admin_state_up	alive
f729f489-7968-4efb-a2df-3d7fe3a4768d	controller2	True	

dhcp-agent所在host名称



Neutron故障案例 - 无法ping通虚拟机

- 4、如果启用DHCP后，还是不能ping通虚拟机固定IP，需检查DHCP命名空间
 - 4.3 从DHCP命名空间ping虚拟机

```
$ ssh HOST_IP
```

```
osbash@controller:~$ ssh controller2
The authenticity of host 'controller2 (10.2.0.11)' can't be established.
ECDSA key fingerprint is SHA256:WjWhSnqJuSWQkY61WXckBR/lgIEoaBv6uXBZEy99+iY.
Are you sure you want to continue connecting (yes/no)? yes
```

```
$ sudo ip netns exec qdhcp-NETWORK_ID ping INSTANCE_IP_ADDRESS
```

```
osbash@controller2:~$ sudo ip netns exec qdhcp-2f0875eb-d43d-4813-8866-a0b694be59d3 ping 203.0.113.110
PING 203.0.113.110 (203.0.113.110) 56(84) bytes of data:
64 bytes from 203.0.113.110: icmp_seq=1 ttl=64 time=7.67 ms
```

能够从dhcp命名空间ping通虚拟机



Neutron故障案例 - 无法ping通虚拟机

- 5、如果不能从DHCP命名空间ping通虚拟机固定IP，需使用tcpdump抓包检查
 - 5.1 找到命名空间中DHCP接口

```
$ ip netns exec qdhcp-NETWORK_ID ip address
```

```
osbash@controller2:~$ sudo ip netns exec qdhcp-2f0875eb-d43d-4813-8866-a0b694be59d3 ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ns-735c082e-d4@if12: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group 0
    link/ether fa:16:3e:16:3eff:fe39:2ee9/64 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 169.254.169.254/16 brd 169.254.255.255 scope global ns-735c082e-d4
        valid_lft forever preferred_lft forever
    inet 203.0.113.101/24 brd 203.0.113.255 scope global ns-735c082e-d4
        valid_lft forever preferred_lft forever
    inet6 fe80::16:3eff:fe39:2ee9/64 scope link
        valid_lft forever preferred_lft forever
```

DHCP接口



Neutron故障案例 - 无法ping通虚拟机

- 5、如果不能从DHCP命名空间ping通虚拟机固定IP，需使用tcpdump抓包检查

- 5.2 抓包DHCP流量

```
$ ip netns exec qdhcp-NETWORK_ID tcpdump -i DHCP_PORT -e port 67 or port 68
```

```
osbash@controller2:~$ sudo ip netns exec qdhcp-2f0875eb-d43d-4813-8866-a0b694be59d3 tcpdump -i ns-735c082e-d4 -e port 67 or port 68  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on ns-735c082e-d4, link-type EN10MB (Ethernet), capture size 262144 bytes
```

- 5.3 如果没有DHCP流量包，从DHCP命名空间ping网关

```
$ ip netns exec qdhcp-NETWORK_ID ping GATEWAY_ADDRESS
```

```
osbash@controller2:~$ sudo ip netns exec qdhcp-2f0875eb-d43d-4813-8866-a0b694be59d3 ping 203.2.113.1  
PING 203.2.113.1 (203.2.113.1) 56(84) bytes of data.  
64 bytes from 203.2.113.1: icmp_seq=1 ttl=64 time=0.663 ms
```




Neutron故障案例 - 无法ping通虚拟机

- 6、如果能ping通虚拟机固定IP，不能ping通浮动IP，需检查虚拟路由器命名空间

- 6.1 找到浮动IP关联的路由ID

```
$ nova interface-list INSTANCE_ID
```

- 6.2 找到网络的l3-agent位置

```
$ neutron l3-agent-list-hosting-router ROUTER_ID
```

- 6.3 从虚拟路由器命名空间ping虚拟机

```
$ ip netns exec qrouter-ROUTER_ID ping INSTANCE_IP_ADDRESS
```

```
$ ip netns exec qrouter-ROUTER_ID ping INSTANCE_FLOATING_IP
```



Nova计算故障处理准备

- 在开始排除OpenStack计算故障前，建议提前准备：
 - 检查Nova相关服务是否正常工作

```
$ nova service-list
```

```
$ service nova-compute status
```

- 检查Nova日志

日志类型	日志路径	所在节点
nova-*	/var/log/nova	控制节点
libvirt	/var/log/libvirt/libvirtd.log	计算节点
VM Console	/var/lib/nova/instances/instance- <id> /console.log	计算节点

```
$ grep -E -R -i "trace|error" /var/log/nova/*.log
```

```
$ lnav /var/log/nova/*.log
```



Nova故障案例 - No valid host was found

- 故障现象：
 - 创建虚拟机时，提示 “No valid host was found” ， 创建失败。
- 故障分析：
 - Nova使用/etc/nova/nova.conf 配置的Filter Scheduler做调度分配，如果所有调度条件均不满足，就会出现 “No valid host was found” ， 虚拟机创建失败。
 - scheduler_default_filters=RetryFilter,AvailabilityZoneFilter,RamFilter,ComputeFilter,ComputeCapabilitiesFilter,ImagePropertiesFilter,CoreFilter



Nova故障案例 - No valid host was found

- 故障处理过程:

- 1、检查Nova日志, 筛选 “No valid host” 错误日志, 找出Request ID

```
$ grep -E -R -i " No valid host" /var/log/nova/*.log
```

```
osbash@controller2:~$ grep -E -R -i "No valid host" /var/log/nova/*.log
/var/log/nova/nova-conductor.log:2019-08-13 08:32:00.470 1533 ERROR nova.conductor.manager [req-db0f93d8-4b1b-41
62-8ff8-7a1cbf88aa95 1ed978f18a5447659282baeb47e9b11c 2cf53da011744e3ab4103e40d45d046d - default default] Failed
to schedule instances: nova.exception_remote.NoValidHost_Remote: No valid host was found. There are not enough
hosts available.
/var/log/nova/nova-conductor.log:nova.exception.NoValidHost: No valid host was found. There are not enough hosts
available.
/var/log/nova/nova-conductor.log:2019-08-13 08:32:00.470 1533 ERROR nova.conductor.manager [req-db0f93d8-4b1b-41
62-8ff8-7a1cbf88aa95 1ed978f18a5447659282baeb47e9b11c 2cf53da011744e3ab4103e40d45d046d - default default] Failed
to schedule instances: nova.exception_remote.NoValidHost_Remote: No valid host was found. There are not enough hosts
available.
/var/log/nova/nova-conductor.log:2019-08-13 08:32:00.470 1533 ERROR nova.conductor.manager [req-db0f93d8-4b1b-41
62-8ff8-7a1cbf88aa95 1ed978f18a5447659282baeb47e9b11c 2cf53da011744e3ab4103e40d45d046d - default default] Failed
to schedule instances: nova.exception_remote.NoValidHost_Remote: No valid host was found. There are not enough hosts
available.
/var/log/nova/nova-conductor.log:2019-08-13 08:32:00.703 1533 WARNING nova.scheduler.utils [req-db0f93d8-4b1b-41
62-8ff8-7a1cbf88aa95 1ed978f18a5447659282baeb47e9b11c 2cf53da011744e3ab4103e40d45d046d - default default] Failed
to compute_task_build_instances: No valid host was found. There are not enough hosts available.
```

Request ID



Nova故障案例 - No valid host was found

- 故障处理过程：
 - 2、检查Nova日志，筛选Request ID所有记录，找出故障详细原因

```
$ grep Request_ID /var/log/nova/*.log
```

```
osbash@controller2:~$ grep req-db0f93d8-4b1b-4162-8ff8-7a1cbf88aa95 /var/log/nova/*.log
/var/log/nova/nova-api.log:2019-08-13 08:31:59.201 1666 INFO nova.osapi.compute.wsgi.server [req-db0f93d8-4b1b-4162-8ff8-7a1cbf88aa95 1ed978f18a5447659282baeb47e9b11c 40d45d046d - default default] 10.2.0.11 "POST /v2.1/servers HTTP/1.1" status: 202 len: 89
/var/log/nova/nova-conductor.log:2019-08-13 08:32:00.470 1533 ERROR nova.compute.manager [req-db0f93d8-4b1b-4162-8ff8-7a1cbf88aa95 1ed978f18a5447659282baeb47e9b11c 2cf53da011744e3ab4103e40d45d046d - default default] Failed
/var/log/nova/nova-scheduler.log:2019-08-13 08:32:00.446 1392 WARNING nova.scheduler.filters.compute_filter [req-db0f93d8-4b1b-4162-8ff8-7a1cbf88aa95 1ed978f18a5447659282baeb47e9b11c 2cf53da011744e3ab4103e40d45d046d - default default] (compute2, compute2) ram: 1352MB disk: 5120MB io_ops: 0 instances: 1 has not been heard from in a while
/var/log/nova/nova-scheduler.log:2019-08-13 08:32:00.447 1392 INFO nova.filters [req-db0f93d8-4b1b-4162-8ff8-7a1cbf88aa95 1ed978f18a5447659282baeb47e9b11c 2cf53da011744e3ab4103e40d45d046d - default default] Filter ComputeFilter returned 0 hosts
/var/log/nova/nova-scheduler.log:2019-08-13 08:32:00.447 1392 INFO nova.filters [req-db0f93d8-4b1b-4162-8ff8-7a1cbf88aa95 1ed978f18a5447659282baeb47e9b11c 2cf53da011744e3ab4103e40d45d046d - default default] Filtering removed all hosts for the request with instance ID '72f8f5b1-9213-4488-9969-f4823dfd6d53'. Filter results: ['RetryFilter: (start: 1, end: 1)', 'AvailabilityZoneFilter: (start: 1, end: 1)', 'ComputeFilter: (start: 1, end: 0)']
```

compute_filter失败



Nova故障案例 - No valid host was found

- 故障处理过程：
 - 3、根据不同调度器故障，采用不同故障处理方法

调度器	故障处理方法	故障处理动作
CoreFilter RamFilter	检查主机可用资源	nova hypervisor-stats
	检查超分配参数 /etc/nova/nova.conf	cpu_allocation_ratio = 16.0 (节点物理CPU*16) ram_allocation_ratio = 1.5 (节点物理内存*1.5) reserved_host_memory_mb=512 (预留，虚拟机不能使用)
RetryFilter	检查hypervisor状态	service nova-compute status
AvailabilityZoneFilter	检查AZ中是否有hypervisor	openstack availability zone list nova service-list --binary nova-compute



Nova故障案例 - No valid host was found

- 故障处理过程：
 - 3、根据不同调度器故障，采用不同故障处理方法

调度器	故障处理方法	故障处理动作
ComputeFilter	检查是否有正常工作的计算节点	nova service-list --binary nova-compute
ComputeCapabilitiesFilter	检查Hypervisor是否能满足Flavor定义的所有属性和自定义参数	openstack flavor list openstack flavor show <ID> nova hypervisor-list nova hypervisor-show <ID>
ImagePropertiesFilter	检查Hypervisor是否能满足Image定义的所有属性和自定义参数	openstack image list openstack image show <ID> nova hypervisor-list nova hypervisor-show <ID>



目录

1. OpenStack故障处理基础
2. OpenStack故障处理工具
3. OpenStack典型故障处理
4. **OpenStack动手实验：故障处理**
5. OpenStack故障处理相关项目



动手实验：使用日志、CLI处理故障

- 日志查看
- 认证失败处理
- 虚拟机实例创建失败处理
- 分配浮动IP失败处理



目录

1. OpenStack故障处理基础
2. OpenStack故障处理工具
3. OpenStack典型故障处理
4. OpenStack动手实验：故障处理
- 5. OpenStack故障处理相关项目**



根因分析Vitrage



VITRAGE

根因分析服务

首次出现在OpenStack的“Newton”版本中。

简介

Vitrage为OpenStack提供故障根本原因分析。

Vitrage能够组织，分析和可视化OpenStack警报和事件，给出问题出现的根本原因，并能提前预测可能出现的问题点。

依赖的OpenStack服务



Keystone

帮助OpenStack故障处理

预防故障，出现故障时帮助找出故障根本原因。



告警Aodh



AODH

告警服务

首次出现在OpenStack的“Liberty”版本中。

简介

Aodh为OpenStack提供资源告警服务。

Aodh主要基于ceilometer所获取的测量值或者事件，当达到告警阈值时，自动触发告警。

依赖的OpenStack服务



Keystone

帮助OpenStack故障处理

监控资源状态，产生故障告警事件和消息。



监控Monasca



MONASCA

监控服务

首次出现在OpenStack的“Mitaka”版本中。

简介

Monasca是与OpenStack集成的多租户，可扩展，高性能，容错的监控即服务解决方案。Monasca使用REST API进行高速处理和查询，并具有流式警报引擎和通知引擎。

依赖的OpenStack服务



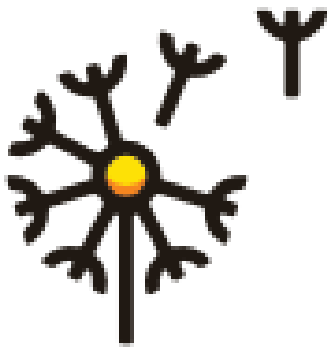
Keystone

帮助OpenStack故障处理

监控告警，及时通知故障处理。



workflowMistral



MISTRAL

workflow服务

首次出现在OpenStack的“Liberty”版本中。

简介

Mistral为OpenStack提供 workflow服务。

Mistral能够将多个业务流程在分布式环境中以特定顺序自动执行，保证业务执行的正确性。

依赖的OpenStack服务



Keystone

帮助OpenStack故障处理

提前定义故障处理 workflow，出现特定故障时自动执行 workflow，自动处理故障。



灾备Freezer



FREEZER

灾备服务

首次出现在OpenStack的“Mitaka”版本中。

简介

Freezer提供分布式备份，还原和灾难恢复服务。

Freezer支持不同类型操作系统（Linux，Windows，OSX ...），提供基于块的备份，基于文件的增量备份，时间点操作，作业同步等功能。

依赖的OpenStack服务



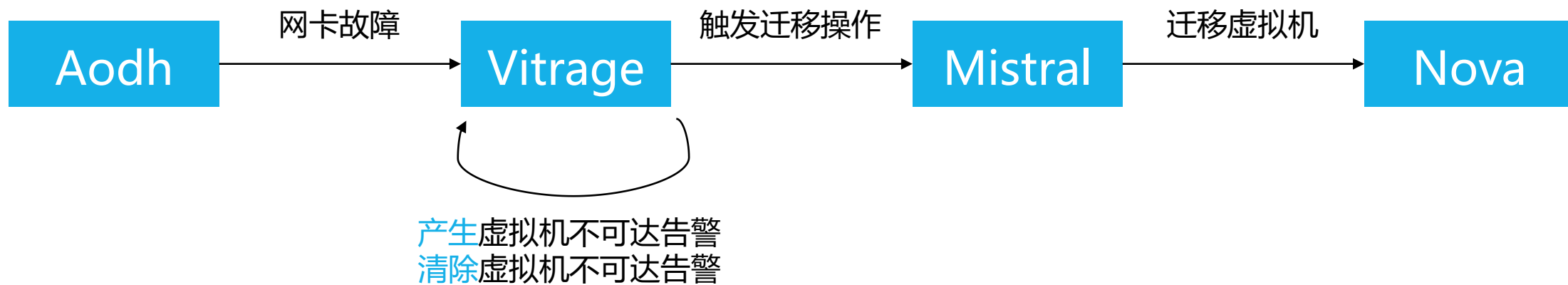
Keystone

帮助OpenStack故障处理

发生故障时，避免丢失数据，恢复数据和业务运行。



示例：智能化OpenStack故障处理





思考题

1. 有哪些常用OpenStack故障处理工具?
2. 如果虚拟机不能从DHCP获取IP, 如何排除故障?
3. 如果不能创建虚拟机, 提示“ No valid host was found” , 如何排除故障?



本章总结

- OpenStack故障处理基础
- OpenStack故障处理工具
- OpenStack典型故障处理
- OpenStack故障处理相关项目



学习推荐

- OpenStack社区
 - <https://www.openstack.org/>

The background of the slide features a blue-tinted image of several business professionals in a modern office. They are standing on a highly reflective floor, and their silhouettes are clearly visible. The people are engaged in various activities: some are looking at documents, others are talking, and one person is holding a folder. The overall atmosphere is professional and collaborative.

谢谢

www.huawei.com