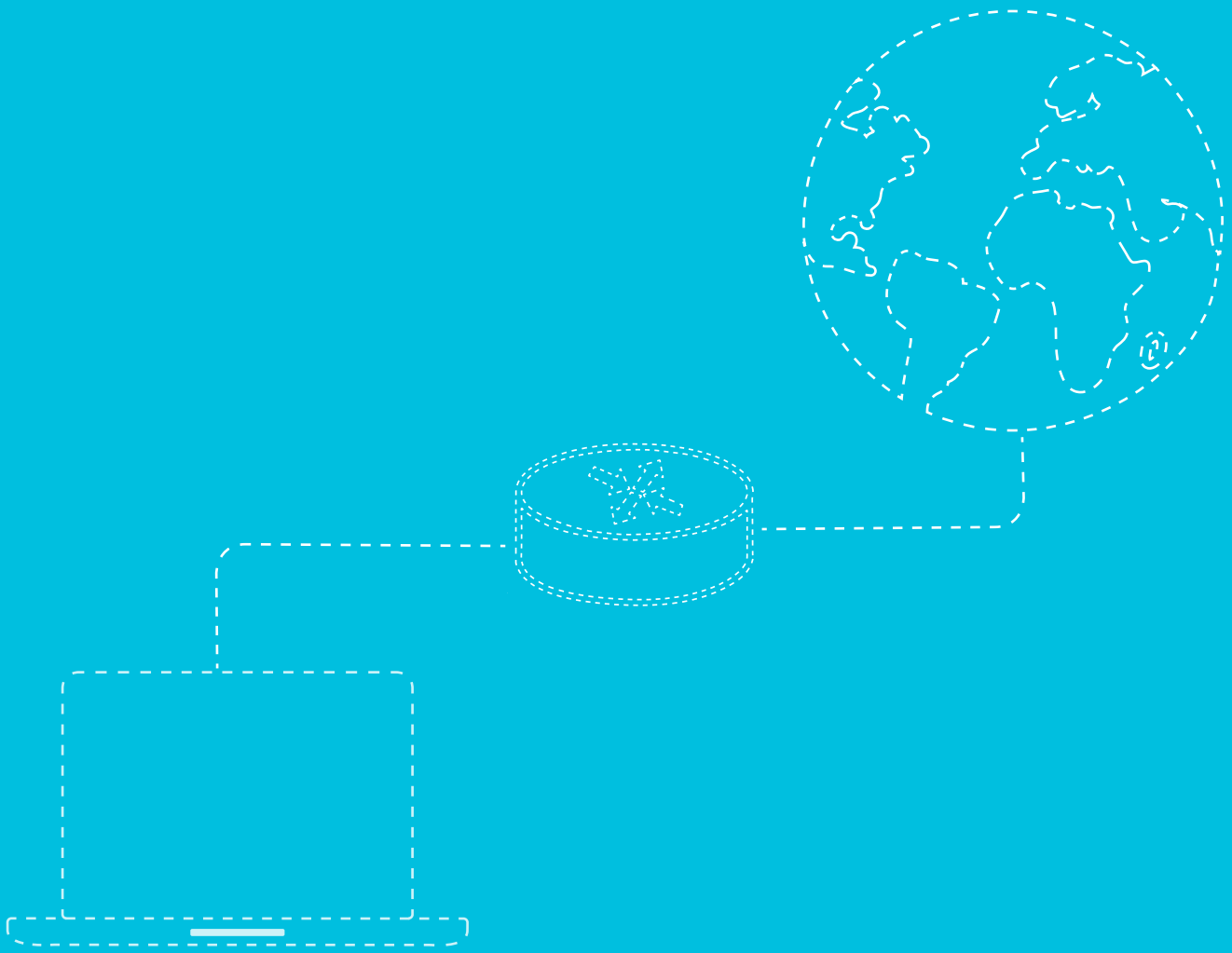
Alibaba Cloud
aliyun.com

Best Practices for
Alibaba Cloud

Setting up a Highly Available NAT



Abstract

Network address translation (NAT) is a method of re-mapping one IP address space into another by modifying network address information in Internet Protocol (IP) datagram packet headers, while they are in transit across a traffic routing device. Originally, users deployed the technique to ease traffic rerouting in IP networks without renumbering every host. It has become a popular and essential tool in the conservation of global address space allocations when faced with IPv4 address exhaustion. It works by sharing one Internet-routable IP address of a NAT gateway for an entire private network.

NAT gateway is a critical component in any cloud setup where instances in a private subnet need a robust and effective way to connect to the Internet.

01 Introduction

The aim of this post is to explain a typical setup for a highly available NAT that avoids single-point-of-failures (SPOF) in outgoing communications from a VPC private subnet. The setup explained below is a combination of Alibaba Cloud features and proven open source technologies.

02 Prerequisites

- A vSwitch where the NAT instances need to be created
- EIP for the HAVIP
- Two instances, one as NAT Master and another as NAT Slave
- Make sure the Alibaba Cloud Cli is setup correctly

    - sudo pip install aliyuncli
    - sudo pip install aliyun-python-sdk-ecs
    - sudo pip install aliyun-python-sdk-rds
    - sudo pip install aliyun-python-sdk-slb
    - sudo pip install aliyun-python-sdk-oss
    - sudo cp -r /usr/local/lib/python2.7/dist-packages/aliyun* /usr/lib/python2.7/dist-packages/
      (If it is an Ubuntu instance)

03 Setup

3.1. HAVIP

HAVIP or Highly Available Virtual IP is a software defined virtual IP interface that allows mapping to more than one ECS. HAVIP allows instant remapping capability using keepalived. It grants you the ability to design and create High Availability (HA) server infrastructures (setups that do not have SPOFs), by adding redundancy to the entry point, or gateway, on your servers. Achieving a complete HA setup also requires redundancy at every layer of your infrastructure, such as your application and database servers. While this is often difficult to implement, it can prove to be invaluable for reducing downtime and maintaining a satisfied user base. One can create and manage HAVIP through webpage or APIs. However, we are using Alibaba Cloud CLI in the examples described below:

Note: Make sure you enable the HAVIP for the account before you commence the installation process.

3.2. Create the HAVIP

Run the CreateHaVip command to create a new HAVIP:

```
$ aliyuncli ecs CreateHaVip --VSwitchId vsw-6wt1c7ls1
{
   "RequestId": "D9DEFA09-B37C-4712-A9C9-1938335C84D2",
   "HaVipId": "havip-fd1i40zx9"
}
```

3.3. Associate Instances

The following commands will help you associate instances:

```
$ aliyuncli ecs AssociateHaVip --InstanceId i-22wfp7s7w --HaVipId havip-fd1i40zx9
{
   "RequestId": "326FA988-A87C-40DE-B7F7-BC0906F54E3D"
}
```

```
$ aliyuncli ecs AssociateHaVip --InstanceId i-22m8hdsjl --HaVipId havip-fd1i40zx9
{
   "RequestId": "C225EA64-8E8D-4F89-BCAF-D3576894DD7B"
}
```

3.4. Associate EIP

We need to associate a public IP to the HAVIP so that the instances behind the HAVIP can connect to the Internet.

```
$ aliyuncli ecs AssociateEipAddress --AllocationId eip-vegv5bc6b --InstanceId havip-fd1i40zx9
--InstanceType havip
{
   "RequestId": "763C6126-AA32-4A44-9FE0-9E22D25E5A17"
}
```

3.5. Router Config

The next step involves finding the details of the router table so we can accordingly set the routes:

```
$ aliyuncli ecs DescribeVRouters
{
   "VRouters": {
      "VRouter": [
         {
            "VRouterId": "vrt-os82ug3j4",
            "Description": "",
            "RegionId": "ap-southeast-1",
            "CreationTime": "2016-09-02T02:56:54Z",
            "VpcId": "vpc-9brecklu1",
            "VRouterName": "",
            "RouteTableIds": {
               "RouteTableId": [
```

```
                    "vtb-ifxyuondn"
                ]
            }
        }
    ]
  },
  "TotalCount": 1,
  "PageNumber": 1,
  "RequestId": "9AB18495-54F7-43C0-872D-AC0B374DC60E",
  "PageSize": 10
}
```


Set the route table entry to make sure that sending all outgoing requests to occur through the NAT instance.


```
$ aliyuncli ecs CreateRouteEntry --NextHopId havip-fd1i40zx9 --NextHopType HaVip
--RouteTableId vtb-ifxyuondn --DestinationCidrBlock 0.0.0.0
{
  "RequestId": "25236262-F255-4D34-8FD7-AFFA15F8593F"
}
```


3.6. Setup NAT

Login to the NAT instances (both master and slave) and execute the commands stated below.

- Enable ip_forwarding by editing the following file:
  $ vi /etc/sysctl.conf
- Make sure this entry is set to 1, and is not commented out:
  net.ipv4.ip_forward = 1
- Restart the network config to take effect by executing the following command:
  $ sysctl -p
- Add the NAT rule to the iptables by executing the command below:
  $ iptables –t nat –A POSTROUTING –j MASQUERADE
- Save the iptables to survive reboot by executing the following: (For Ubuntu)
  $ iptables-save

3.7. Setup Keepalived

Keepalived is a routing software written in C. The main goal of this project is to provide simple and robust facilities for load balancing and high availability for Linux-based infrastructures. Load balancing framework relies on well-known and widely used Linux Virtual Server (IPVS) kernel module providing Layer4 load balancing.

Perform the following actions in both the NAT instances.

Prepare the instance for setup

vi /etc/apt/source.list

```
deb http://mirrors.aliyun.com/ubuntu/ trusty main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ trusty-security main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ trusty-updates main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ trusty-proposed main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ trusty-backports main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ trusty main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ trusty-security main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ trusty-updates main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ trusty-proposed main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ trusty-backports main restricted universe multiverse

deb http://sg.mirrors.aliyuncs.com/ubuntu/ trusty main restricted universe multiverse
deb http://sg.mirrors.aliyuncs.com/ubuntu/ trusty-security main restricted universe multiverse
deb http://sg.mirrors.aliyuncs.com/ubuntu/ trusty-updates main restricted universe multiverse
deb http://sg.mirrors.aliyuncs.com/ubuntu/ trusty-proposed main restricted universe multiverse
deb http://sg.mirrors.aliyuncs.com/ubuntu/ trusty-backports main restricted universe multiverse
deb-src http://sg.mirrors.aliyuncs.com/ubuntu/ trusty main restricted universe multiverse
deb-src http://sg.mirrors.aliyuncs.com/ubuntu/ trusty-security main restricted universe multiverse
deb-src http://sg.mirrors.aliyuncs.com/ubuntu/ trusty-updates main restricted universe multiverse
deb-src http://sg.mirrors.aliyuncs.com/ubuntu/ trusty-proposed main restricted universe multiverse
deb-src http://sg.mirrors.aliyuncs.com/ubuntu/ trusty-backports main restricted universe multiverse
```

apt-get update

```
sudo apt-get install libssl-dev
wget   http://www.keepalived.org/software/keepalived-1.2.19.tar.gz
tar -zxf keepalived-1.2.19.tar.gz
cd keepalived-1.2.19/
./configure
make && make install
```

Setup the keepalived start and stop scripts

mkdir /etc/keepalived
cd /etc/keepalived/
mkdir scripts

vi /etc/keepalived/scripts/ha_vip_start.sh

```
#!/bin/bash

echo "start; `date`" >> /tmp/log
sysctl -w net.ipv4.ip_forward=1
iptables -t nat -A POSTROUTING -d 100.64.0.0/10 -j RETURN
iptables -t nat -A POSTROUTING -s 10.0.0.0/8 ! -p vrrp -j SNAT --to-source 10.100.1.0
```

vi /etc/keepalived/scripts/ha_vip_stop.sh

```
#!/bin/bash

echo "stop; `date`" >> /tmp/log
iptables -t nat –F
```

```
chmod +x /etc/keepalived/scripts/*
cp /usr/local/sbin/keepalived /usr/sbin/
```

Setup the keepalived configuration

```
vi /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global_defs {
  router_id LVS_DEVEL
}

vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
        10.100.1.0 dev eth0 label eth0:havip
}
    notify_master /etc/keepalived/scripts/ha_vip_start.sh
    notify_backup /etc/keepalived/scripts/ha_vip_stop.sh
    notify_fault  /etc/keepalived/scripts/ha_vip_stop.sh
    notify_stop   /etc/keepalived/scripts/ha_vip_stop.sh
    unicast_src_ip 10.100.1.1
    unicast_peer {
          10.100.1.2
              }
}
```

Setup the auto startup scripts

```
vi /etc/init/keepalived.conf
vi /etc/rc.local
```

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
```

```
description "load-balancing and high-availability service"

start on runlevel [2345]
stop on runlevel [!2345]

respawn

exec /usr/local/sbin/keepalived --dont-fork
```

Start Keepalived

sudo start keepalived

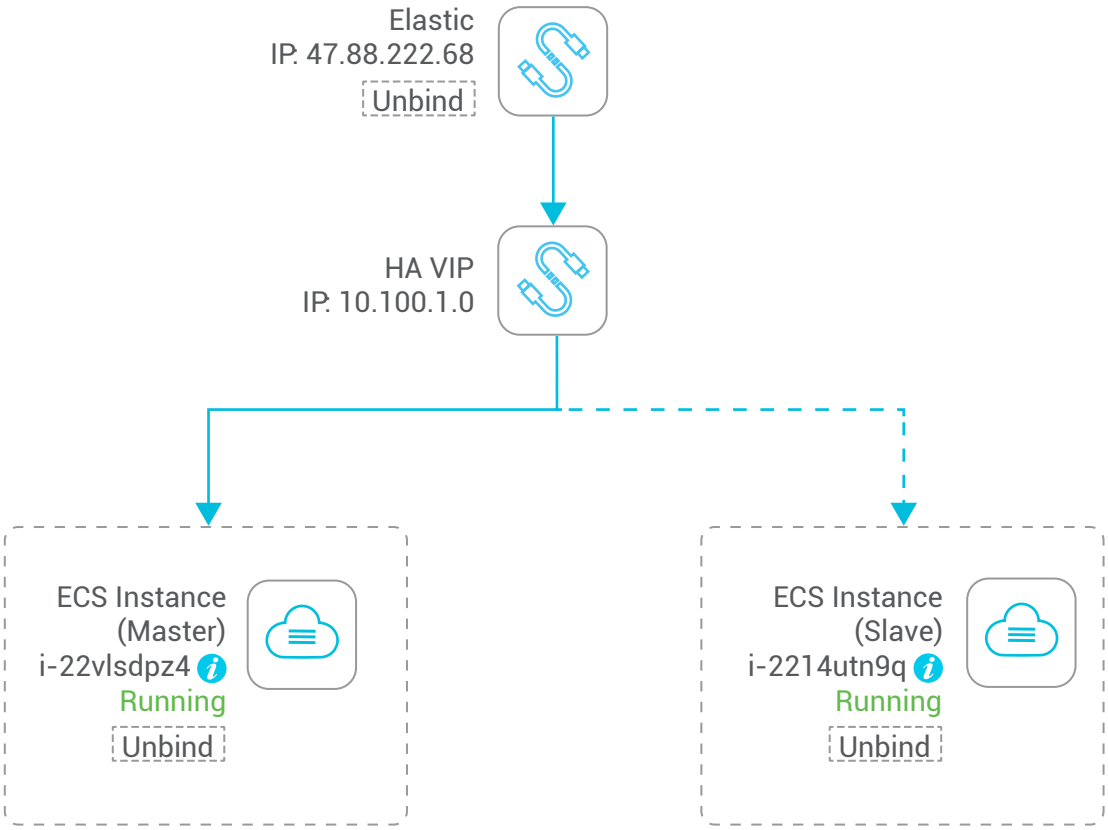
Add more routes as required

sudo iptables -t nat -A PREROUTING -p tcp --dport 10106 -j DNAT --to-destination 172.16.102.6:22
sudo iptables -t nat -A PREROUTING -p tcp --dport 10106 -j DNAT --to-destination 172.16.102.6:22

04 Testing

You can manually bring down instances in this step. Also, ensure that outward communications from client instances in the same VPC work fine.

05 End Result



06 Conclusion

This document attempts to introduce the concept of HA NAT and its basic setup. While this architecture is not fully available (cross zones), one can easily extend it to be failsafe by relying on HA features available with Alibaba Cloud. There are other options to perform the connection. Please refer to http://intl.aliyun.com for more reference documents.

Further Information

VPC
https://intl.aliyun.com/docs#/pub/vpc_en_us

ECS
https://intl.aliyun.com/docs#/pub/ecs_en_us

Reference

http://www.keepalived.org/software/keepalived-1.2.23.tar.gz

https://raymii.org/s/tutorials/Keepalived-Simple-IP-failover-on-Ubuntu.html

intl.aliyun.com