

OrcaWave Communication Platform

High Availability Solution

Aug 2018

COPYRIGHT

Copyright © 2007–2018 BDRS, Inc. All rights reserved.

PROPRIETARY NOTICES

OrcaWave is a registered trademark of BDRS, Inc.

VMware, VMware ESX, and VMware server are trademarks of VMware, Inc.

XenServer, and XenCenter are trademarks of Citrix Systems, Inc.

Asterisk is a registered trademark of Digium.

FreePBX is a registered trademark of Sangoma.

All other trademarks are the property of their respective owners.

RELEASE DATE: October 2015

Table of Contents

OrcaWave Communication Platform.....	1
Preface.....	4
Intended Audience.....	5
Chapter 1: OrcaWave HA Configuration.....	6
Hardware Requirements	7
Vyatta Clustering.....	7
To monitor Asterisk with Monit	10
Create SSH Key to login without password	13
Freepbx Backup and Restore	13
Glossary.....	18

Preface

This document describes the various deployment and installation of OrcaWave.

Intended Audience

This guide is intended for experienced system and network administrators. Depending on the functionality to be used, readers should have specific knowledge in the following areas:

- Networking and data communications
- TCP/IP protocols
- General router configuration
- Routing protocols
- Network administration
- Network security
- IP service
- SIP protocol

Chapter 1: OrcaWave HA Configuration

This section presents the following:

- OrcaWave HA configuration

Hardware Requirements

Two OrcaWave Same Model Number with 2 NIC cards

First Step we do the Router Clustering.

Vyatta Clustering

Login to OrcaWave 192.168.100.1

```
vyatta@OrcaWave$ configure
vyatta@OrcaWave# set system host-name OrcaWave-Primary
vyatta@OrcaWave# set interfaces ethernet eth0 address 192.168.100.2/24
vyatta@OrcaWave# commit
vyatta@OrcaWave# save
vyatta@OrcaWave# nano /etc/hosts
```

127.0.0.1 localhost

The following lines are desirable for IPv6 capable hosts

```
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
192.168.100.2 OrcaWave-Primary
127.0.1.1 OrcaWave-Primary #vyatta entry
```

Relog to OrcaWave with the new address 192.168.100.2

```
vyatta@OrcaWave-Primary# set cluster interface eth0
vyatta@OrcaWave-Primary# set cluster group OrcaWave primary OrcaWave-Primary
vyatta@OrcaWave-Primary# set cluster group OrcaWave secondary OrcaWave-secondary
vyatta@OrcaWave-Primary# set cluster group OrcaWave service 192.168.100.1/24/eth0
vyatta@OrcaWave-Primary# set cluster pre-shared-secret ANY
vyatta@OrcaWave-Primary# set cluster group OrcaWave auto-failback true
```

```
vyatta@OrcaWave-Primary# set service conntrack-sync failover-mechanism cluster group
OrcaWave
vyatta@OrcaWave-Primary# set service conntrack-sync interface eth0
```

Login to OrcaWave Second Box 192.168.100.1

```
vyatta@OrcaWave$ configure
vyatta@OrcaWave# set system host-name OrcaWave-Secondary
vyatta@OrcaWave# set interfaces ethernet eth0 address 192.168.100.3/24
OrcaWave
```

```
vyatta@OrcaWave# commit
vyatta@OrcaWave# save
vyatta@OrcaWave# nano /etc/hosts
```

127.0.0.1 localhost

The following lines are desirable for IPv6 capable hosts

```
::1 ip6-localhost ip6-loopback
```

```
fe00::0 ip6-localnet
```

```
ff00::0 ip6-mcastprefix
```

```
ff02::1 ip6-allnodes
```

```
ff02::2 ip6-allrouters
```

```
ff02::3 ip6-allhosts
```

192.168.100.3 OrcaWave-Secondary

```
127.0.1.1 OrcaWave-Secondary #vyatta entry
```

Relog to OrcaWave with the new address 192.168.100.3

```
vyatta@OrcaWave-Secondary# set cluster interface eth0
```

```
vyatta@OrcaWave-Secondary# set cluster group OrcaWave primary OrcaWave-Primary
```

```
vyatta@OrcaWave-Secondary# set cluster group OrcaWave secondary OrcaWave-secondary
```

```
vyatta@OrcaWave-Secondary# set cluster group OrcaWave service 192.168.100.1/24/eth0
```

```
vyatta@OrcaWave-Secondary# set cluster pre-shared-secret ANY
```

```
vyatta@OrcaWave-Secondary# set cluster group OrcaWave auto-failback true
```

```
vyatta@OrcaWave-Secondary# set service conntrack-sync failover-mechanism cluster group
OrcaWave
```

```
vyatta@OrcaWave-Secondary# set service conntrack-sync interface eth0
```

```
vyatta@OrcaWave-Secondary# commit
```

```
vyatta@OrcaWave-Secondary# save
```

Log back to Primary Box 192.168.100.2

```
vyatta@OrcaWave-Primary# run show cluster status
```

```
=== Status report on primary node OrcaWave-Primary ===
```

Primary OrcaWave-Primary (this node): Active

Secondary OrcaWave-Secondary: Active (standby)

Resources [192.168.100.1/24/eth0]:

Active on primary OrcaWave-Primary (this node)

[edit]

```
vyatta@OrcaWave-Primary#
```

Login to the Secondary Box OrcaWave-Secondary 192.168.100.3


```
vyatta@OrcaWave-Secondary# run show cluster status  
=== Status report on secondary node OrcaWave-Secondary ===
```

```
Primary OrcaWave-Primary: Active
```

```
Secondary OrcaWave-Secondary (this node): Active (standby)
```

```
Resources [192.168.100.1/24/eth0]:
```

```
Active on primary OrcaWave-Primary
```

```
[edit]
```

```
vyatta@OrcaWave-Secondary#
```

```
vyatta@OrcaWave-Primary# show service dhcp-server
```

```
shared-network-name Primary {
```

```
subnet 192.168.1.0/24 {
```

```
default-router 192.168.1.1
```

```
dns-server 192.168.1.1
```

```
failover {
```

```
local-address 192.168.1.10
```

```
name FAILOVER
```

```
peer-address 192.168.1.11
```

```
status primary
```

```
}
```

```
start 192.168.1.20 {
```

```
stop 192.168.1.100
```

```
}
```

```
tftp-server-name 192.168.1.1
```

```
}
```

```
}
```

```
[edit]
```

```
vyatta@OrcaWave-Primary#
```

```
vyatta@OrcaWave-Secondary# show service dhcp-server
```

```
shared-network-name Primary {
```

```
subnet 192.168.1.0/24 {
```

```
default-router 192.168.1.1
```

```
dns-server 192.168.1.1
```

```
failover {
```

```
local-address 192.168.1.11
```

```
name FAILOVER
```

```
peer-address 192.168.1.10
```

```
status secondary
```

```
}
```

```
start 192.168.1.20 {
```

```
stop 192.168.1.100
```

```
}
```

```
tftp-server-name 192.168.1.1
}
}
```

[edit]

vyatta@OrcaWave-Secondary#

Time should be synced on both routers!

Now try to Shutdown the OrcaWave-Primary and check the cluster configuration.

- Automatically start Asterisk at boot time

```
cd /usr/src/asterisk-1.8.8.0/contrib/init.d/ ## contrib folder is in source directory
```

```
sudo cp rc.debian.asterisk /etc/init.d/asterisk
```

```
sudo chmod 755 /etc/init.d/asterisk
```

```
# to set the script as boot
```

```
sudo update-rc.d asterisk defaults
```

```
# to remove the script from boot
```

```
update-rc.d -f asterisk remove
```

```
Modify the
```

```
sudo nano /etc/init.d/asterisk
```

```
# Full path to asterisk
```

```
binary## DAEMON=__ASTERISK_SBIN_DIR__/asteriskDAEMON=/usr/sbin/asterisk
```

```
## ASTVARRUNDIR=__ASTERISK_VARRUN_DIR__ASTVARRUNDIR=/var/run/asterisk/
```

```
## ASTETCDIR=__ASTERISK_ETC_DIR__ASTETCDIR=/etc/asterisk/
```

To monitor Asterisk with Monit

Make sure to configure both servers in this step

```
vyatta@OrcaWave-Secondary:~$ sudo apt-get install monit
```

```
vyatta@OrcaWave-Secondary:~$ sudo nano /etc/rc.local
```

```
add sudo /etc/init.d/monit start
```

```
#!/bin/sh -e
```

```
#
```

```
# rc.local
```

```
#
```

```
# This script is executed at the end of each multiuser runlevel.
```

```
# value on error.
```

```
#  
# In order to enable or disable this script just change the execution  
# bits.  
#  
# By default this script does nothing.  
  
# Do not remove the following call to vyatta-postconfig-bootup.script.  
# Any boot time workarounds should be put in script below so that they  
# get preserved for the new image during image upgrade.  
POSTCONFIG=/opt/vyatta/etc/config/scripts/vyatta-postconfig-bootup.script  
[ -x $POSTCONFIG ] && $POSTCONFIG  
sudo /usr/local/sbin/amportal start  
sudo /etc/init.d/tftpd-hpa start  
sudo /etc/init.d/monit start  
exit 0
```

save and exit.

Based on Monit documentation <http://mmonit.com/monit/documentation/monit.html>

```
vyatta@OrcaWave-Secondary:~$ sudo nano /etc/default/monit
```

```
Defaults for monit initscript  
# sourced by /etc/init.d/monit  
# installed at /etc/default/monit by maintainer scripts  
# Stefan Alfredsson <alfs@debian.org>
```

```
# You must set this variable to for monit to start  
startup=1
```

```
# You can change the location of the state file here  
# It can also be set in monitrc  
# STATEFILE="/var/lib/monit/monit.state"
```

```
# To change the intervals which monit should run,  
# edit the configuration file /etc/monit/monitrc  
# It can no longer be configured here.  
sudo nano /etc/monit/conf.d/orcawave
```

```
# Make monit start its web-server. So you can access it from webbrowser.  
set httpd port 2812  
use address 0.0.0.0
```

```
##Monit web-server ACL.  
allow localhost    # allow localhost to connect to the server and  
allow 0.0.0.0/0.0.0.0 # allow 192.168.1.2 to connect to the server,  
                    # You can give only one per entry
```

```
allow admin:monit # user name and password for authentication.
```

```
allow hardik:hardik # set multiple user to access through browser.
```

```
set daemon 120
```

```
# poll at 2-minute intervals. Monit will wakeup every two minute to
```

```
# monitor things. Time must be given in seconds.
```

```
# Asterisk Monitoring rule
```

```
set daemon 30 # Check every 30s
```

```
set logfile syslog facility log_daemon
```

```
check process asterisk with pidfile /var/run/asterisk/asterisk.pid
```

```
group asterisk
```

```
start program = "/etc/init.d/asterisk start"
```

```
stop program = "/etc/init.d/asterisk stop"
```

```
# Check uptime via Asterisk Manager Interface (AMI) port 5038
```

```
if failed host 127.0.0.1 port 5038 then restart
```

```
if 5 restarts within 5 cycles then exec /sbin/reboot
```

Reboot the OrcaWave and check if monit is working with

```
vyatta@OrcaWave-Secondary:~$ sudo monit status
```

```
The Monit daemon 5.1.1 uptime: 4m
```

```
Process 'asterisk'
```

```
status          running
monitoring status    monitored
pid              3541
parent pid        1
uptime           4m
children          0
memory kilobytes    16784
memory kilobytes total 16784
memory percent      0.8%
memory percent total 0.8%
cpu percent         0.0%
cpu percent total    0.0%
port response time   0.000s to 127.0.0.1:5038 [DEFAULT via TCP]
data collected       Thu Oct 9 14:26:45 2014
```

```
System 'OrcaWave-Secondary'
```

```
status          running
monitoring status    monitored
load average       [0.00] [0.05] [0.05]
```

```
OrcaWave
```

cpu 0.0%us 0.0%sy 0.0%wa
memory usage 102332 kB [4.9%]
data collected Thu Oct 9 14:26:45 2014

vyatta@OrcaWave-Secondary:~\$

Create SSH Key to login without password

vyatta@OrcaWave-Primary:~\$ configure
[edit]

vyatta@OrcaWave-Primary# set service ssh allow-root

login to OrcaWave-Secondary 192.168.100.3

sudo -u asterisk ssh-keygen

sudo -u asterisk ssh-copy-id -i /var/lib/asterisk/.ssh/id_rsa.pub -p2212 [root@OrcaWave-Primary](#)

Try to login from OrcaWave-Secondary to OrcaWave-Primary

sudo ssh -i /var/lib/asterisk/.ssh/id_rsa root@[192.168.100.2](#) -p 2212

Freepbx Backup and Restore

Create Primary Server under Backup - Servers - SSH

OrcaWave
UNIFIED COMMUNICATIONS
Admin ▾ Applications ▾ Connectivity ▾ Reports ▾ Settings ▾ User Panel

SSH Server

Server Name	OrcaWave-Primary
Description	OrcaWave-Primary
Hostname	192.168.100.2
Port	2212
User Name	root
Key	/var/lib/asterisk/.ssh/id_rsa
Path	/usr/src/

Save
Delete

Create Backup and don't forget to exclude the backup setting so it doesn't change the backup configuration on the secondary server or the backup setting on the secondary server will be lost and will never run again!

OrcaWave
UNIFIED COMMUNICATIONS
Admin Applications Connectivity Reports Settings User Panel

Backup

Backup Name OrcaWave-Primary
Description OrcaWave-Primary

Items

Backup Items

Type	Path/DB	Exclude	Delete
Directory	__ASTETCDIR__	PATTERNs, one per line	
Directory	/etc/dahdi	PATTERNs, one per line	
Directory	__ASTSPOOLDIR__/voicemail	PATTERNs, one per line	
Directory	__AMPWEBROOT__	PATTERNs, one per line	
Mysql	CDR server	table names, one per line	
Asterisk DB		Family, one per line	
Mysql	Config server	<div> <div> <div>backup</div> <div>backup_cache</div> <div>backup_details</div> <div>backup_items</div> <div>backup_server_details</div> <div>backup_servers</div> <div>backup_template_details</div> <div>ls</div> </div> </div>	
Directory	__ASTVARLIBDIR__/sounds/custom	PATTERNs, one per line	
Directory	__ASTVARLIBDIR__/moh	PATTERNs, one per line	

Templates

- CDR's
- Config Backup
- Exclude Backup Settings
- Full Backup
- System Audio
- Voice Mail

== chose ==

Choose to restore from OrcaWave Primary to Local Server (OrcaWave-Secondary).
It will copy the Primary configuration and restore it on the Secondary.

OrcaWave
UNIFIED COMMUNICATIONS
Admin Applications Connectivity Reports Settings

Backup Server

Backup Server OrcaWave-Primary

Restore Here ☒

Disable Registered Trunks ☐

Apply Configs ☒

Storage Locations

Storage Servers

+ Local Storage (local)

Available Servers

+ Legacy Backup (local)
+ OrcaWave-Primary (ssh)

Backup Schedule

Run Automatically Daily

Randomize ☐

Maintenance

Delete after 0 Minutes

Delete after 0 runs

Save and Run Delete

Try to save and run, it should look like this


```
Run backup
/bin/tar: ./var/www/Vyatta/images/logo_top_banner_bg.gif: Cannot open: File exists
/bin/tar: Exiting with failure status due to previous errors
File restore complete!
Restoring CDR's...
Getting CDR size...
Processed 100% of CDR's!
Restoring CDR's complete
Restoring settings...
Getting Settings size...
Processed 100% of Settings!
Restoring astDB...
Restoring Settings' complete
Running post-restore hooks, if any...
Cleaning up...
Restore complete!
Reloading...
Done!
Running post-backup hooks...
Backup successfully completed!
done!
```

And you have OrcaWave HA Solution ready!

Glossary

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	Dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point

DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Ouput
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP security
IPv4	IP Version 4
IPv6	IP Version 6
ISP	Internet Service Provider
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol

LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
MIB	Management Information Base
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
ND	Neighbor Discovery
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
P2P	peer-to-peer
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol

PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
Rx	receive
S3	Amazon Simple Storage Service
SIP	Session Initiation Protocol
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SSH	Secure Shell
SSID	Service Set Identifier
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
Tx	transmit
UDP	User Datagram Protocol
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	Virtual Private Network

VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access