Red Hat
Enterprise Linux

Ansible Automation Platform with External HA PostgreSQL Cluster

Marc Skinner
Principal Solutions Architect

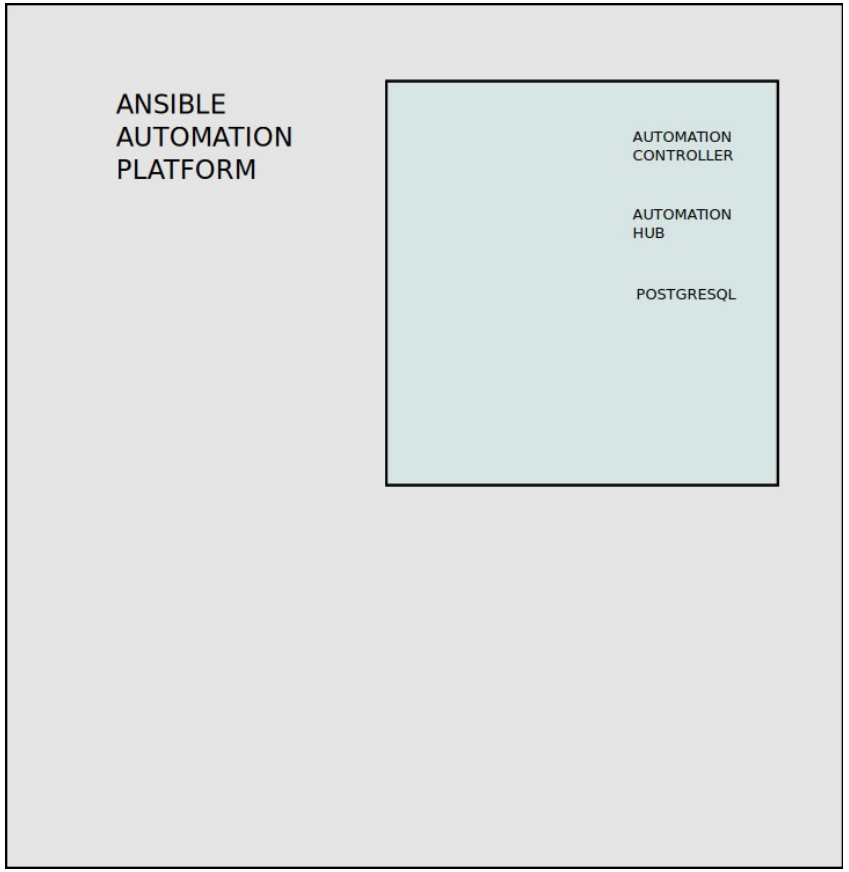
Red Hat

AAP Architecture?

Single Node

All Services

- Easy
- Lab, Test, Small

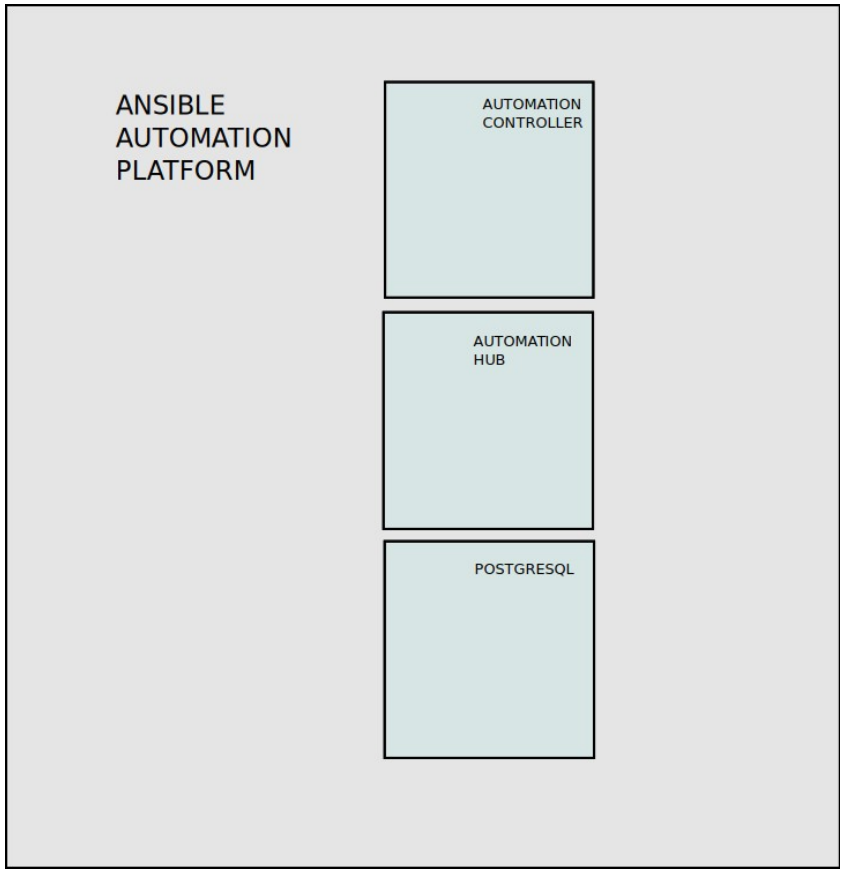


Red Hat

Three Node

One Service Per Node

- Separation
- Small/Medium

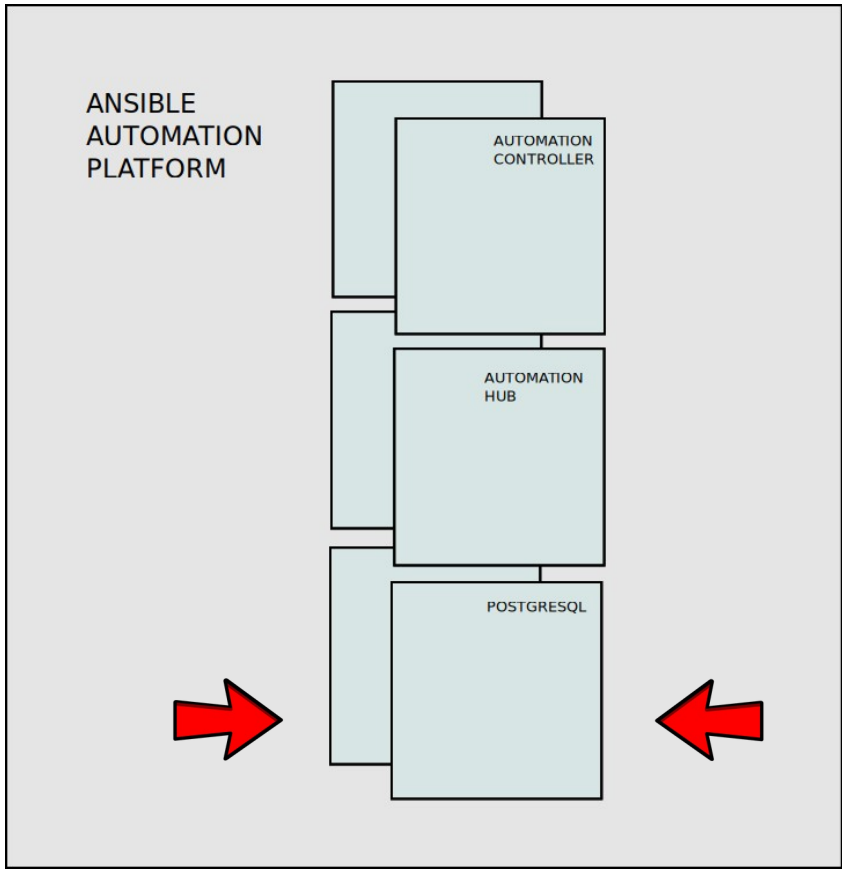


Red Hat

N- Node

Many Services On Many Nodes

- HA
- Large
- Scale
- Performance



Red Hat

What does support say?

AAP Database Scope of Coverage

Supported By Red Hat

- Embedded PostgreSQL installed by AAP installer
- Version 13

Unsupported by Red Hat

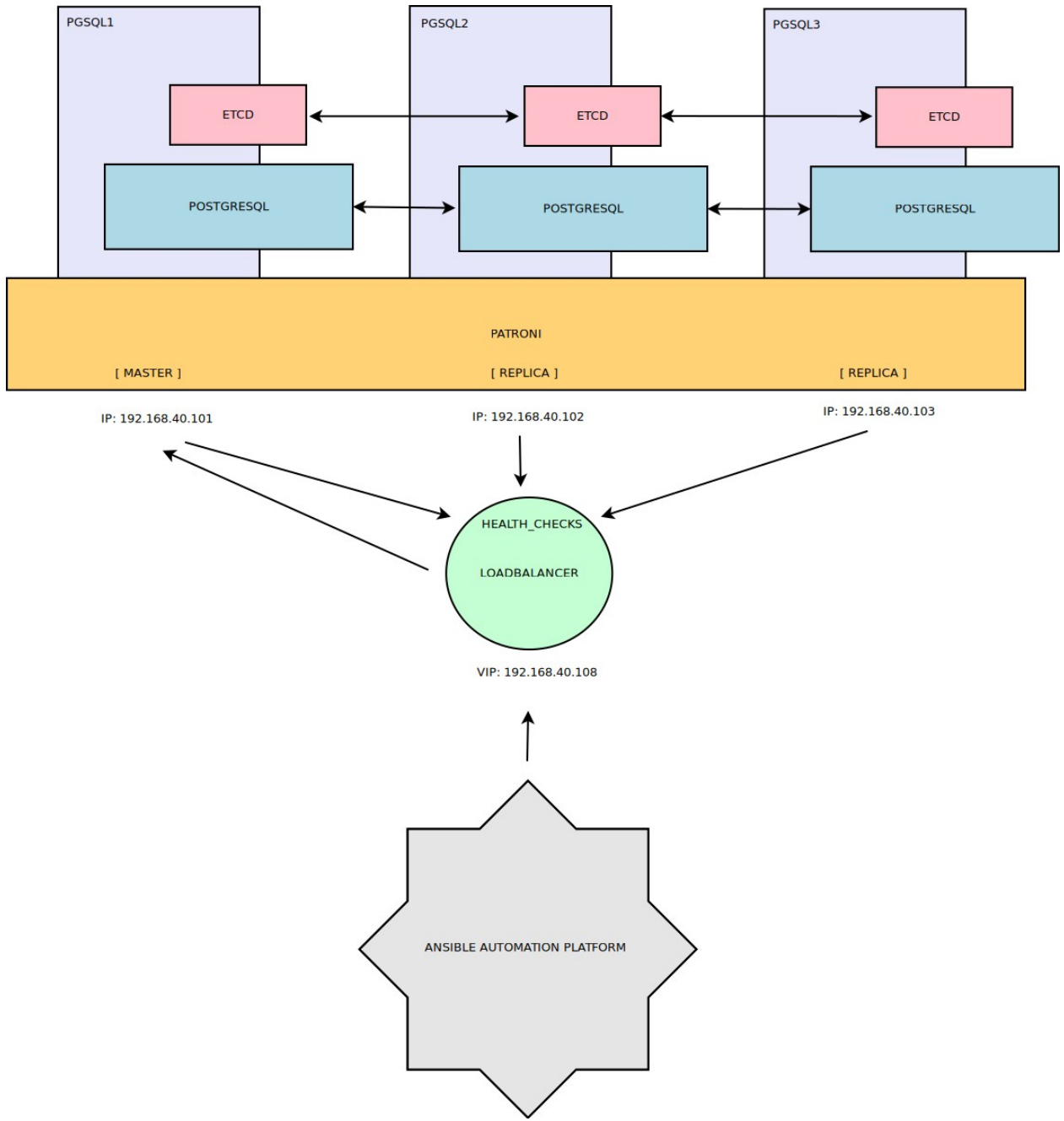
- External PostgreSQL (HA / DR)
- Customer provided PostgreSQL
- Cloud-hosted PostgreSQL
- 3rd party PostgreSQL offering
- Other major versions

* https://access.redhat.com/articles/4010491 *

What are our options?

Two of Many External HA Database Options

Percona PostgreSQL HA Cluster - Walkthrough


PGSQL1
PGSQL2
PGSQL3
ETCD
ETCD
ETCD
POSTGRESQL
POSTGRESQL
POSTGRESQL
PATRONI
[MASTER]
[REPLICA]
[REPLICA]
IP: 192.168.40.101
IP: 192.168.40.102
IP: 192.168.40.103
HEALTH_CHECKS
LOADBALANCER
VIP: 192.168.40.108
ANSIBLE AUTOMATION PLATFORM

Installation

Requirements

- Three nodes of RHEL 8.6 or 9.0
- Access to Percona repository (free or paid)

Node Details

- pgsql1.rhlab.skinnerlabs.com : 192.168.40.101
- pgsql2.rhlab.skinnerlabs.com : 192.168.40.102
- pgsql3.rhlab.skinnerlabs.com : 192.168.40.103

Installation

Watchdog Installation and Configuration

```
# yum -y install watchdog
# sh -c 'echo "KERNEL==\"watchdog\", OWNER=\"postgres\", GROUP=\"postgres\"" >> /etc/udev/rules.d/61-watchdog.rules'
# echo “softdog” > /etc/modules-load.d/softdog.conf
# systemctl enable watchdog
# REBOOT
# lsmod | grep softdog
```

Installation

Percona ETCD Installation

yum -y install https://repo.percona.com/yum/percona-release-latest.noarch.rpm

percona-release setup ppg13 OR # percona-release setup ppg14

yum -y install etcd python3-python-etcd

Installation

Percona ETCD Configuration (pgsql1,pgsql2,pgsql3)

- Edit /etc/etcd/etcd.conf **BOLD IS UNIQUE PER SYSTEM**

** SERVER **ETCD1** **

ETCD_DATA_DIR="/var/lib/etcd/default.etcd"

ETCD_LISTEN_PEER_URLS="http://0.0.0.0:2380"

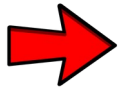
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"

ETCD_NAME="**ETCD1**"

ETCD_INITIAL_ADVERTISE_PEER_URLS="http://**pgsql1**.rhlab.skinnerlabs.com:2380"

ETCD_ADVERTISE_CLIENT_URLS="http://**pgsql1**.rhlab.skinnerlabs.com:2379"

ETCD_INITIAL_CLUSTER="ETCD1=http://pgsql1.rhlab.skinnerlabs.com:2380,ETCD2=http://pgsql2.rhlab.skinnerlabs.com:2380,ETCD3=http://pgsql3.rhlab.skinnerlabs.com:2380"



ETCD_INITIAL_CLUSTER_TOKEN="**etcd-cluster**"

Installation

Percona ETCD Testing

- Start ETCD on each node (pgsql1,pgsql2,pgsql3)

  # systemctl enable --now etcd

- From any node – show/test key value addition, update and removal

  # etcdctl member list

  # etcdctl cluster-health

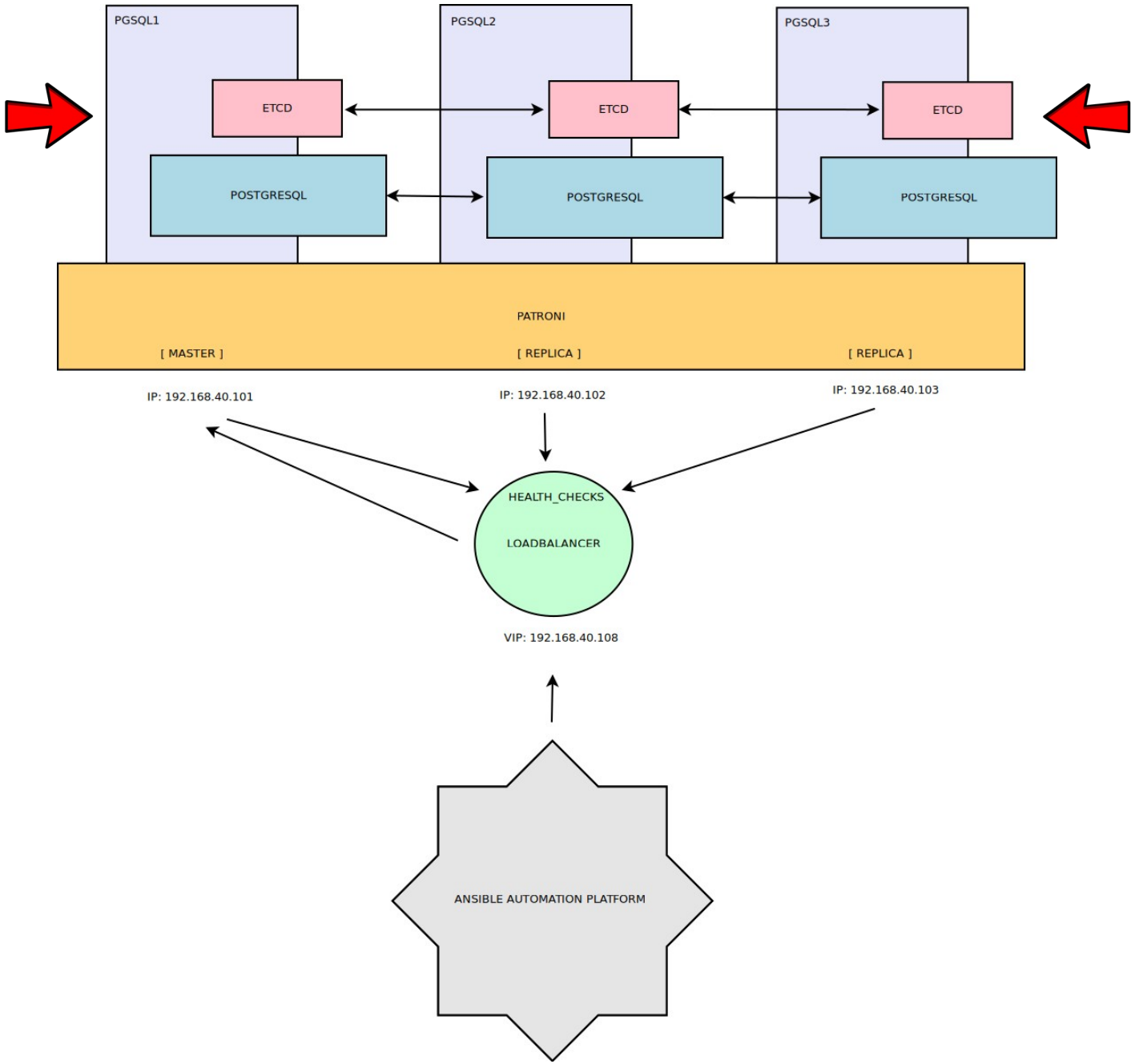
  # etcdctl set **test** "Testing..1,2,3!"

  # etcdctl get **test**

  # etcdctl update **test** "Testing..4,5,6!"

  # etcdctl get **test**

  # etcdctl rm **test**


PGSQL1
PGSQL2
PGSQL3
ETCD
ETCD
ETCD
POSTGRESQL
POSTGRESQL
POSTGRESQL
PATRONI
[MASTER]
[REPLICA]
[REPLICA]
IP: 192.168.40.101
IP: 192.168.40.102
IP: 192.168.40.103
HEALTH_CHECKS
LOADBALANCER
VIP: 192.168.40.108
ANSIBLE AUTOMATION PLATFORM

Installation

Percona PostgreSQL Installation (pgsql1,pgsql2,pgsql3)

- Version 13

  # yum -y install percona-pg_repack13 percona-pgaudit percona-pgbackrest percona-patroni percona-pgbouncer percona-pgaudit13_set_user percona-pgbadger percona-wal2json13 percona-postgresql13-contrib

- Or Version 14

  # yum -y install percona-pg_repack14 percona-pgaudit percona-pgbackrest percona-patroni percona-pgbouncer percona-pgaudit14_set_user percona-pgbadger percona-wal2json14 percona-postgresql14-contrib

Installation

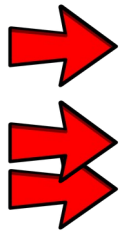
Percona Patroni Installation (pgsql1,pgsql2,pgsql3)

```
# python3 -m pip install patroni[etcd]
# mkdir -p /etc/patroni
# chown -R  postgres:postgres /etc/patroni
```

Installation

Percona Patroni Configuration (pgsql1,pgsql2,pgsql3)

- Create /etc/patroni/patroni.yml **BOLD IS UNIQUE PER SYSTEM**



```
scope: postgres
namespace: /pg_cluster/
name: pgsql1

restapi:
  listen: 192.168.40.101:8008                 # PostgreSQL node IP address
  connect_address: 192.168.40.101:8008        # PostgreSQL node IP address

etcd:
  hosts: 192.168.40.101:2379,192.168.40.102:2379,192.168.40.103:2379  # ETCD node IP addresses
```

Installation

Percona Patroni Configuration (pgsql1,pgsql2,pgsql3)

- Edit /etc/patroni/patroni.yml ... continued from previous

```
bootstrap:
  # this section will be written into Etcd:/<namespace>/<scope>/config after initializing new cluster
  dcs:
    ttl: 30
    loop_wait: 10
    retry_timeout: 10
    maximum_lag_on_failover: 1048576
    postgresql:
      use_pg_rewind: true
      use_slots: true
      parameters:
        wal_level: replica
        hot_standby: "on"
        logging_collector: 'on'
        max_wal_senders: 5
        max_replication_slots: 5
        wal_log_hints: "on"
```

Installation

Percona Patroni Configuration (pgsql1,pgsql2,pgsql3)

- Edit /etc/patroni/patroni.yml ... continued from previous

```
  # some desired options for 'initdb'
  initdb:  # Note: It needs to be a list (some options need values, others are switches)
  - encoding: UTF8
  - data-checksums

  pg_hba:  # Add following lines to pg_hba.conf after running 'initdb'
  - host replication replicator 127.0.0.1/32 md5
  - host replication replicator 192.168.40.101/32 md5
  - host replication replicator 192.168.40.102/32 md5
  - host replication replicator 192.168.40.103/32 md5
  - host all all 0.0.0.0/0 md5
#  - hostssl all all 0.0.0.0/0 md5
```

Installation

Percona Patroni Configuration (pgsql1,pgsql2,pgsql3)

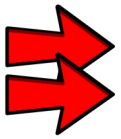
- Edit /etc/patroni/patroni.yml … continued from previous

```
# Some additional users users which needs to be created after initializing new cluster
users:
  admin:
    password: admin
    options:
      - createrole
      - createdb
```

Installation

Percona Patroni Configuration (pgsql1,pgsql2,pgsql3)

- Edit /etc/patroni/patroni.yml **BOLD IS UNIQUE PER SYSTEM** … continued from previous



```
postgresql:
  listen: 192.168.40.101:5432              # PostgreSQL node IP address
  connect_address: 192.168.40.101:5432     # PostgreSQL node IP address
  data_dir: /var/lib/pgsql/13/data         # The datadir you created depending on version
  bin_dir: /usr/pgsql-13/bin
  pgpass: /tmp/pgpass0
  authentication:
    replication:
      username: replicator
      password: replicator
    superuser:
      username: postgres
      password: postgres
  parameters:
    unix_socket_directories: '.'
```

Installation

Percona Patroni Configuration (pgsql1,pgsql2,pgsql3)

- Edit /etc/patroni/patroni.yml … continued from previous

```
tags:
    nofailover: false
    noloadbalance: false
    clonefrom: false
    nosync: false
```

Installation

Percona Patroni Configuration (pgsql1,pgsql2,pgsql3)

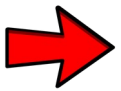
- Create systemd unit file /etc/patroni/patroni.yml

```
[Unit]
Description=Runners to orchestrate a high-availability PostgreSQL
After=syslog.target network.target

[Service]
Type=simple

User=postgres
Group=postgres

# Start the patroni process
ExecStart=/bin/patroni /etc/patroni/patroni.yml

# Send HUP to reload from patroni.yml
ExecReload=/bin/kill -s HUP $MAINPID
```

Installation

Percona Patroni Configuration (pgsql1,pgsql2,pgsql3)

- Create systemd unit file /etc/patroni/patroni.yml … continued from previous



```
# only kill the patroni process, not its children, so it will gracefully stop postgres
KillMode=process

# Give a reasonable amount of time for the server to start up/shut down
TimeoutSec=30

# Do not restart the service if it crashes, we want to manually inspect database on failure
Restart=no

[Install]
WantedBy=multi-user.target
```

Installation

Percona Patroni Start and Testing

- Start

  # systemctl daemon-reload

- On first run, it will initdb and set up replication

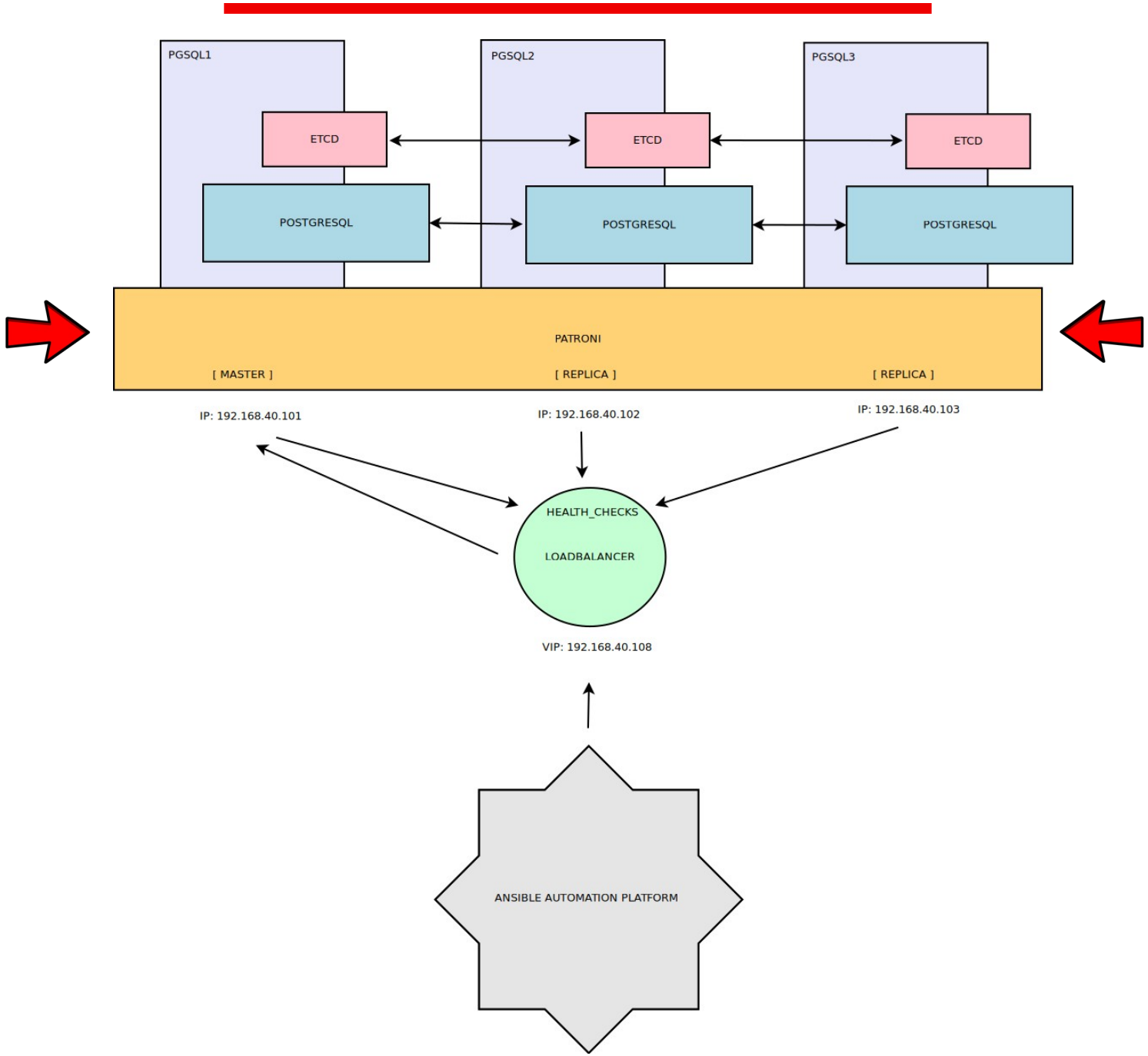
  # systemctl enable --now patroni

- Testing

  # patronictl -c /etc/patroni/patroni.yml list

  # patronictl -c /etc/patroni/patroni.yml switchover

  # patronictl -c /etc/patroni/patroni.yml topology


PGSQL1
PGSQL2
PGSQL3
ETCD
ETCD
ETCD
POSTGRESQL
POSTGRESQL
POSTGRESQL
PATRONI
[MASTER]
[REPLICA]
[REPLICA]
IP: 192.168.40.101
IP: 192.168.40.102
IP: 192.168.40.103
HEALTH_CHECKS
LOADBALANCER
VIP: 192.168.40.108
ANSIBLE AUTOMATION PLATFORM

Red Hat

Load Balancer Options

Common Load Balancer Options

Load Balancers

- F5 Virtual Appliance
- HAProxy
- Many, many more ...

Why a Load Balancer?

- Direct traffic to single write server
- Auto fail to new write server
- Manipulate traffic flow

HAProxy

Installation

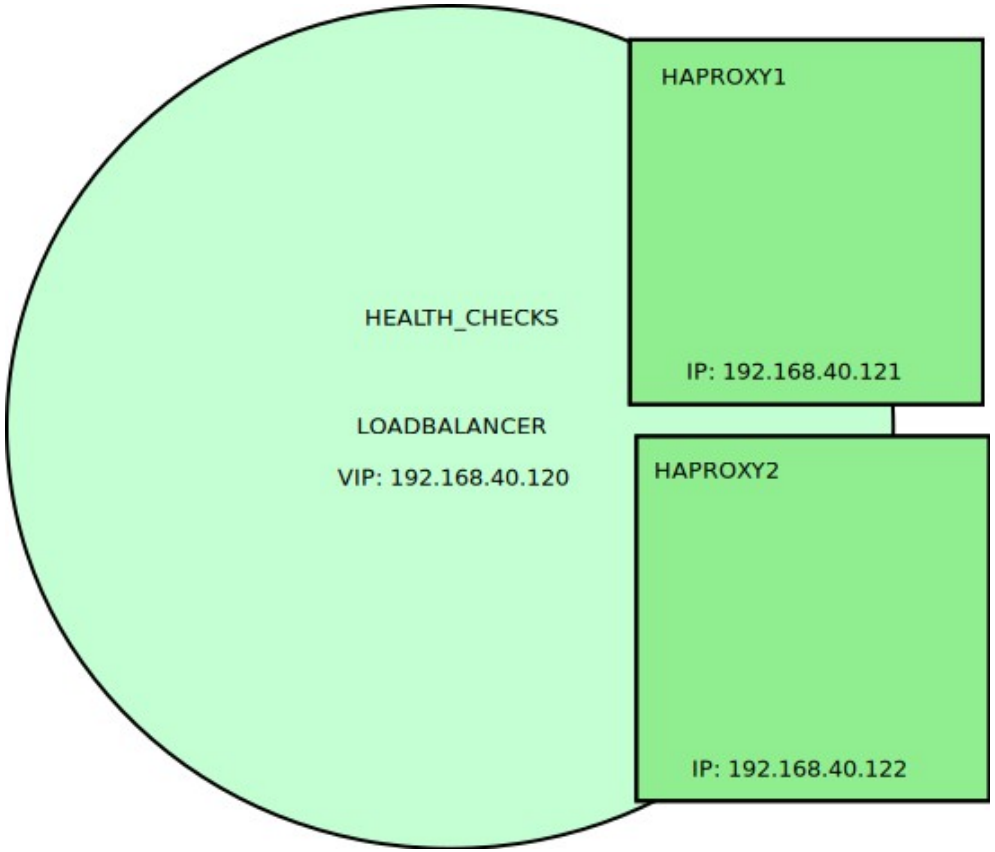
Requirements

- Two nodes of RHEL 8.6 or 9.0
- Floating IP address:

  haproxy-vip.rhlab.skinnerlabs.com
  192.168.40.120

Node Details

- haproxy1.rhlab.skinnerlabs.com :
  192.168.40.121
- haproxy2.rhlab.skinnerlabs.com :
  192.168.40.122


HAPROXY1
IP: 192.168.40.121
HEALTH_CHECKS
LOADBALANCER
VIP: 192.168.40.120
HAPROXY2
IP: 192.168.40.122

Red Hat

Installation

HAProxy Installation (haproxy1,haproxy2)

yum -y install haproxy keepalived

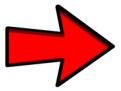
- Enable ability to bind to a non-local IP Address for failover – the VIP
- Create /etc/sysctl.d/haproxy.conf

```
net.ipv4.ip_nonlocal_bind = 1
```

Installation

HAProxy Configuration (haproxy1,haproxy2)

- Create /etc/haproxy/haproxy.conf

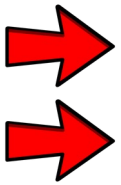


```
global
    maxconn 100

defaults
    log global
    mode tcp
    retries 2
    timeout client 30m
    timeout connect 4s
    timeout server 30m
    timeout check 5s

listen stats
    mode http
    bind *:7000
    stats enable
    stats uri /
```

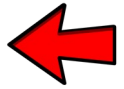
Installation

HAProxy Configuration (haproxy1,haproxy2)

- Create /etc/haproxy/haproxy.conf … continued from previous



```
listen primary
    bind *:5432
    option httpchk /primary
    http-check expect status 200
    default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
    server pgsql1 192.168.40.101:5432 maxconn 100 check port 8008
    server pgsql2 192.168.40.102:5432 maxconn 100 check port 8008
    server pgsql3 192.168.40.103:5432 maxconn 100 check port 8008
```



Installation

KeepAliveD Configuration (haproxy1,haproxy2) BOLD IS UNIQUE PER SYSTEM

- Create /etc/keepalived/keepalived.conf

```
vrrp_script chk_haproxy {              # Requires keepalived-1.1.13
        script "killall -0 haproxy"    # cheaper than pidof
        interval 2                     # check every 2 seconds
        weight 2                       # add 2 points of prio if OK
}

vrrp_instance VI_1 {
        interface ens3
        state MASTER
        virtual_router_id 51
        priority 101                   # 101 on master, 100 on backup
        virtual_ipaddress {
            192.168.40.120
        }
        track_script {
            chk_haproxy
        }
}
```

Installation

HAProxy Start and Testing

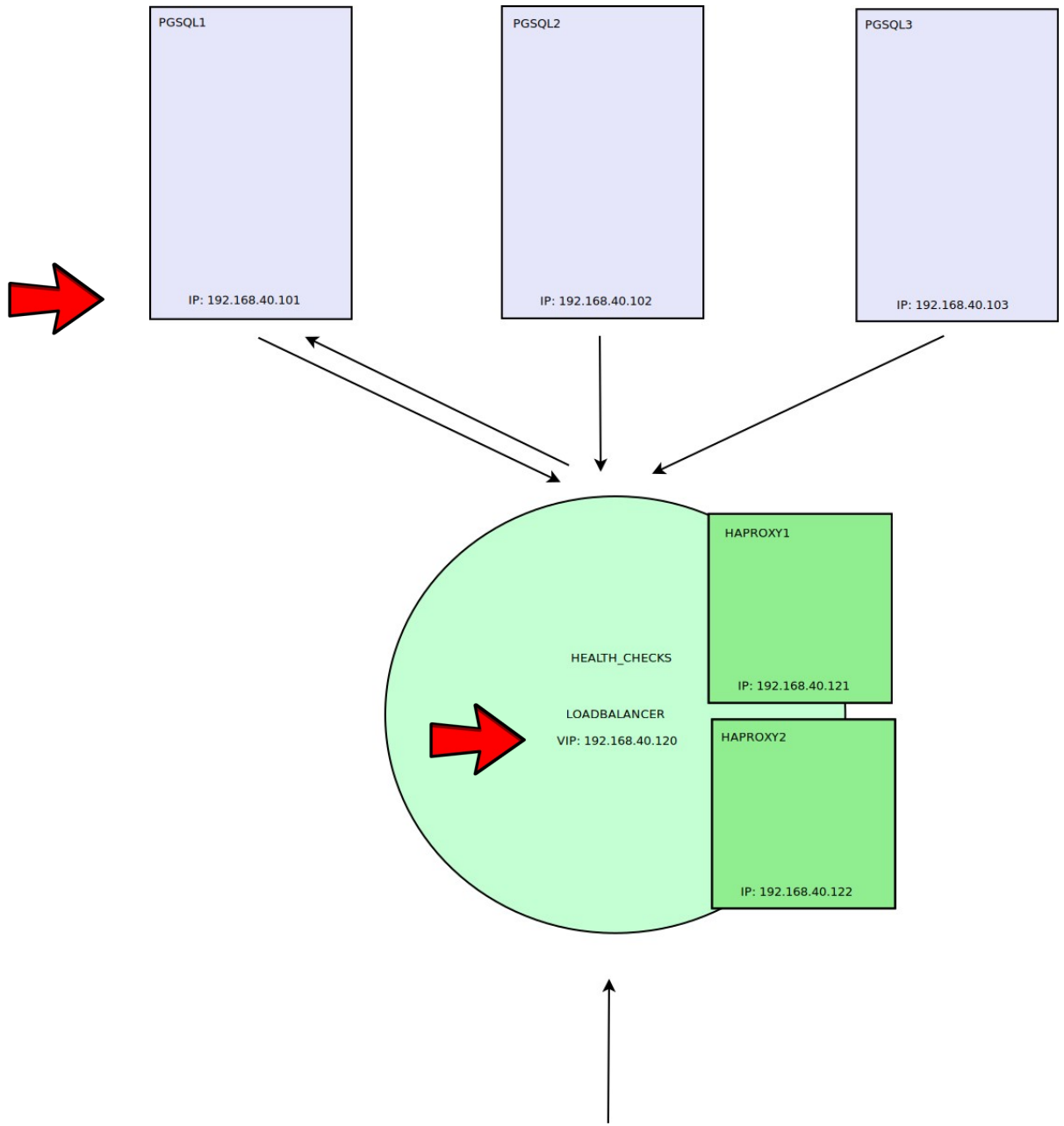
- Start

  # systemctl enable --now haproxy

  # systemctl enable --now keepalived

- Testing – where is the VIP?

  # ip addr | grep 192.168.40.120


PGSQL1
IP: 192.168.40.101
PGSQL2
IP: 192.168.40.102
PGSQL3
IP: 192.168.40.103
HAPROXY1
IP: 192.168.40.121
HEALTH_CHECKS
LOADBALANCER
VIP: 192.168.40.120
HAPROXY2
IP: 192.168.40.122

Red Hat

HAProxy version 1.8.27-493ce0b, released 2020/11/06

Statistics Report for pid 984

> General process information

pid = 984 (process #1, nbproc = 1, nbthread = 1)
uptime = 5d 18h11m26s
system limits: memmax = unlimited; ulimit-n = 218
maxsock = 218; **maxconn =** 100; **maxpipes =** 0
current conns = 1; current pipes = 0/0; conn rate = 1/sec
Running tasks: 1/8; idle = 100 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance
backup UP
backup UP, going down
backup DOWN, going up
not checked
Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:
- Scope :
- Hide 'DOWN' servers
- Refresh now
- CSV export

External resources:
- Primary site
- Updates (v1.8)
- Online manual

stats

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				1	1	-	1	1	2 000	3			1 043	51 206	0	0	0					OPEN								
Backend	0	0		0	0		0	0	200	0	0	0s	1 043	51 206	0	0		0	0	0	0	5d18h UP		0	0	0		0		

primary

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	0	-	0	0	2 000	0			0	0	0	0	0					OPEN								
pgsql1	0	0	-	0	0		0	0	100	0	0	?	0	0		0		0	0	0	0	14s UP	L7OK/200 in 2ms	1	Y	-	1	1	5d18h	-
pgsql2	0	0	-	0	0		0	0	100	0	0	?	0	0		0		0	0	0	0	11s DOWN	L7STS/503 in 1ms	1	Y	-	4	2	5d17h	-
pgsql3	0	0	-	0	0		0	0	100	0	0	?	0	0		0		0	0	0	0	5d18h DOWN	L7STS/503 in 1ms	1	Y	-	1	1	5d18h	-
Backend	0	0		0	0		0	0	200	0	0	?	0	0	0	0		0	0	0	0	16m6s UP		1	1	0		1	5d17h	

HAProxy Status

- HAProxy provides status on port 7000
- pgsql1 is write server – so Patroni API returns success to HAProxy health check

HAProxy version 1.8.27-493ce0b, released 2020/11/06

Statistics Report for pid 984

> General process information

pid = 984 (process #1, nbproc = 1, nbthread = 1)
uptime = 5d 17h57m03s
system limits: memmax = unlimited; ulimit-n = 218
maxsock = 218; **maxconn** = 100; **maxpipes** = 0
current conns = 1; current pipes = 0/0; conn rate = 1/sec
Running tasks: 1/8; idle = 100 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance
backup UP
backup UP, going down
backup DOWN, going up
not checked
Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:
- Scope :
- Hide 'DOWN' servers
- Refresh now
- CSV export

External resources:
- Primary site
- Updates (v1.8)
- Online manual

stats

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server							
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				1	1	-	1	1	2 000	2			350	16 912	0	0	0					OPEN							
Backend	0	0		0	0		0	0	200	0	0	0s	350	16 912	0	0		0	0	0	0	5d17h UP		0	0	0		0		

primary

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server							
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	0	-	0	0	2 000	0			0	0	0	0	0					OPEN							
pgsql1	0	0	-	0	0		0	0	100	0	0	?	0	0		0		0	0	0	0	5d17h DOWN	L7STS/503 in 1ms	1	Y	-	1	1	5d17h	-
pgsql2	0	0	-	0	0		0	0	100	0	0	?	0	0		0		0	0	0	0	1m43s UP	L7OK/200 in 2ms	1	Y	-	1	1	5d17h	-
pgsql3	0	0	-	0	0		0	0	100	0	0	?	0	0		0		0	0	0	0	5d17h DOWN	L7STS/503 in 1ms	1	Y	-	1	1	5d17h	-
Backend	0	0		0	0		0	0	200	0	0	?	0	0	0	0		0	0	0	0	1m43s UP		1	1	0		1	5d17h	

HAProxy Testing

- Shut down pgsql1
- Patroni API updates HAProxy health check
- pgsql2 is now write server

F5 Virtual Appliance

Installation

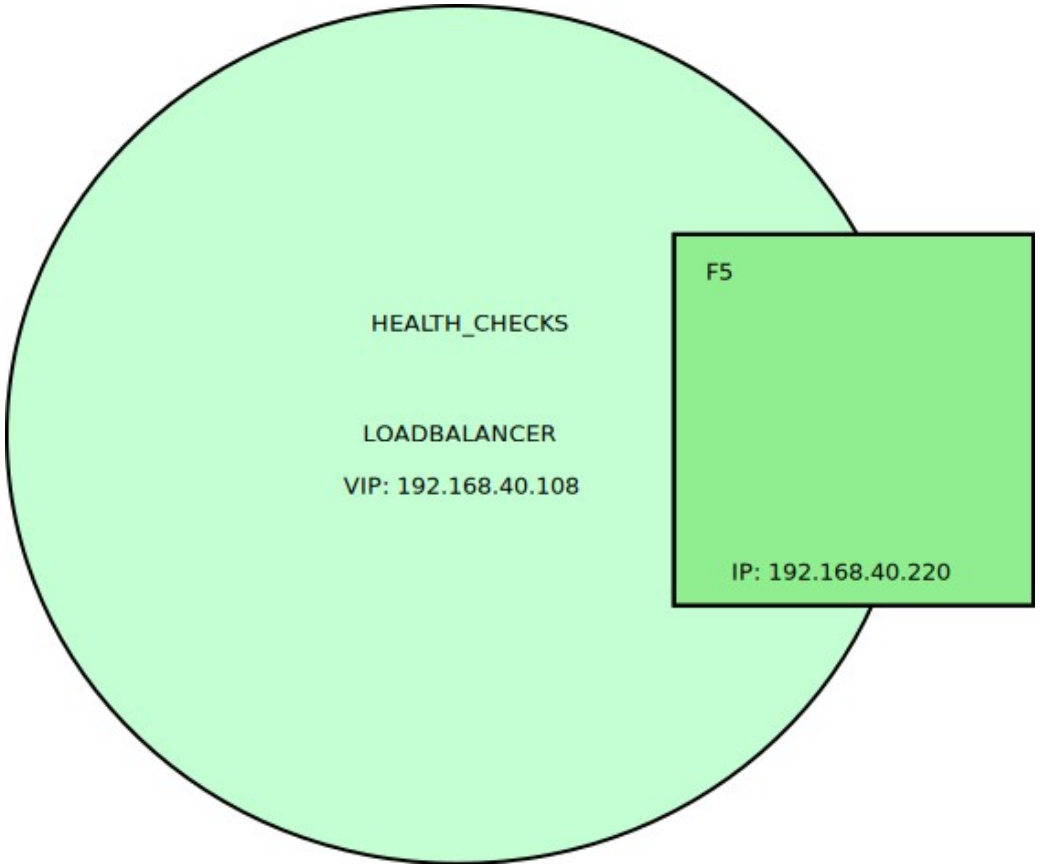
Requirements

- One F5 Virtual Appliance
- Lab License ($160)
  - Limited to 10Mb/s
- Floating IP address:

  pgsql-vip.rhlab.skinnerlabs.com
  192.168.40.108

Node Details

- f5.rhlab.skinnerlabs.com :
  192.168.40.220


HEALTH_CHECKS
LOADBALANCER
VIP: 192.168.40.108
F5
IP: 192.168.40.220

Red Hat

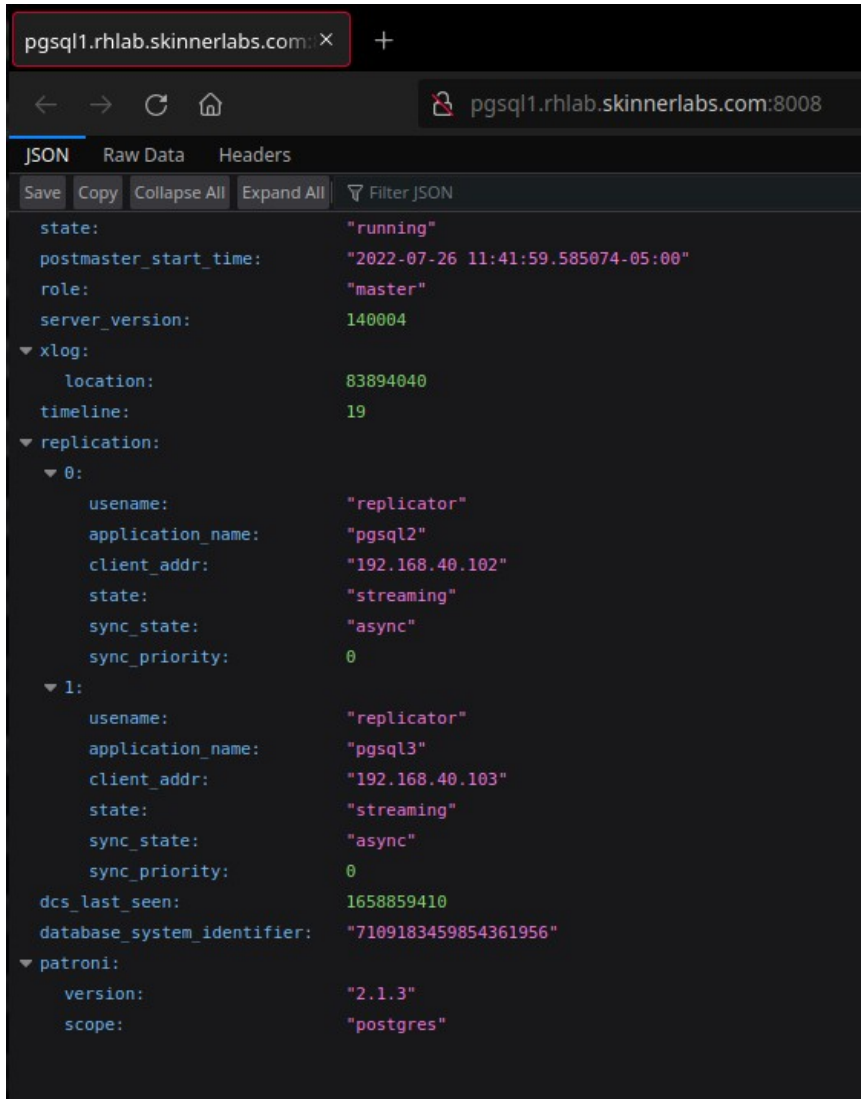
F5 Configuration

Monitor (Health Check)

- Create ‘PostgreSQL-monitor’

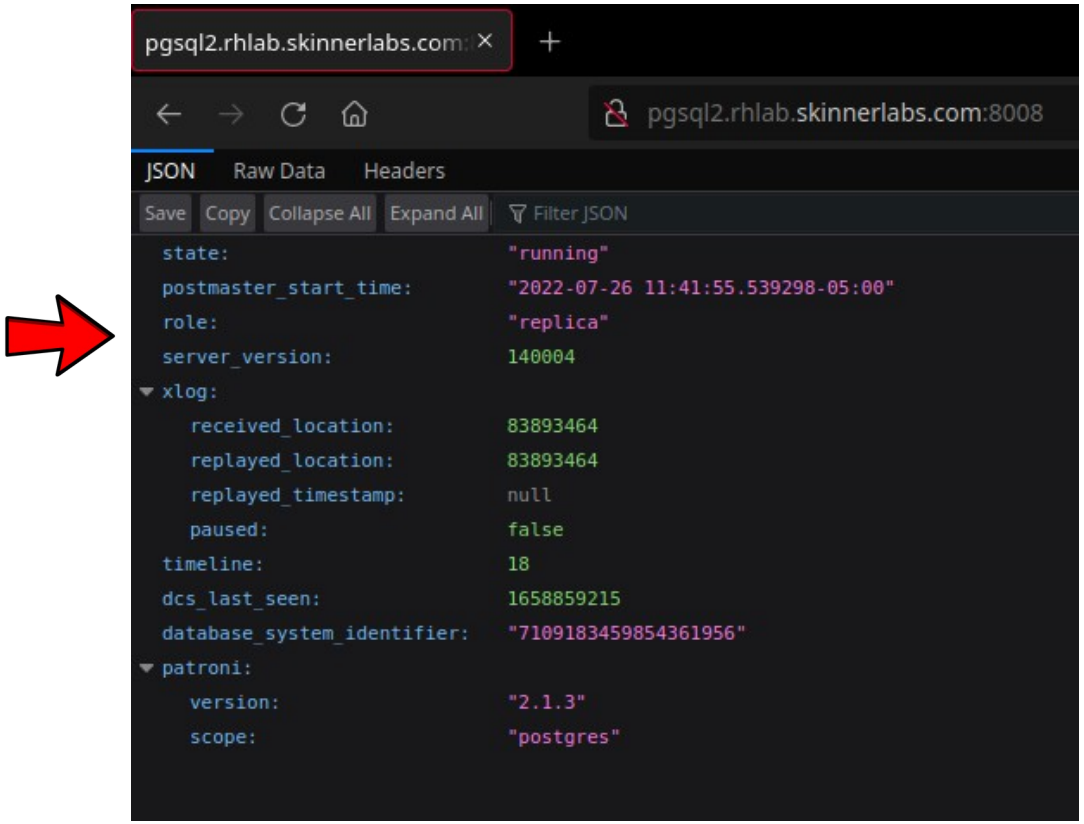
Monitor Details

- Type: HTTP
- Receive String: \”role\”: \”master\”
- Alias Service Port: 8008

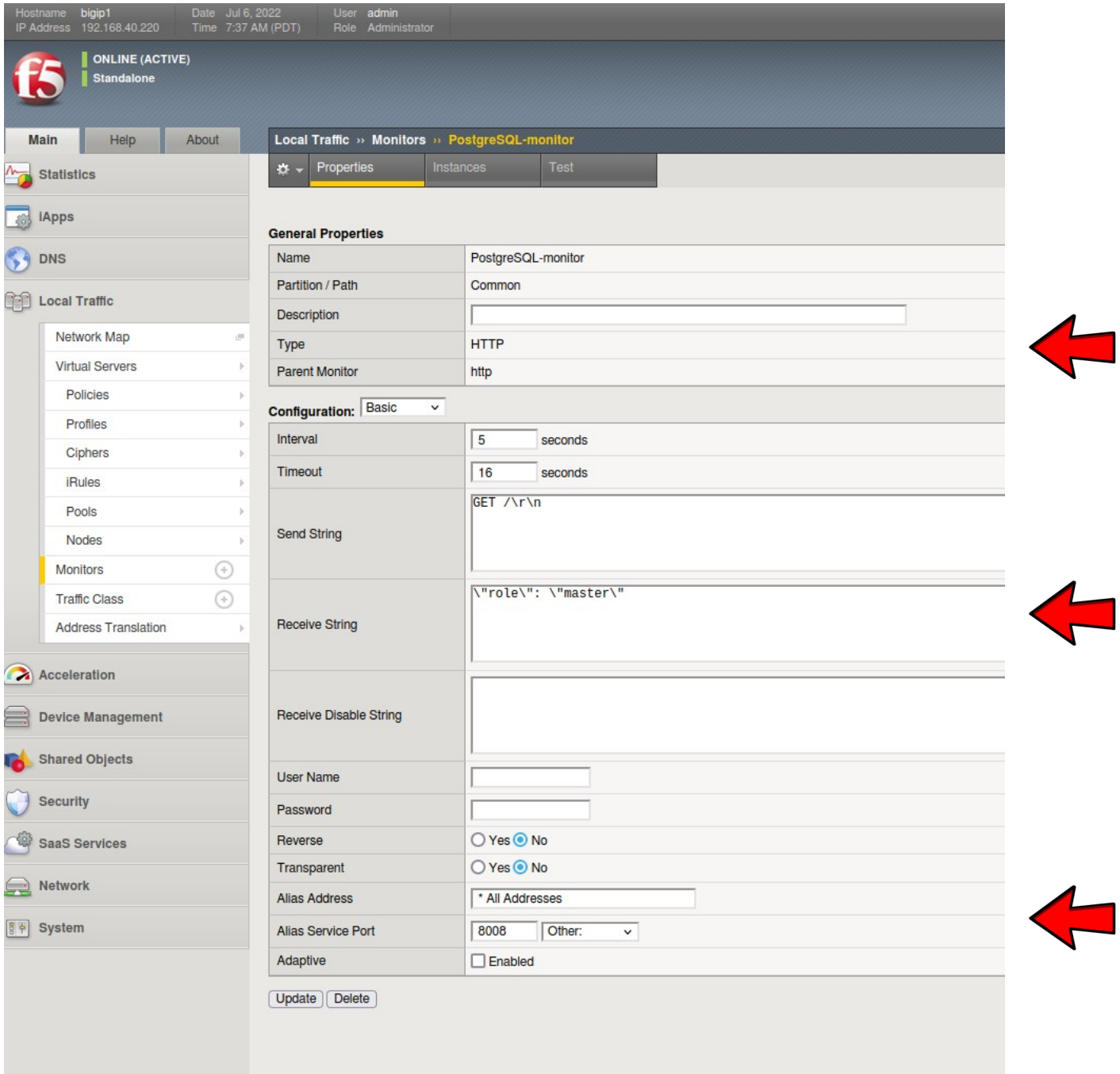


What is the role status?

- Check Patroni API on port 8008



Red Hat


Hostname bigip1
IP Address 192.168.40.220
Date Jul 6, 2022
Time 7:37 AM (PDT)
User admin
Role Administrator
ONLINE (ACTIVE)
Standalone
Main
Help
About
Local Traffic ›› Monitors ›› PostgreSQL-monitor
Statistics
iApps
DNS
Local Traffic
Network Map
Virtual Servers
Policies
Profiles
Ciphers
iRules
Pools
Nodes
Monitors
Traffic Class
Address Translation
Acceleration
Device Management
Shared Objects
Security
SaaS Services
Network
System
Properties
Instances
Test
General Properties
Name PostgreSQL-monitor
Partition / Path Common
Description
Type HTTP
Parent Monitor http
Configuration: Basic
Interval 5 seconds
Timeout 16 seconds
Send String GET /\r\n
Receive String \"role\": \"master\"
Receive Disable String
User Name
Password
Reverse Yes No
Transparent Yes No
Alias Address * All Addresses
Alias Service Port 8008 Other:
Adaptive Enabled
Update
Delete

Red Hat

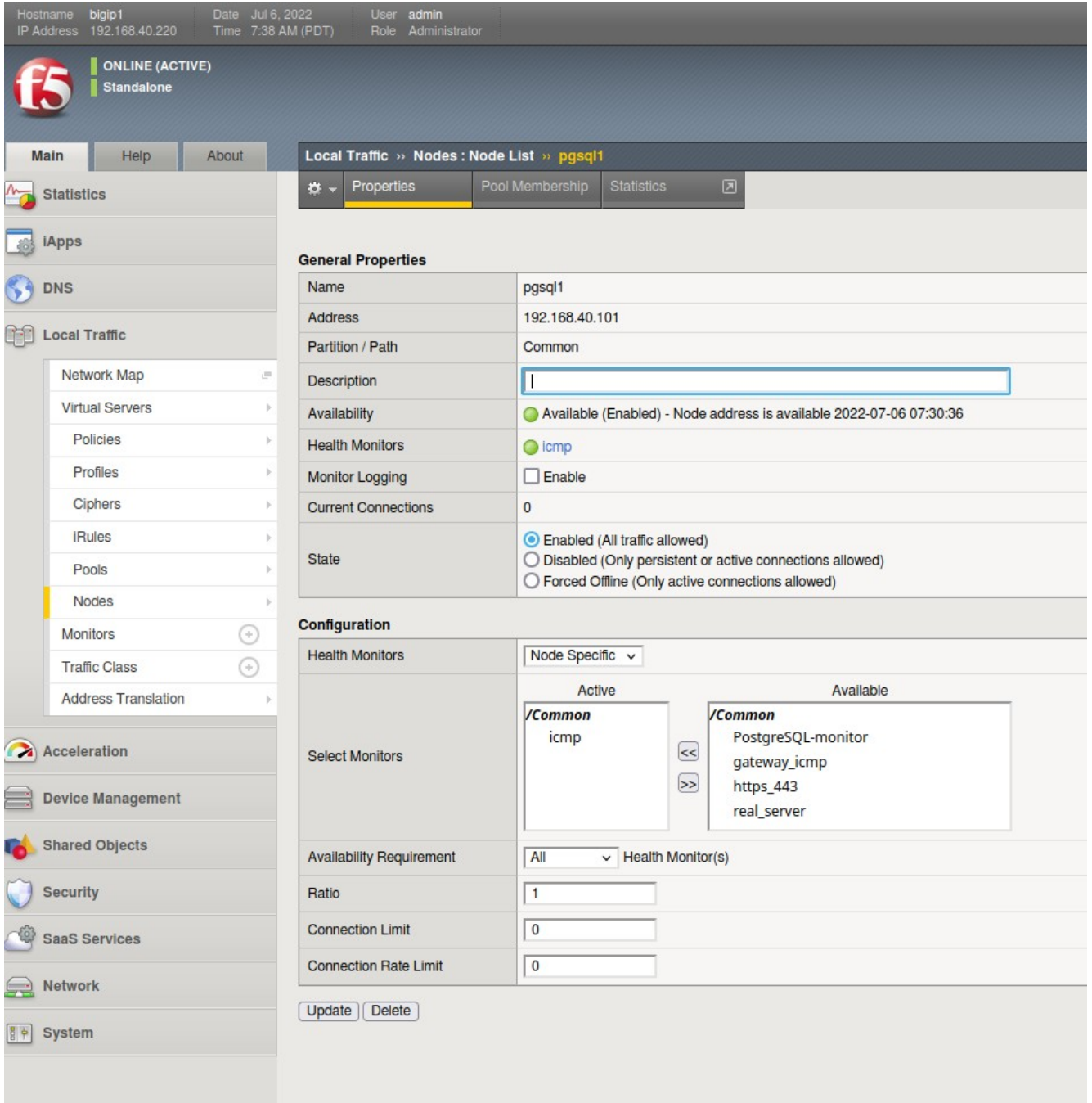
F5 Configuration

Create Nodes

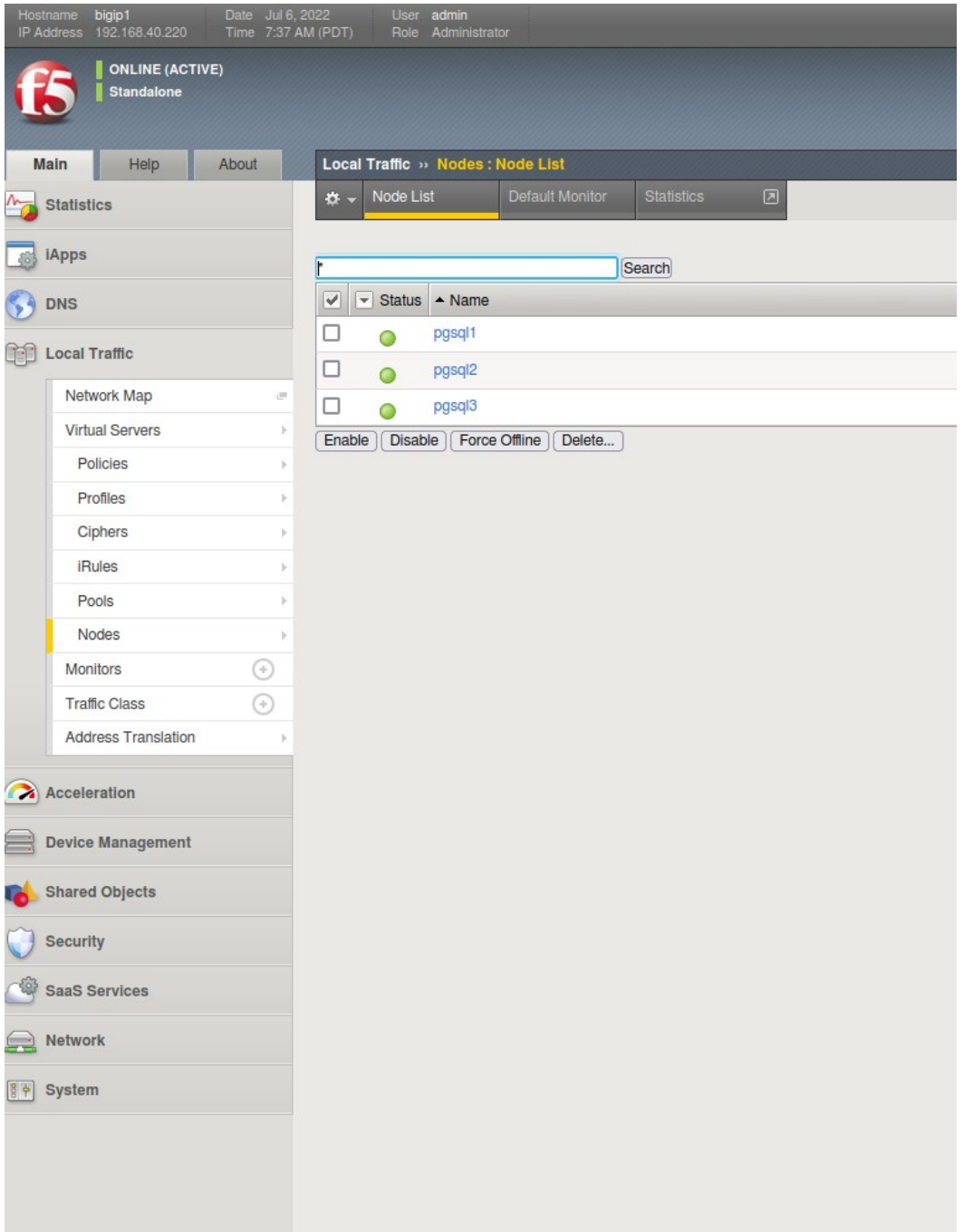
- Create ‘pgsql1’, ‘pgsql2’, ’pgsql3’

Node Details

- Node Name
- Node IP Address
- Node Health Monitor
  - ICMP


Hostname bigip1
IP Address 192.168.40.220
Date Jul 6, 2022
Time 7:38 AM (PDT)
User admin
Role Administrator
ONLINE (ACTIVE)
Standalone
Main
Help
About
Statistics
iApps
DNS
Local Traffic
Network Map
Virtual Servers
Policies
Profiles
Ciphers
iRules
Pools
Nodes
Monitors
Traffic Class
Address Translation
Acceleration
Device Management
Shared Objects
Security
SaaS Services
Network
System
Local Traffic ›› Nodes : Node List ›› pgsql1
Properties
Pool Membership
Statistics
General Properties
Name pgsql1
Address 192.168.40.101
Partition / Path Common
Description
Availability Available (Enabled) - Node address is available 2022-07-06 07:30:36
Health Monitors icmp
Monitor Logging Enable
Current Connections 0
State Enabled (All traffic allowed)
Disabled (Only persistent or active connections allowed)
Forced Offline (Only active connections allowed)
Configuration
Health Monitors Node Specific
Active
Available
/Common
icmp
/Common
PostgreSQL-monitor
gateway_icmp
https_443
real_server
Select Monitors
Availability Requirement All Health Monitor(s)
Ratio 1
Connection Limit 0
Connection Rate Limit 0
Update
Delete

Red Hat


Hostname bigip1
IP Address 192.168.40.220
Date Jul 6, 2022
Time 7:37 AM (PDT)
User admin
Role Administrator
ONLINE (ACTIVE)
Standalone
Main
Help
About
Local Traffic ›› Nodes : Node List
Statistics
iApps
DNS
Local Traffic
Network Map
Virtual Servers
Policies
Profiles
Ciphers
iRules
Pools
Nodes
Monitors
Traffic Class
Address Translation
Acceleration
Device Management
Shared Objects
Security
SaaS Services
Network
System
Node List
Default Monitor
Statistics
Search
Status
Name
pgsql1
pgsql2
pgsql3
Enable
Disable
Force Offline
Delete...

Red Hat

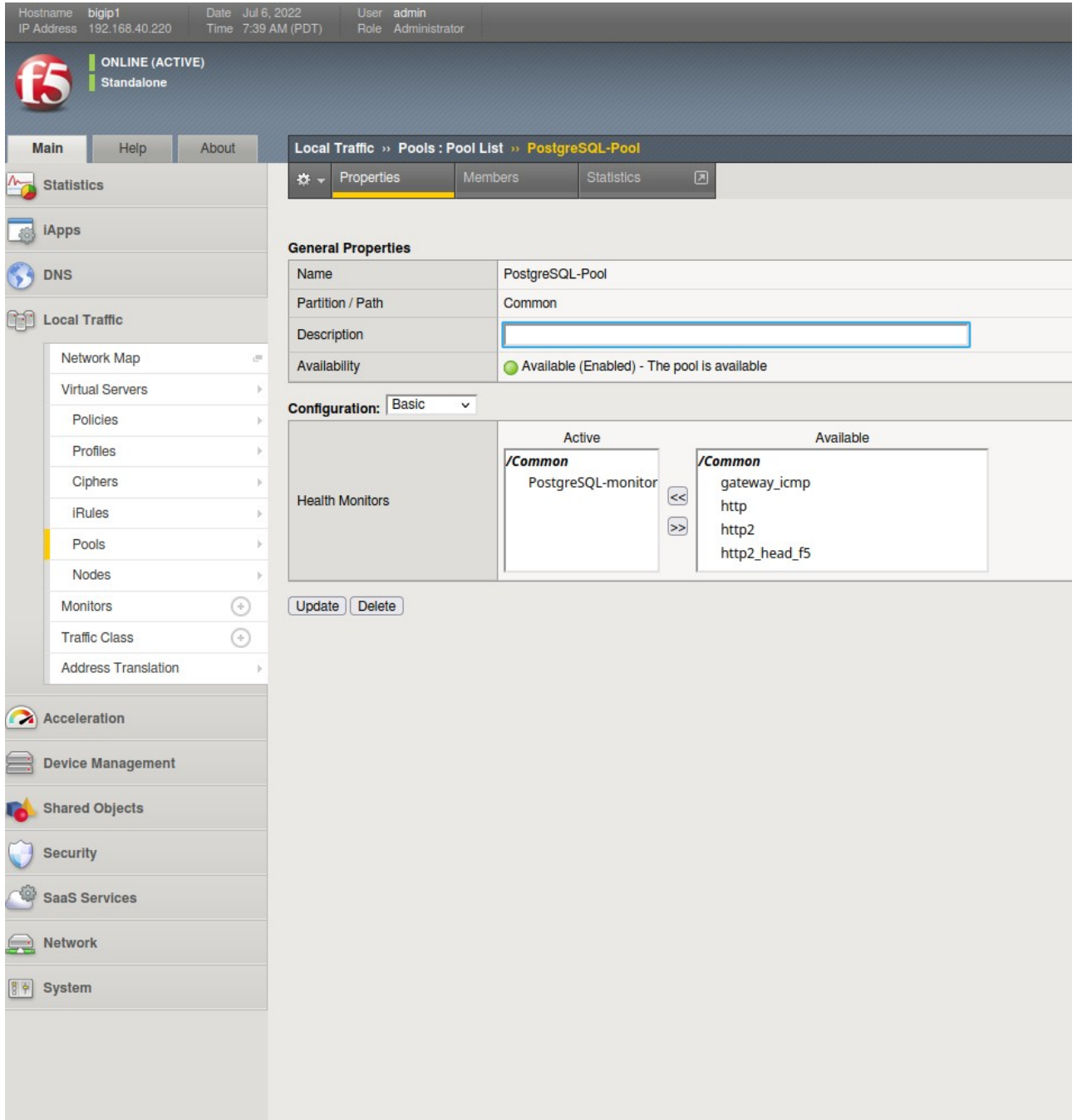
F5 Configuration

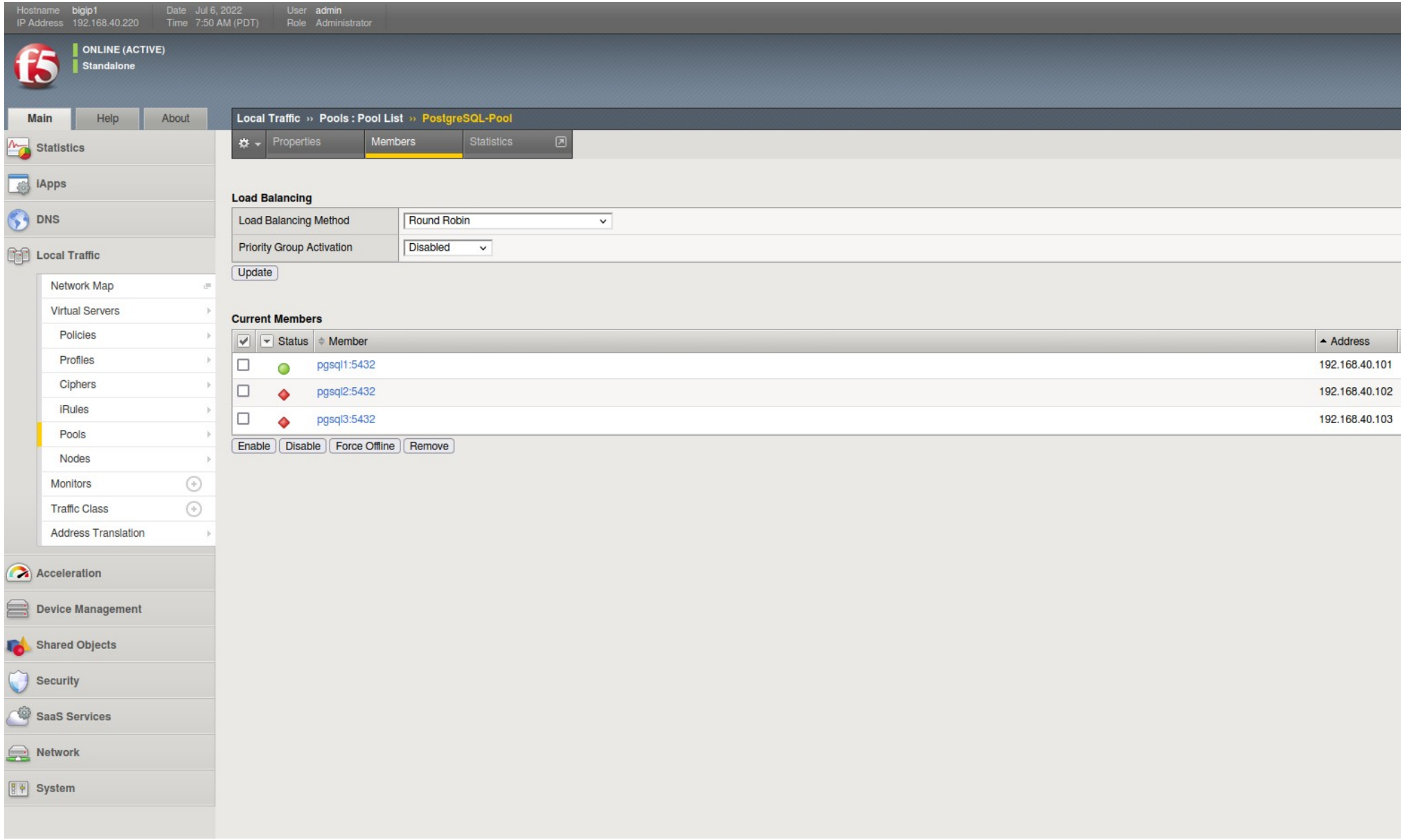
Create Pool

- Create ‘PostgreSQL-Pool’

Pool Details

- Assign all Node members
- Assign PostgreSQL-monitor as Health Monitor


Hostname bigip1
IP Address 192.168.40.220
Date Jul 6, 2022
Time 7:39 AM (PDT)
User admin
Role Administrator
ONLINE (ACTIVE)
Standalone
Main
Help
About
Local Traffic ›› Pools : Pool List ›› PostgreSQL-Pool
Properties
Members
Statistics
Statistics
iApps
DNS
Local Traffic
Network Map
Virtual Servers
Policies
Profiles
Ciphers
iRules
Pools
Nodes
Monitors
Traffic Class
Address Translation
Acceleration
Device Management
Shared Objects
Security
SaaS Services
Network
System
General Properties
Name
PostgreSQL-Pool
Partition / Path
Common
Description
Availability
Available (Enabled) - The pool is available
Configuration: Basic
Health Monitors
Active
/Common
PostgreSQL-monitor
<<
>>
Available
/Common
gateway_icmp
http
http2
http2_head_f5
Update
Delete


Hostname bigip1
IP Address 192.168.40.220
Date Jul 6, 2022
Time 7:50 AM (PDT)
User admin
Role Administrator
ONLINE (ACTIVE)
Standalone
Main
Help
About
Statistics
iApps
DNS
Local Traffic
Network Map
Virtual Servers
Policies
Profiles
Ciphers
iRules
Pools
Nodes
Monitors
Traffic Class
Address Translation
Acceleration
Device Management
Shared Objects
Security
SaaS Services
Network
System
Local Traffic ›› Pools : Pool List ›› PostgreSQL-Pool
Properties
Members
Statistics
Load Balancing
Load Balancing Method
Round Robin
Priority Group Activation
Disabled
Update
Current Members
Status
Member
Address
pgsql1:5432
192.168.40.101
pgsql2:5432
192.168.40.102
pgsql3:5432
192.168.40.103
Enable
Disable
Force Offline
Remove

Red Hat

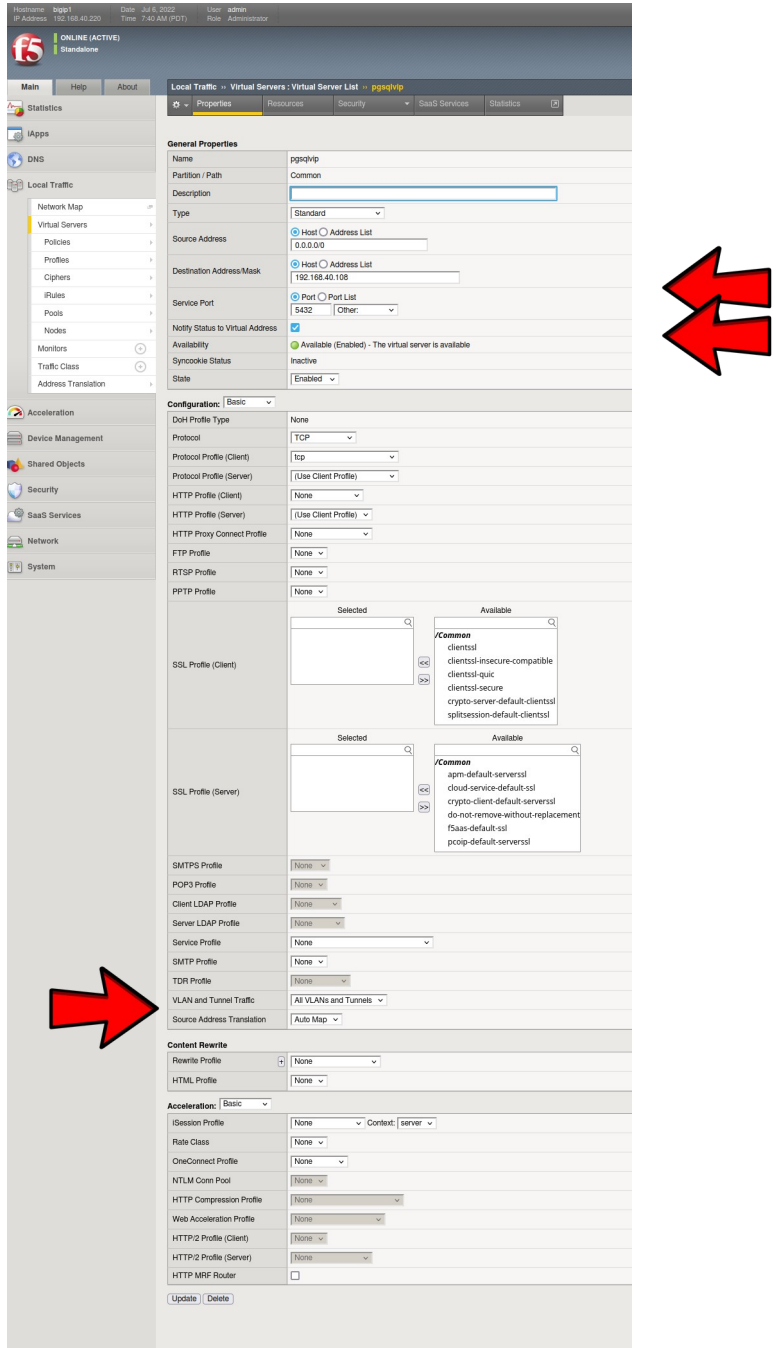
F5 Configuration

Create Virtual Server (VIP)

- Create ‘pgsqlvip’

Virtual Server Details

- Destination Address
- Service Port: 5432
- Source Address Translation: Auto Map

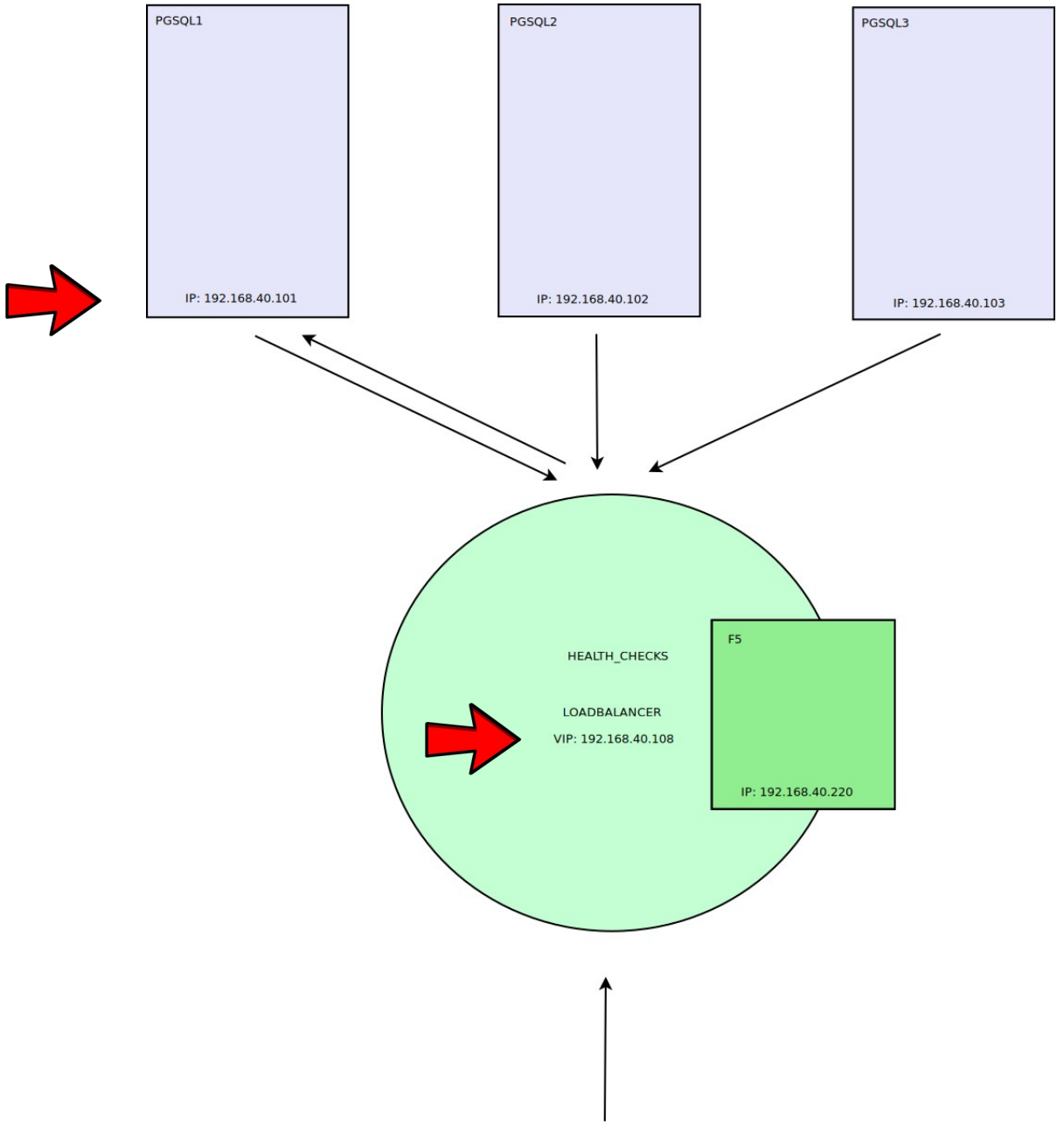

Hostname bigip1
IP Address 192.168.40.220
Date Jul 6, 2022
Time 7:40 AM (PDT)
User admin
Role Administrator
ONLINE (ACTIVE)
Standalone
Main
Help
About
Local Traffic ›› Virtual Servers : Virtual Server List ›› pgsqlvip
Properties
Resources
Security
SaaS Services
Statistics
Statistics
iApps
DNS
Local Traffic
Network Map
Virtual Servers
Policies
Profiles
Ciphers
iRules
Pools
Nodes
Monitors
Traffic Class
Address Translation
Acceleration
Device Management
Shared Objects
Security
SaaS Services
Network
System
General Properties
Name
pgsqlvip
Partition / Path
Common
Description
Type
Standard
Source Address
Host
Address List
0.0.0.0/0
Destination Address/Mask
Host
Address List
192.168.40.108
Service Port
Port
Port List
5432
Other:
Notify Status to Virtual Address
Availability
Available (Enabled) - The virtual server is available
Syncookie Status
Inactive
State
Enabled
Configuration:
Basic
DoH Profile Type
None
Protocol
TCP
Protocol Profile (Client)
tcp
Protocol Profile (Server)
(Use Client Profile)
HTTP Profile (Client)
None
HTTP Profile (Server)
(Use Client Profile)
HTTP Proxy Connect Profile
None
FTP Profile
None
RTSP Profile
None
PPTP Profile
None
SSL Profile (Client)
Selected
Available
/Common
clientssl
clientssl-insecure-compatible
clientssl-quic
clientssl-secure
crypto-server-default-clientssl
splitsession-default-clientssl
SSL Profile (Server)
Selected
Available
/Common
apm-default-serverssl
cloud-service-default-ssl
crypto-client-default-serverssl
do-not-remove-without-replacement
f5aas-default-ssl
pcoip-default-serverssl
SMTPS Profile
None
POP3 Profile
None
Client LDAP Profile
None
Server LDAP Profile
None
Service Profile
None
SMTP Profile
None
TDR Profile
None
VLAN and Tunnel Traffic
All VLANs and Tunnels
Source Address Translation
Auto Map
Content Rewrite
Rewrite Profile
None
HTML Profile
None
Acceleration:
Basic
iSession Profile
None
Context:
server
Rate Class
None
OneConnect Profile
None
NTLM Conn Pool
None
HTTP Compression Profile
None
Web Acceleration Profile
None
HTTP/2 Profile (Client)
None
HTTP/2 Profile (Server)
None
HTTP MRF Router
Update
Delete

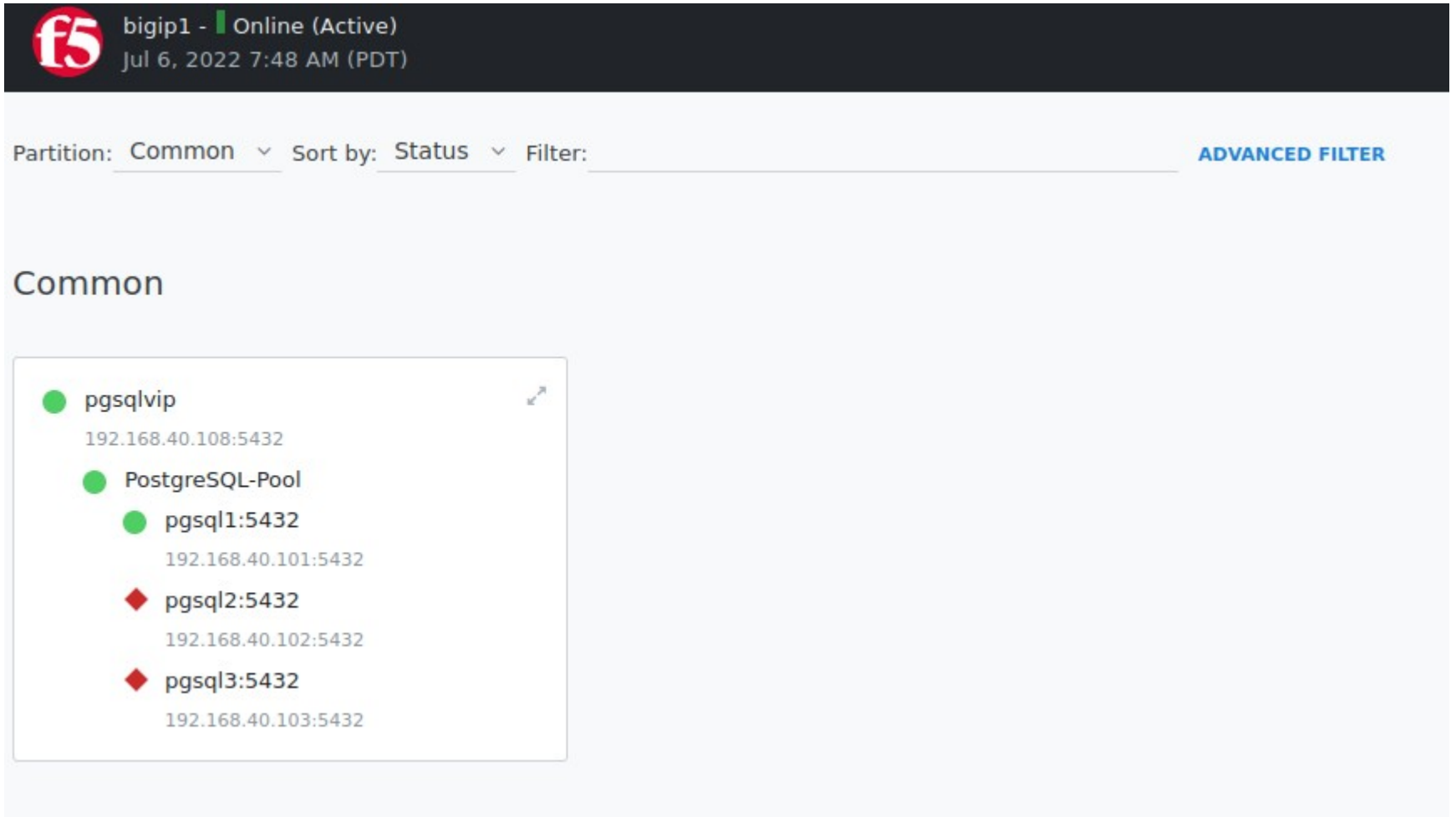
Red Hat

F5 Testing

F5 Status

- pgsql1 is write server – so Patroni API returns success to F5 health check


PGSQL1
IP: 192.168.40.101
PGSQL2
IP: 192.168.40.102
PGSQL3
IP: 192.168.40.103
HEALTH_CHECKS
LOADBALANCER
VIP: 192.168.40.108
F5
IP: 192.168.40.220


bigip1 - Online (Active)
Jul 6, 2022 7:48 AM (PDT)
Partition: Common
Sort by: Status
Filter:
ADVANCED FILTER
Common
pgsqlvip
192.168.40.108:5432
PostgreSQL-Pool
pgsql1:5432
192.168.40.101:5432
pgsql2:5432
192.168.40.102:5432
pgsql3:5432
192.168.40.103:5432

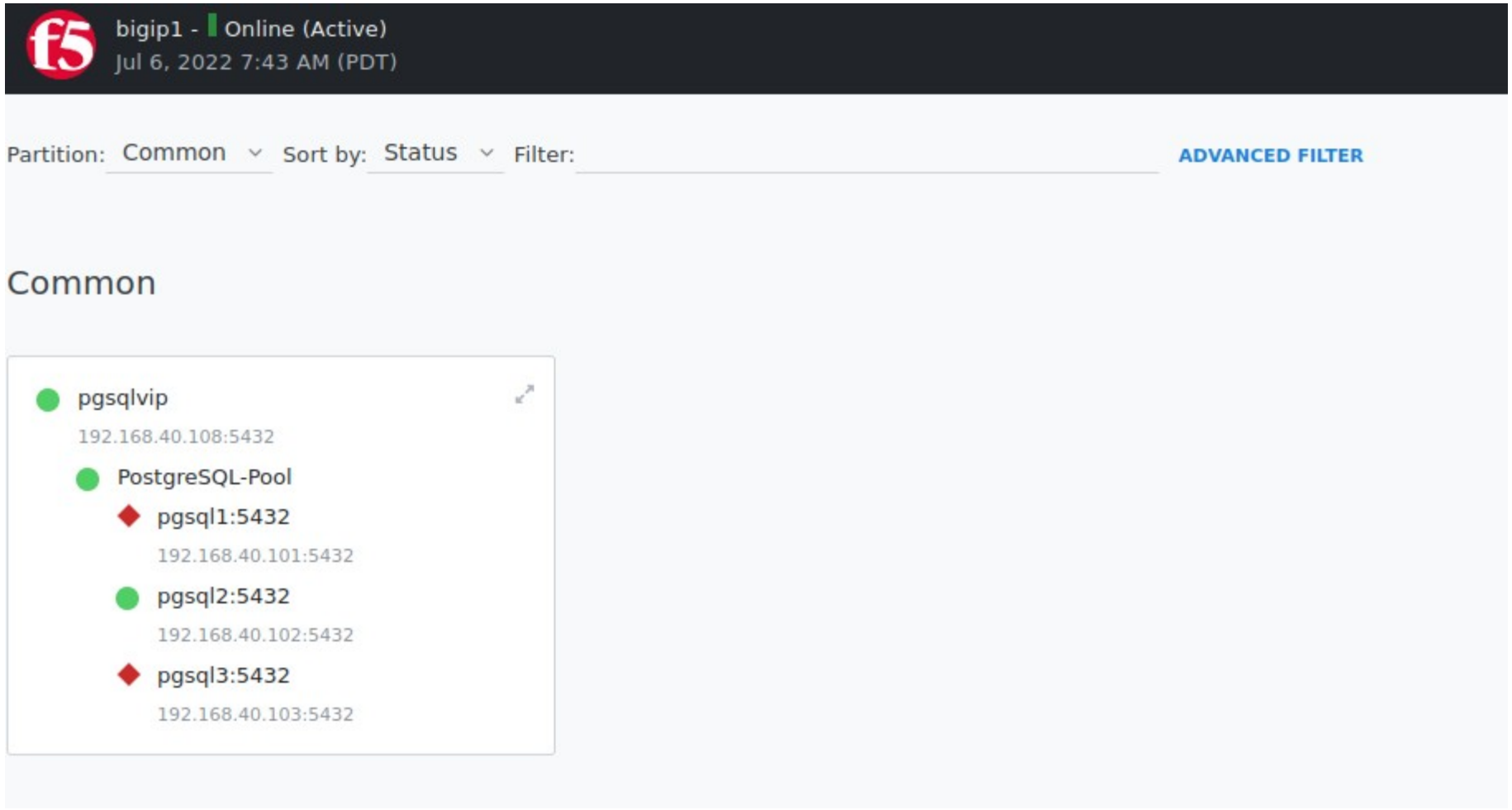
F5 Testing

Create Failure

- Shut down pgsql1

F5 Re-directs

- Patroni API updates F5 health check
- pgsql2 is now write server


bigip1 - Online (Active)
Jul 6, 2022 7:43 AM (PDT)
Partition: Common
Sort by: Status
Filter:
ADVANCED FILTER
Common
pgsqlvip
192.168.40.108:5432
PostgreSQL-Pool
pgsql1:5432
192.168.40.101:5432
pgsql2:5432
192.168.40.102:5432
pgsql3:5432
192.168.40.103:5432

Red Hat

Install AAP on PostgreSQL HA

Using a HA PostgreSQL

Test PostgreSQL HA VIP

- HAProxy

  # psql -h haproxy-vip.rhlab.skinnerlabs.com

- F5

  # psql -h pgsql-vip.rhlab.skinnerlabs.com

Edit installation inventory file – AAP install

[database]

pgsql-vip.rhlab.skinnerlabs.com

[all:vars]

pg_password='password'

pg_host='pgsql-vip.rhlab.skinnerlabs.com'

pg_port='5432'

pg_database='awx'

pg_username='awx'

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat