OVHcloud

Security and traceability on distributed database systems

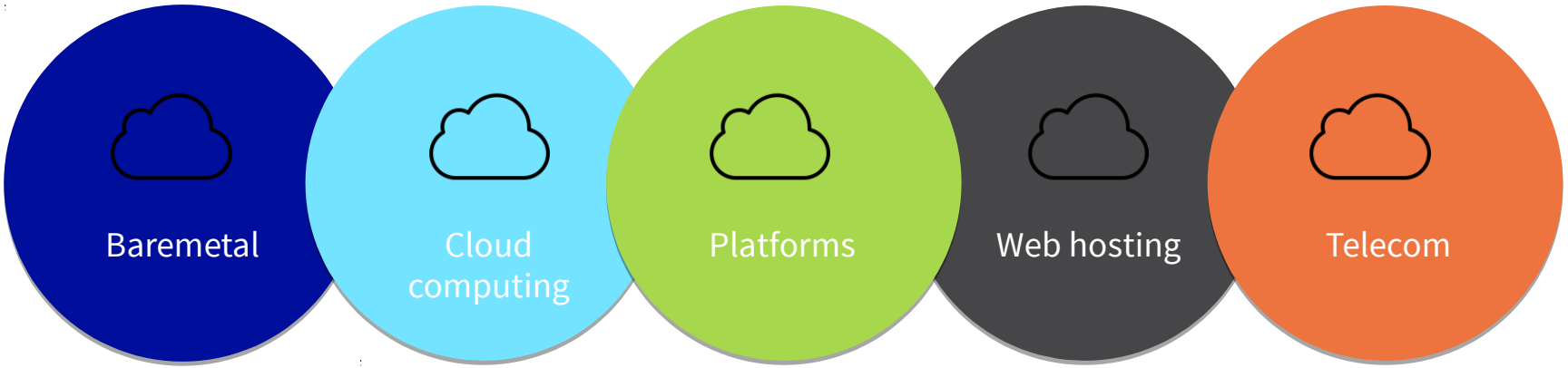
Julien Riou
Percona Live Online
October 21, 2020

Speaker

- Julien Riou
- DBA since 2012
- Tech lead in the databases team at OVHcloud since 2015
- https://julien.riou.xyz/


OVHcloud

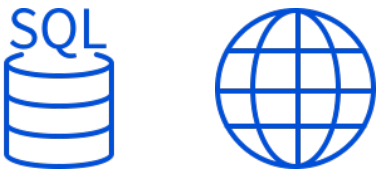
Critical services



They all rely on **internal databases** for control plane!

Internal databases

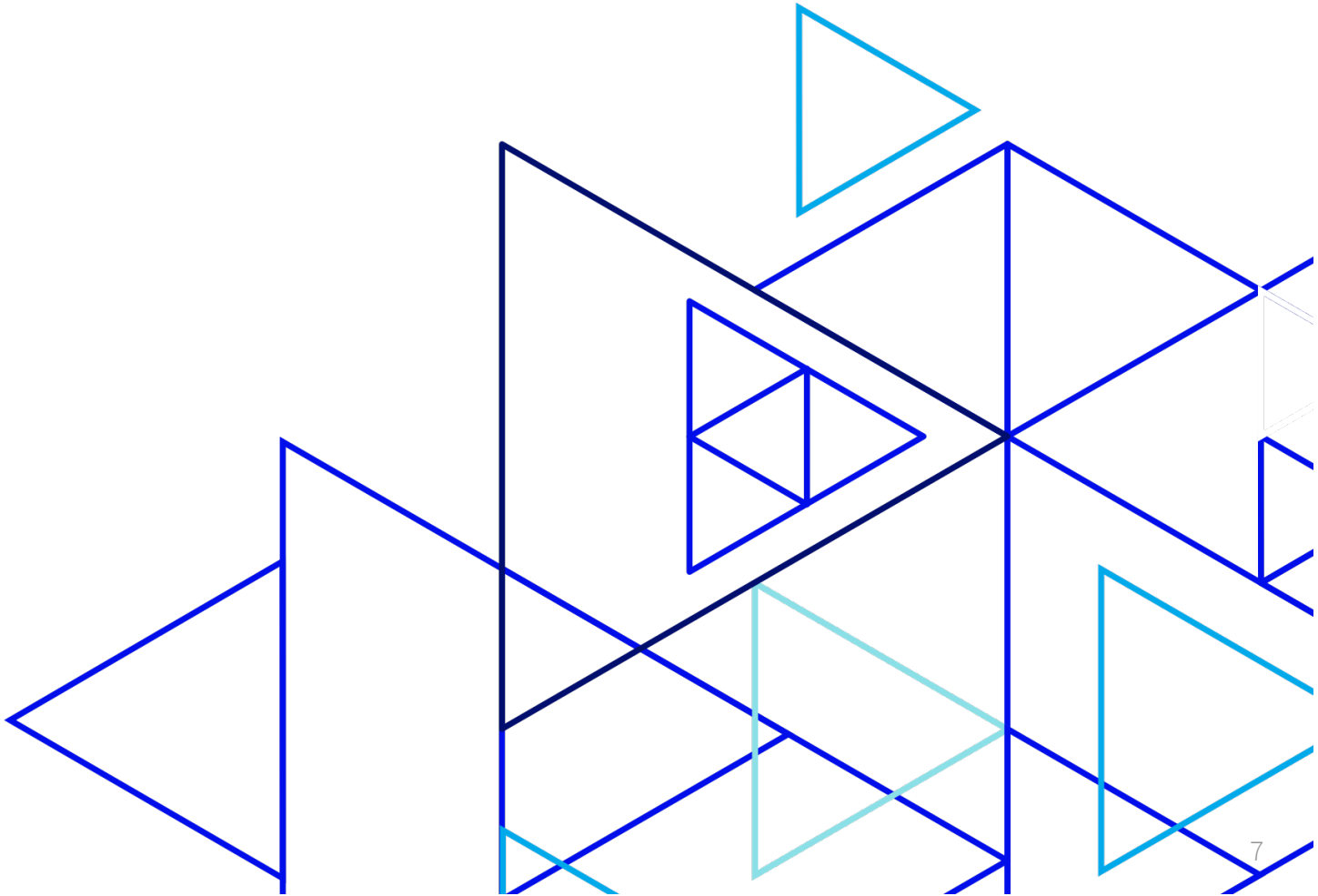
- 60 clusters
- MySQL and PostgreSQL
- 3000 applications
- 700 users
- 500 databases
- Worldwide



Compliance

- PCI DSS
- SecNumCloud
- And more

Overview



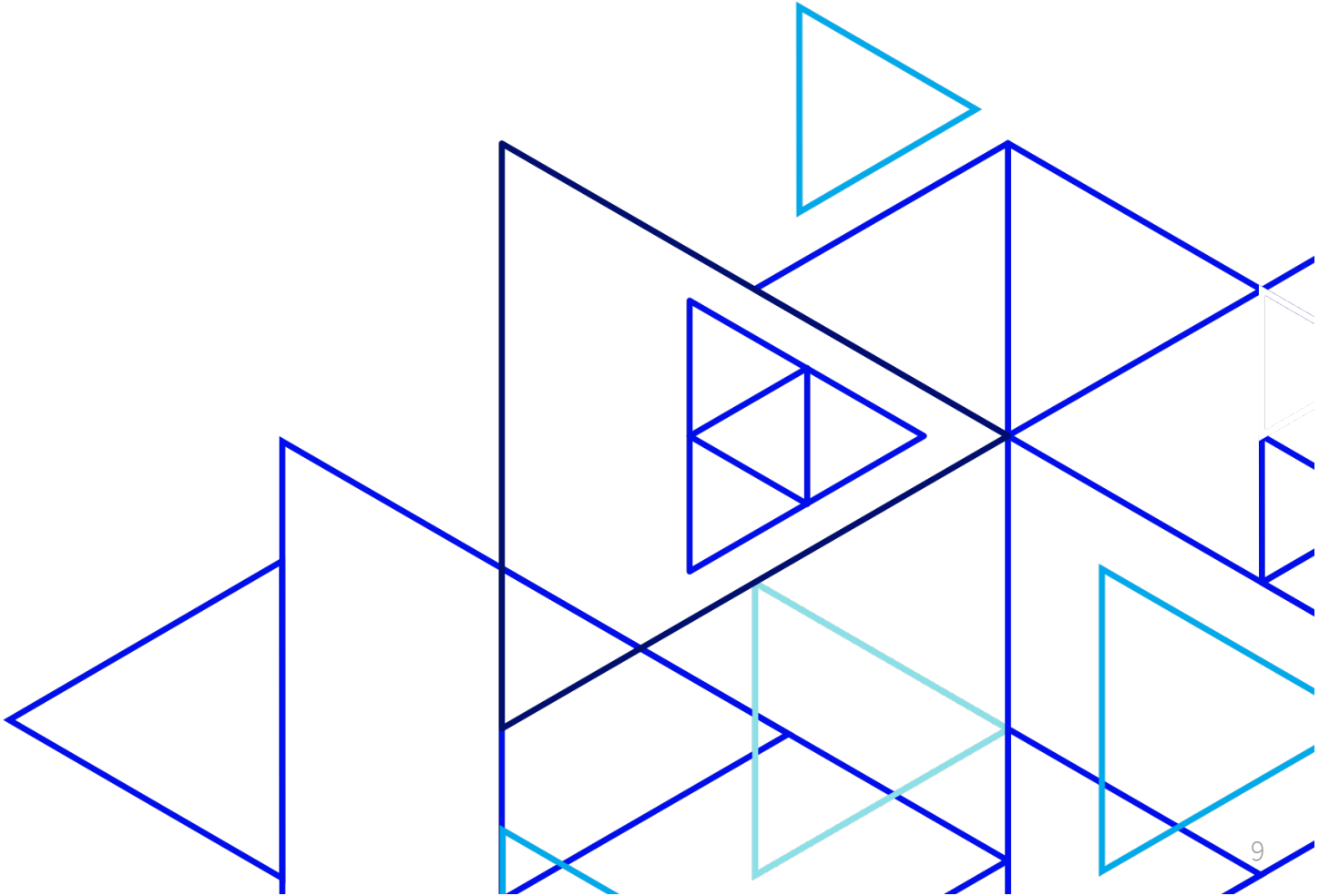
Percona Live Online
October 21, 2020

OVHcloud

Overview

- Context
- Security
- Traceability
- PostgreSQL solutions
- Generic solutions
- Conclusion

Context

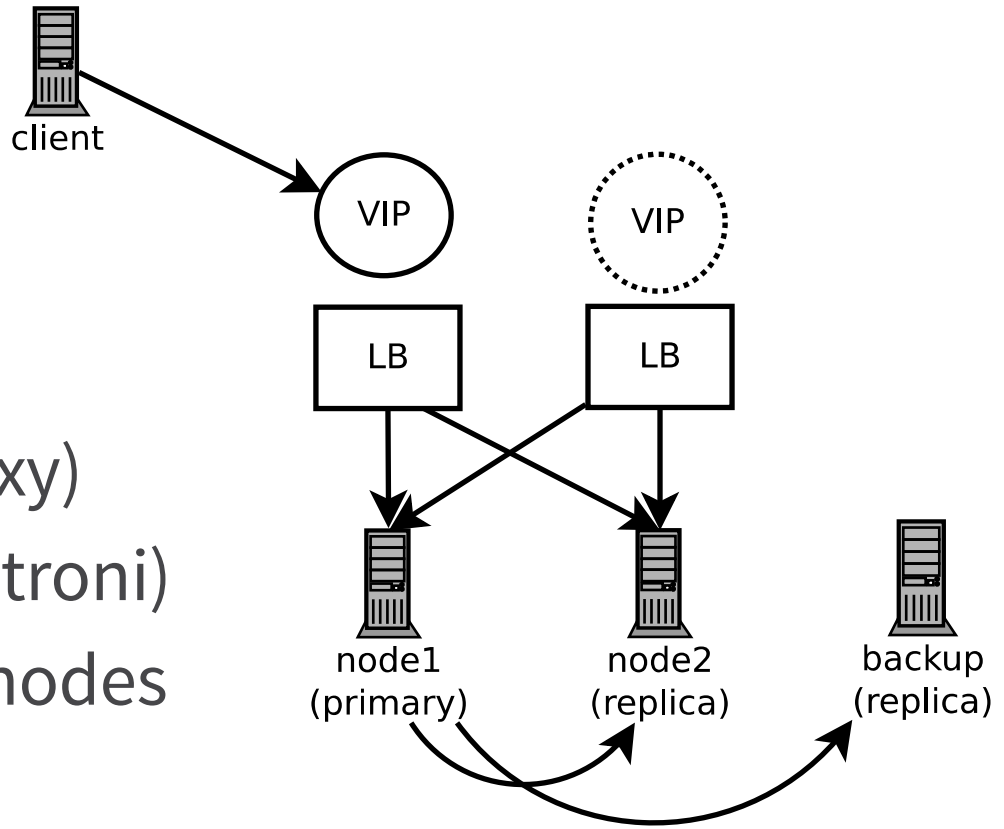


Percona Live Online
October 21, 2020

OVHcloud

Cluster example at OVHcloud

- Redundant
  - VIP (keepalived)
  - Load balancing (HAProxy)
  - Automatic failovers (Patroni)
- Distributed on multiple nodes

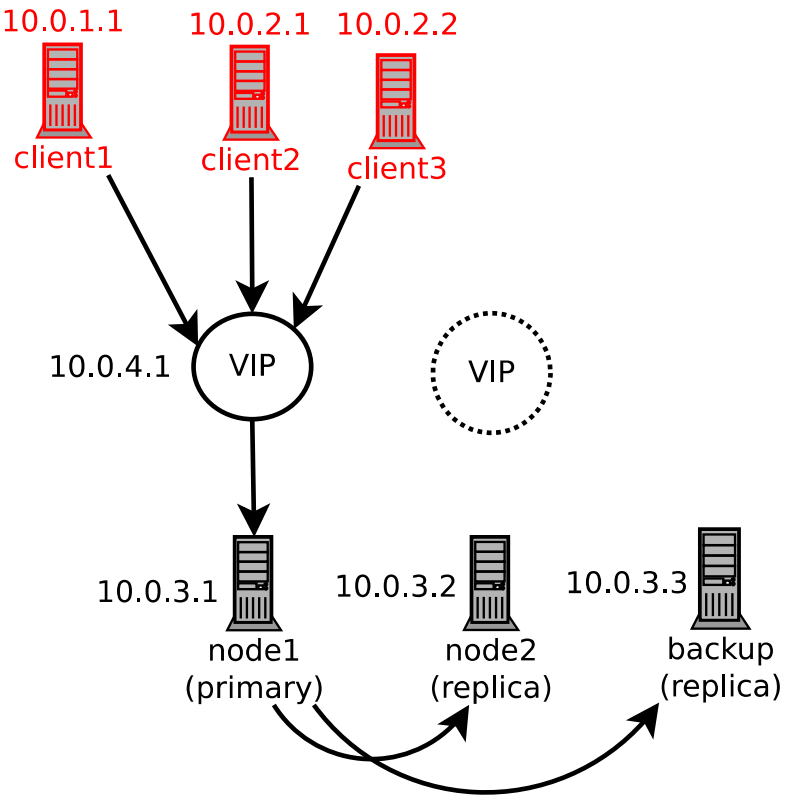


Proxies

- Between client and server
- HAProxy is not transparent
- Source IP address is altered!

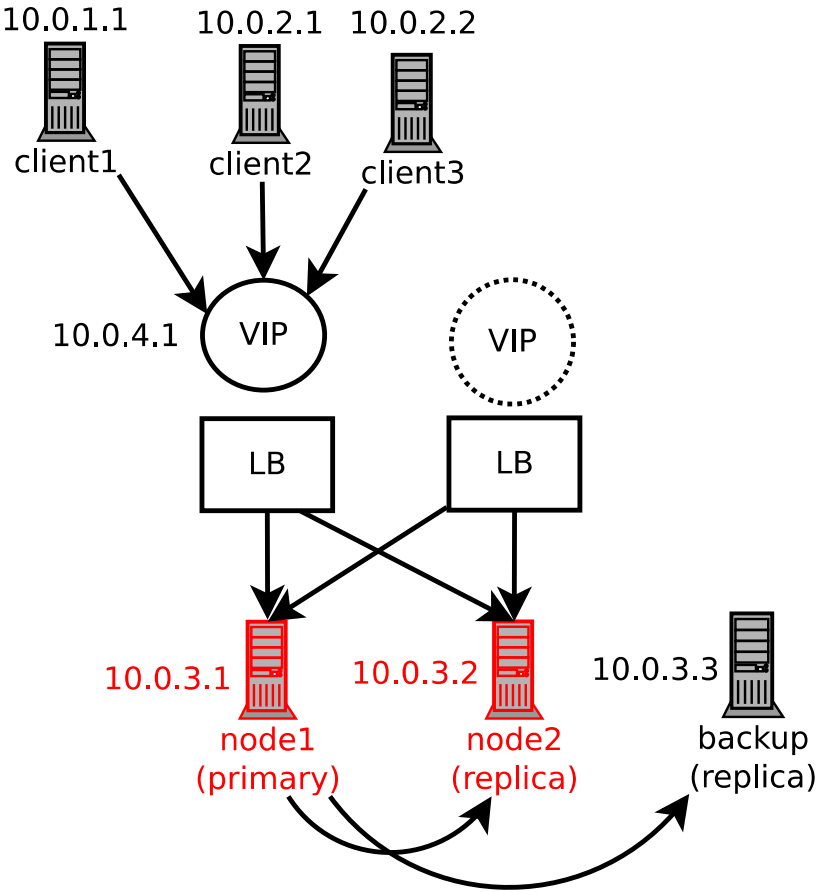
Without proxy

- client → nodes
- Nodes see real client IP address



With proxy

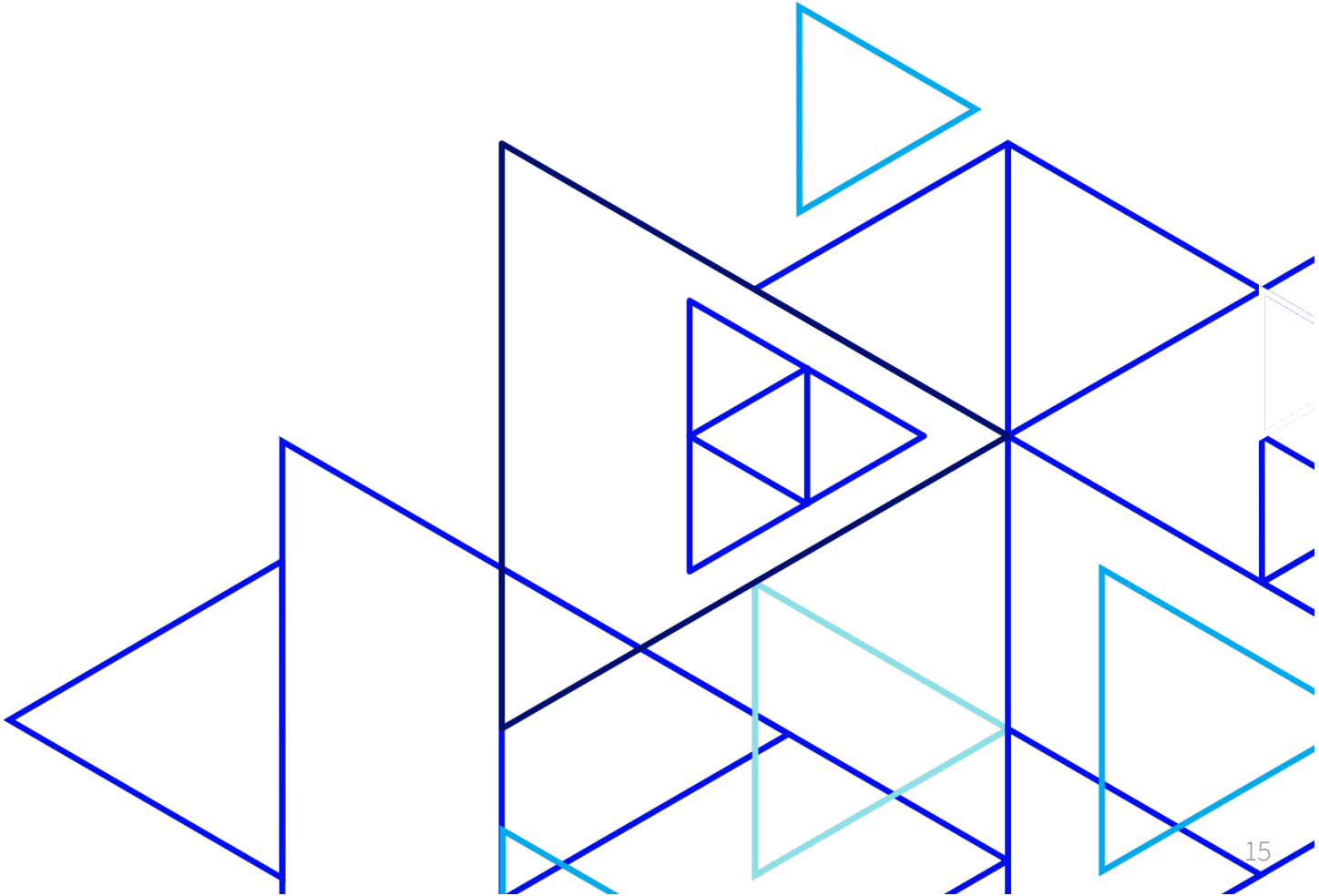
- client → proxy → nodes
- Nodes see proxy IP address



Credentials sharing at OVHcloud

- Multiple applications → one user → one database
- Use an API with RBAC
  - Project still running since 2015
- One application → one user
  - Database user management not industrialized yet

Security



Percona Live Online
October 21, 2020

OVHcloud

Host-based authentication (pg_hba.conf)

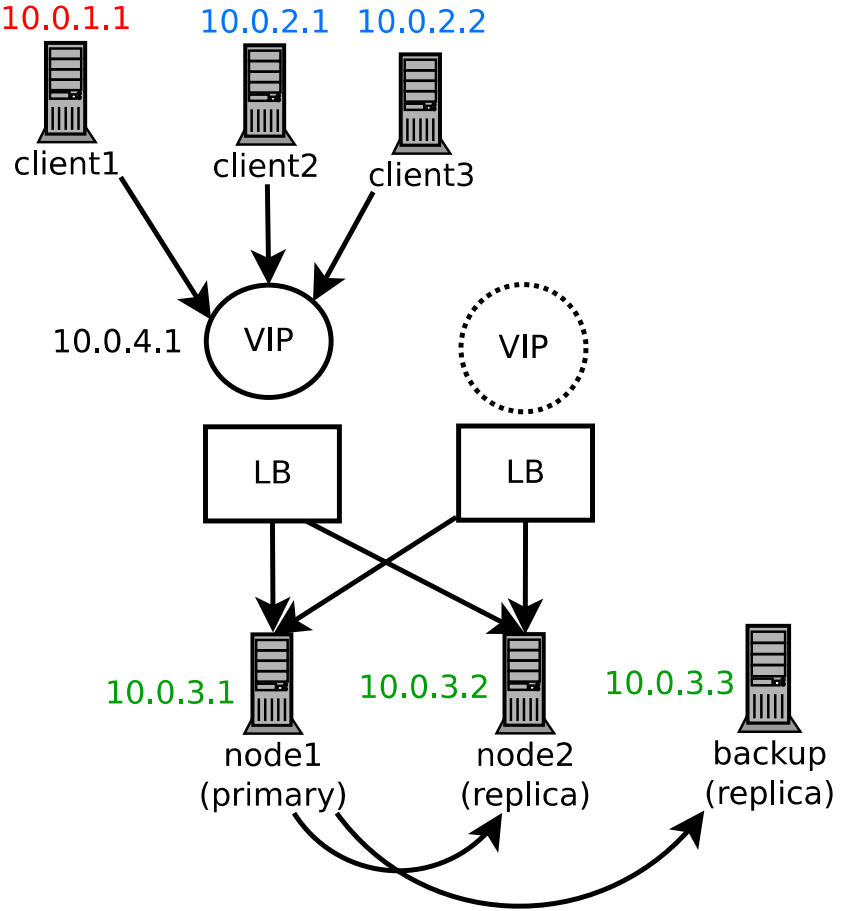
- Username
- IP address/range
- Database name
- Authentication method
  - md5, scram-sha-256
  - cert
  - etc.

Host-based authentication (pg_hba.conf)

- Username
- **IP address/range**
- Database name
- Authentication method
  - md5, scram-sha-256
  - cert
  - etc.

Host-based authentication

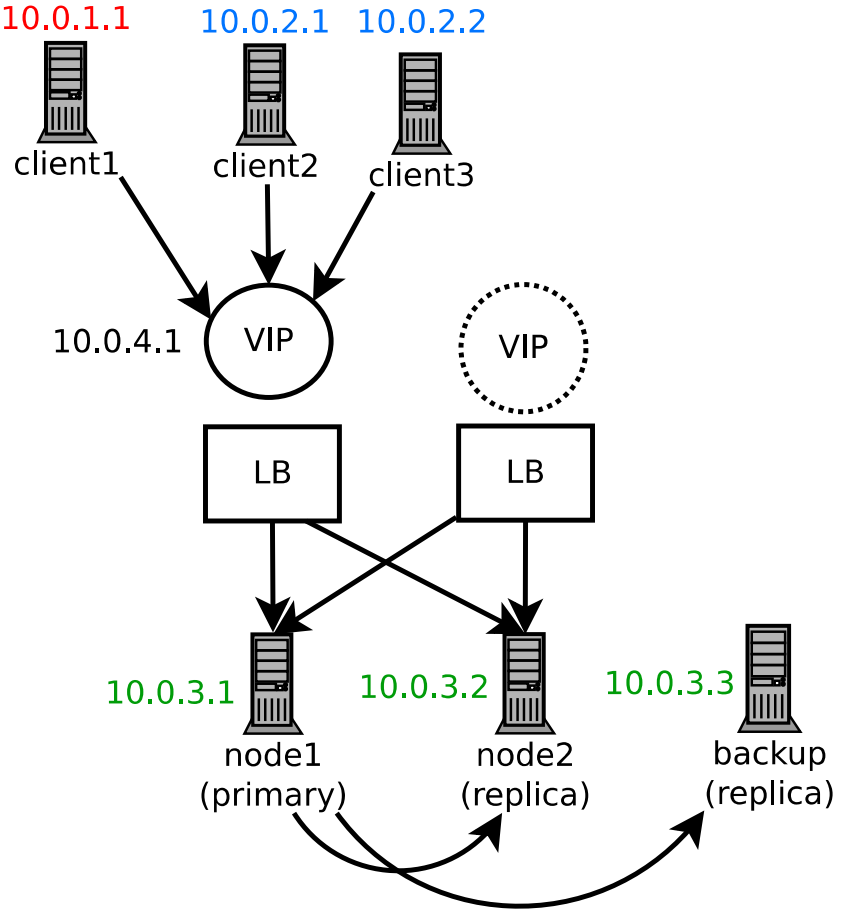
- Users
  - **client1** → **app1**
  - **client2** & **3** → **app2**
- Requirements
  - **client1** → database **app1**
  - **client2** & **3** → database **app1**



Host-based authentication

- **client1** → database **app1**
- **client2** & **3** → database **app1**

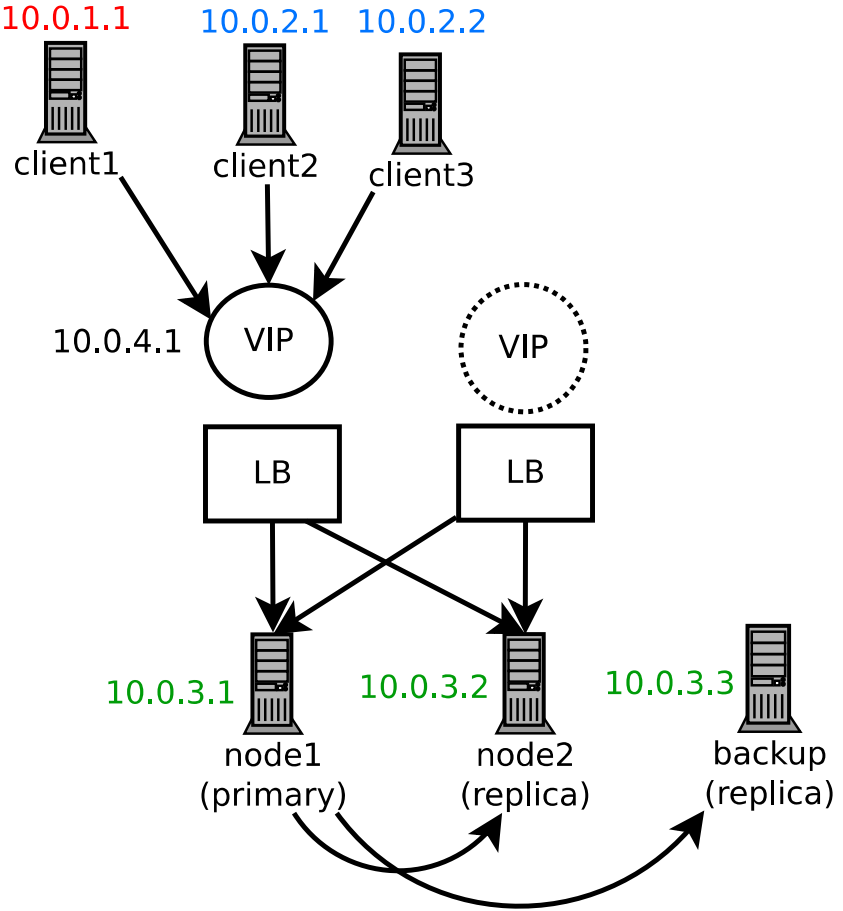
```
hostssl app1 app1 10.0.1.1 md5
```



Host-based authentication

- **client1** → database **app1**
- **client2** & **3** → database **app1**

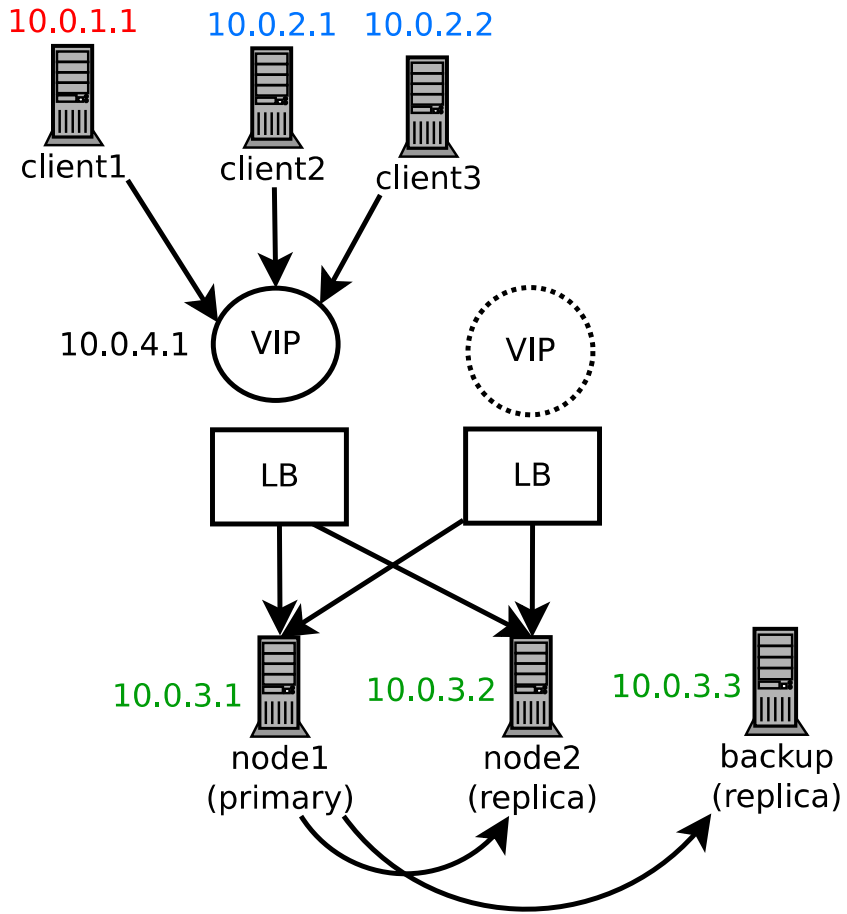
```
hostssl app1 app1 10.0.1.1 md5
```



Host-based authentication

- **client1** → database **app1**
- **client2** & **3** → database **app1**

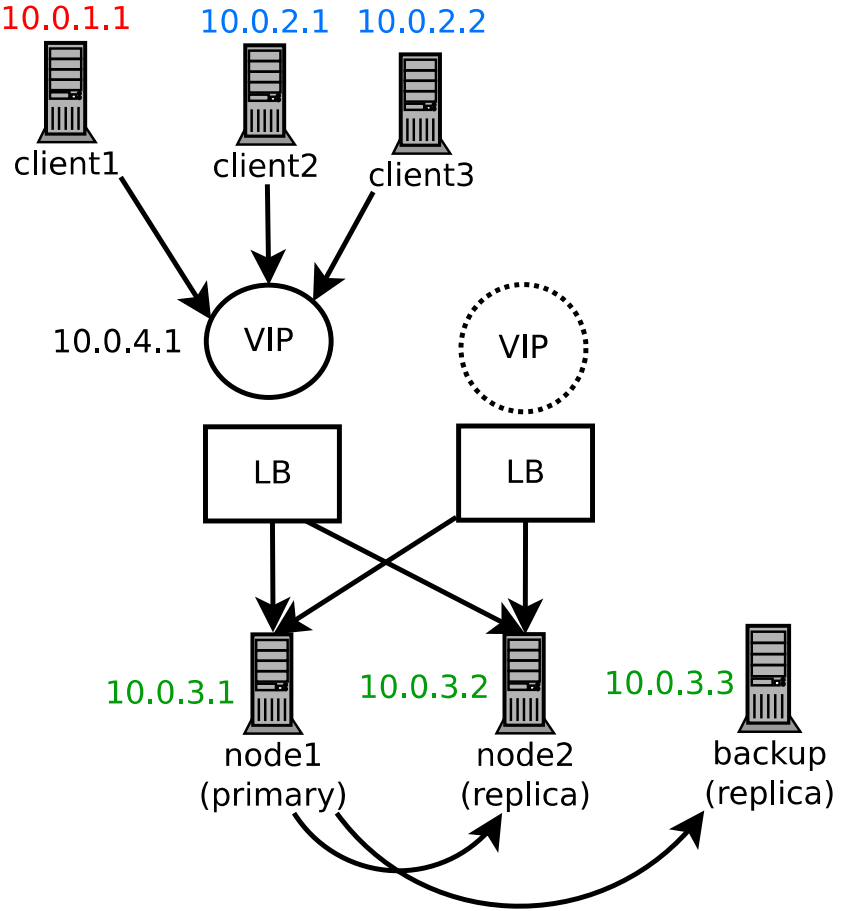
```
hostssl app1 app1 10.0.1.1 md5
hostssl app1 app1 10.0.3.1 md5
hostssl app1 app1 10.0.3.2 md5
```



Host-based authentication

- **client1** → database **app1**
- **client2** & **3** → database **app1**

```
hostssl app1 app1 10.0.1.1 md5
hostssl app1 app1 10.0.3.1 md5
hostssl app1 app1 10.0.3.2 md5
```

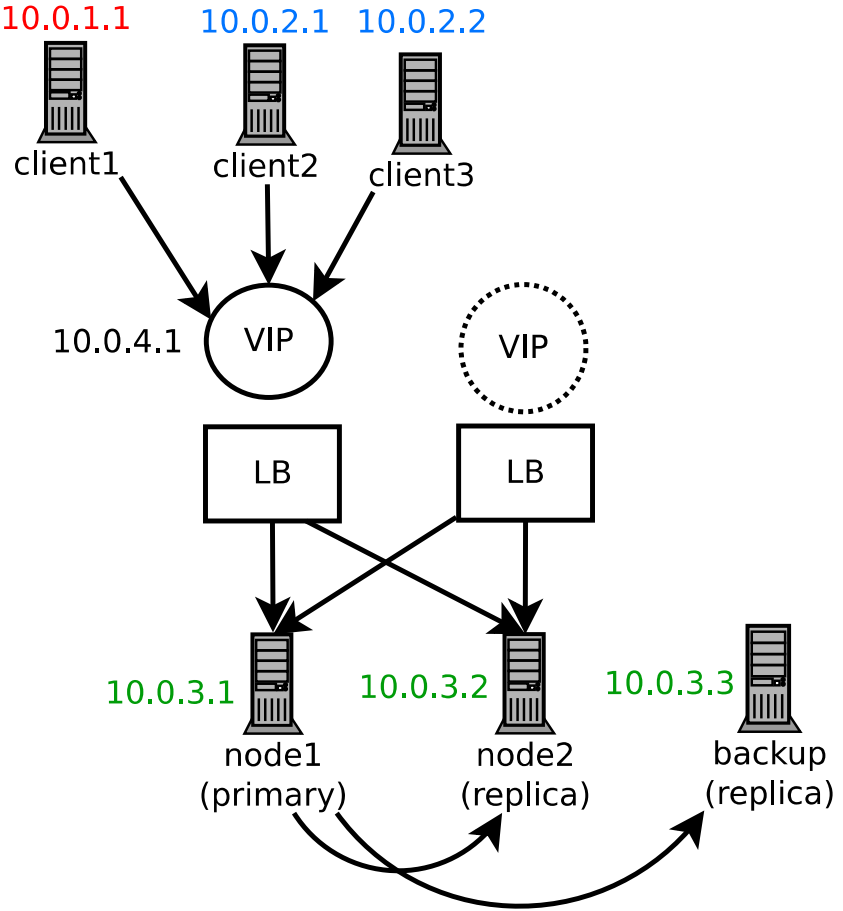


Host-based authentication

- **client1** → database **app1**
- **client2** & **3** → database **app1**

```
hostssl app1 app1 10.0.1.1 md5
hostssl app1 app1 10.0.3.1 md5
hostssl app1 app1 10.0.3.2 md5
```

- **client2** steal **app1** credentials

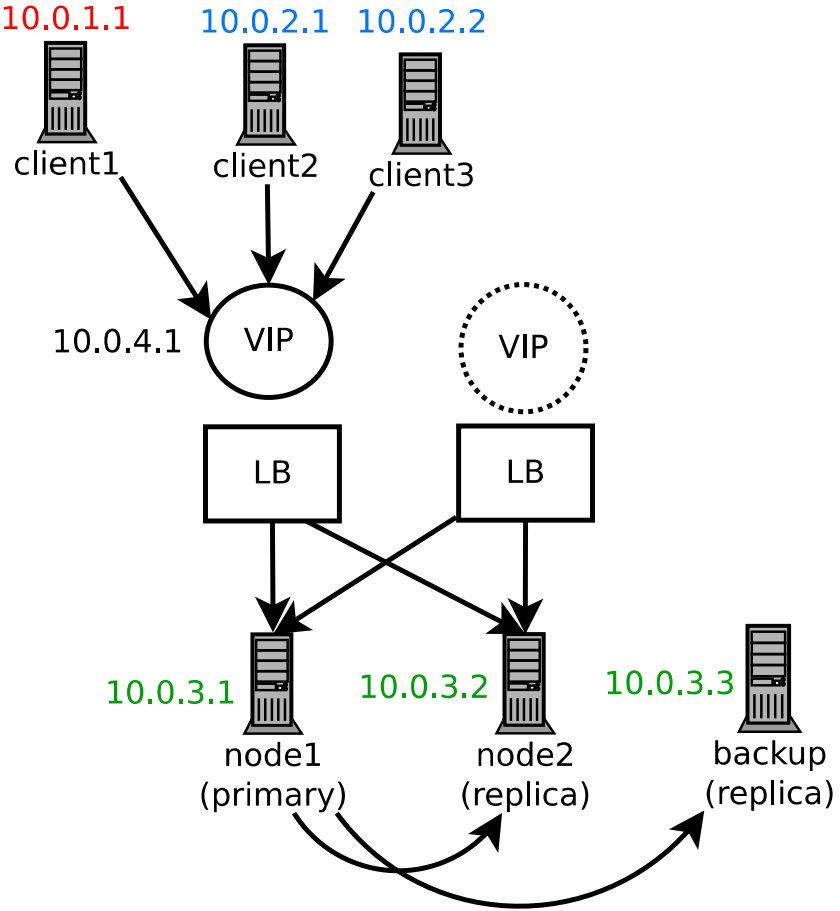


Host-based authentication

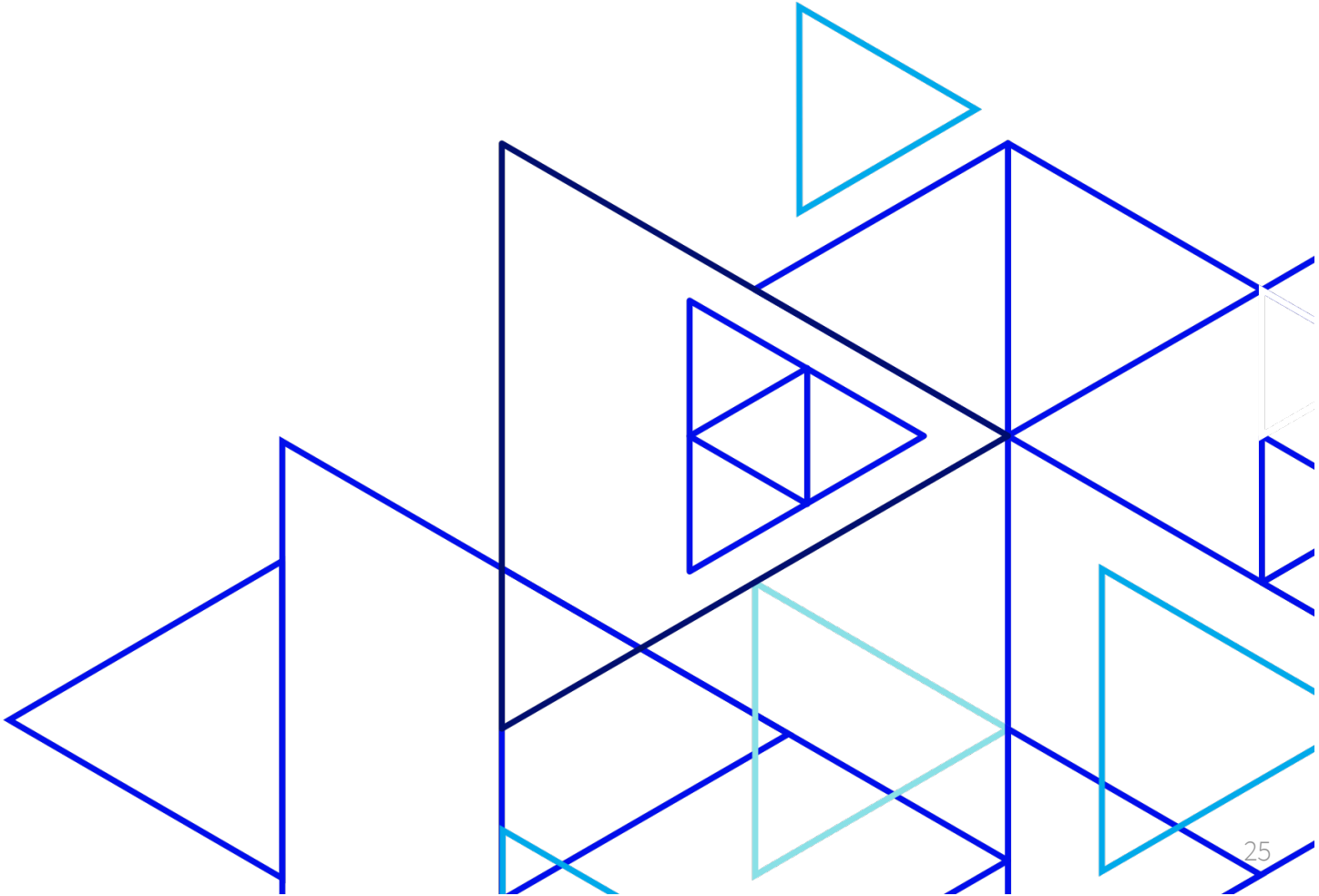
- **client1** → database **app1**
- **client2** & **3** → database **app1**

```
hostssl app1 app1 10.0.1.1 md5
hostssl app1 app1 10.0.3.1 md5
hostssl app1 app1 10.0.3.2 md5
```

- **client2** steal **app1** credentials



Traceability



Percona Live Online
October 21, 2020

OVHcloud

Audit

- Who
- What
- Where
- When
- Why

Audit

- Who
- What
- Where
- When
- Why

Five Ws

(information gathering)

Audit

- Who → user identification
- What
- Where
- When
- Why

Audit

- Who → user identification
- What
- Where
- When
- Why

Audit

- Who → user identification
- What → type of event, content
- Where
- When
- Why

Audit

- Who → user identification
- What → type of event, content
- Where
- When
- Why

Audit

- Who → user identification
- What → type of event, content
- Where → event location and origin
- When
- Why

Audit

- Who → user identification
- What → type of event, content
- Where → event location and origin
- When
- Why

Audit

- Who → user identification
- What → type of event, content
- Where → event location and origin
- When → date and time
- Why

Audit

- Who → user identification
- What → type of event, content
- Where → event location and origin
- When → date and time
- Why

Audit

- Who → user identification
- What → type of event, content
- Where → event location and origin
- When → date and time
- Why → deduced by previous questions

Audit

- Who → user identification
- What → type of event, content
- Where → event location and origin
- When → date and time
- Why → deduced by previous questions

Audit – Connections

- `log_connections = 1`
- Real client is hidden

```
2020-09-08 14:04:19 UTC [3868]: [2-1]
db=x,user=client1,app=[unknown],client=10.0.3.1* LOG:
  connection authorized: user=client1 database=x SSL
enabled (protocol=TLSv1.2, cipher=ECDHE-RSA-AES256-GCM-
                SHA384, compression=off)
```

*load balancer IP address

Audit – Disconnections

- `log_disconnections = 1`
- Real client is hidden

```
2020-09-08 14:04:19 UTC [3868]: [3-1]
db=x,user=client1,app=real_application,client=10.0.3.1* LOG:
    disconnection: session time: 0:00:00.022 user=client1
    database=x host=10.0.3.1* port=59514
```

*load balancer IP address

Audit – DDL

- Data Definition Language (CREATE, ALTER, DROP, ...)
- `log_statement = 'ddl'`
- Real client is hidden

```
2020-09-08 15:21:35 UTC [30554]: [1-1]
db=x,user=client1,app=psql,client=10.0.3.1* LOG:
statement: alter role jriou with password 'cleartext' ;
```

*load balancer IP address

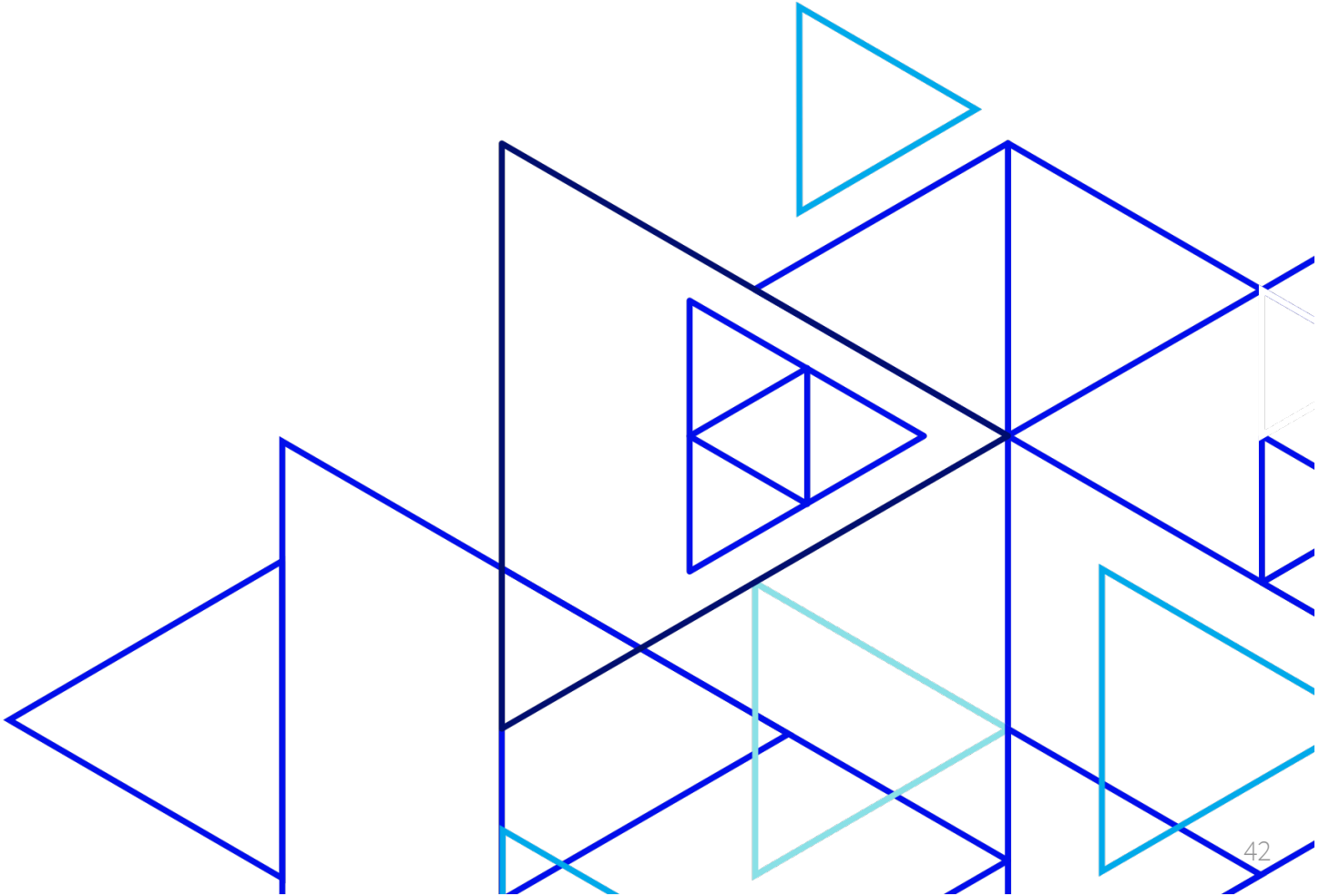
Audit – Slow queries

- `log_min_duration_statement = 1000` (1s)
- Real client is hidden

```
2020-09-07 22:15:12 UTC [13715]: [1-1]
db=x,user=client1,app=real_application,client=10.0.3.1*
LOG:  duration: 1629.788 ms  execute <unnamed>: SELECT
                        *  FROM …
```

*load balancer IP address

PostgreSQL solutions



Percona Live Online
October 21, 2020

OVHcloud

Application name

- Arbitrary string set by client at connection
- `set application_name = '10.0.1.1';`
- Logging problem solved
- Not compatible with pg_hba.conf rules

Application name at OVHcloud

+

- Used in containerized environments
- Cannot be used for security

Credits
https://github.com/googlefonts/noto-emoji/blob/master/svg/emoji_u1f44d.svg
https://github.com/googlefonts/noto-emoji/blob/master/svg/emoji_u1f44e.svg

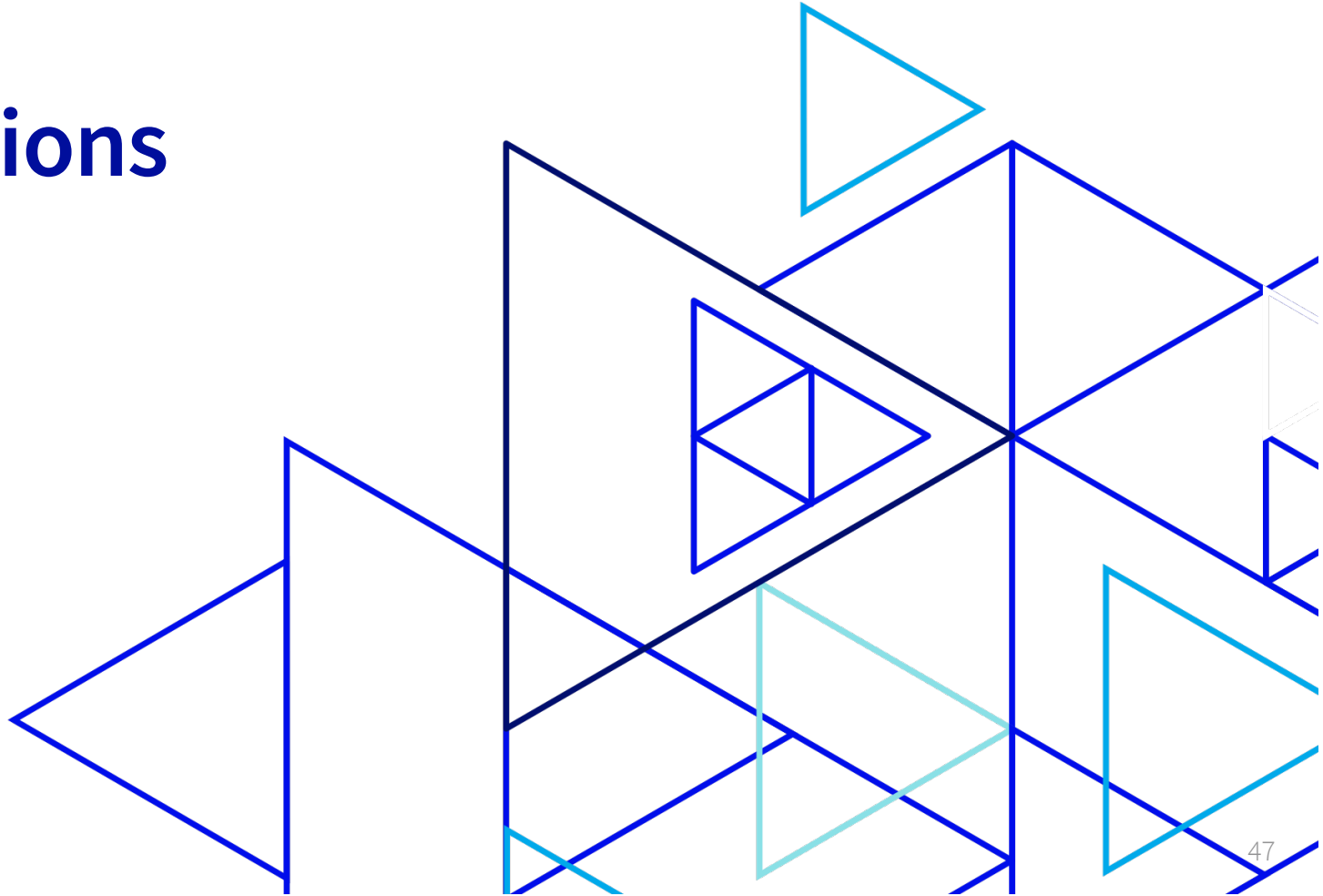
PgBouncer

- Lightweight connection pooler for PostgreSQL
- `application_name_add_host = 1`
- Compatible with pg_hba.conf rules

PgBouncer at OVHcloud

- Database side connection pooling not required
- Unidentified random connection errors

Generic solutions



Percona Live Online
October 21, 2020

OVHcloud

Proxy Protocol

- Proposed by HAProxy
- Protocol solution
- Same idea as `X-Forwarded-For` HTTP header
- Text (v1) and binary (v2) formats
- IPv4 and IPv6 supported
- Both ends must support it

Proxy Protocol

- Supported by plenty of products
  - Load balancers → HAProxy
  - Web servers → Apache HTTPD, nginx, gunicorn, varnish
  - Mail servers → postfix, dovecot, exim

Proxy Protocol

- Supported by other DBMS
  - Percona Server
  - MariaDB, MaxScale
- But not by PostgreSQL
  - Suggestion on pgsql-hackers → https://bit.ly/2GGBqTP
  - Contribution to PgBouncer → https://bit.ly/35lMfVS

Proxy Protocol at OVHcloud

- Not yet implemented
- Hard to contribute

Transparent proxy

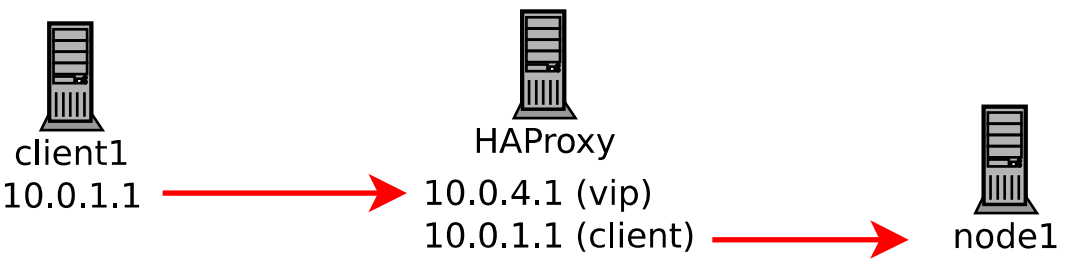
- HAProxy can spoof client IP address
  - kernel
  - HAProxy configuration
  - iptables
  - ip rule/route

Transparent proxy – kernel

- `CONFIG_NETFILTER_XT_TARGET_TPROXY=y`
- `net.ipv4.ip_nonlocal_bind=1`

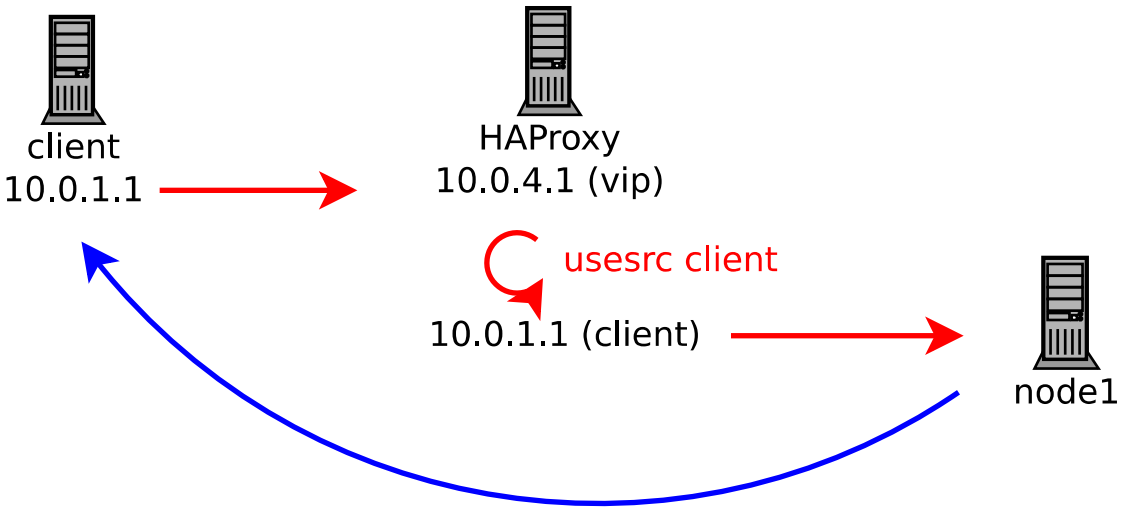
Transparent proxy – HAProxy

- TPROXY compilation option
  - Enabled when TARGET is linux26 or linux2628
- Run as root
- `source 0.0.0.0:5432 usesrc client`
- Client IP and port are used now



Transparent proxy – Reverse route

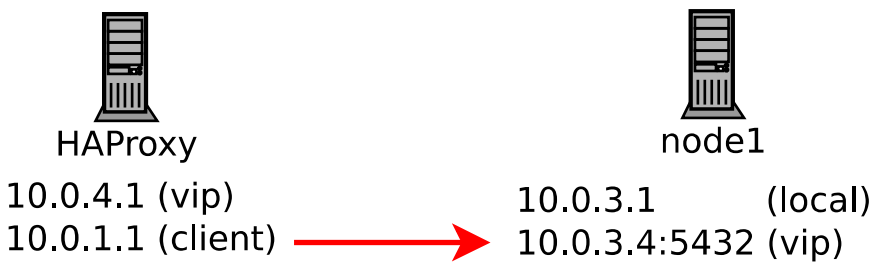
- Asymmetric routing doesn't work



Transparent proxy – HAProxy

- Send traffic to a second IP address on nodes

```
server node1 10.0.3.4:5432 [...]
```



Transparent proxy – iptables

- On nodes, mark outgoing PostgreSQL traffic

```
iptables -A OUTPUT -p tcp -s 10.0.3.4 --sport 5432
         -j MARK --set-mark 1
```

Transparent proxy – ip rule/route

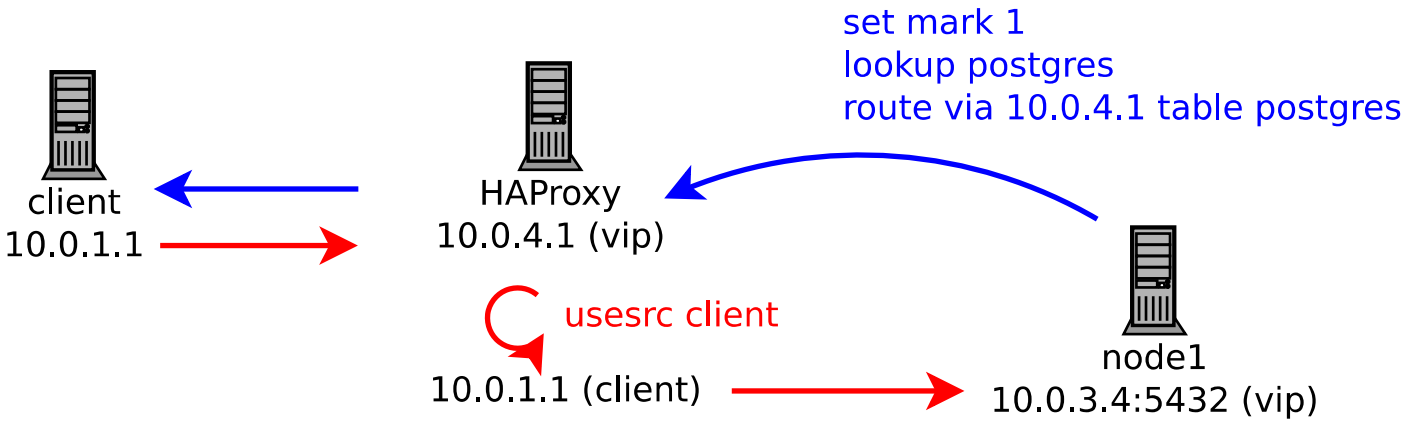
- Add a rule to define routing table for mark

```
ip rule add fwmark 1 lookup postgres
```

- And route to the service VIP

```
ip route add default via 10.0.4.1 dev eth0 table postgres
```

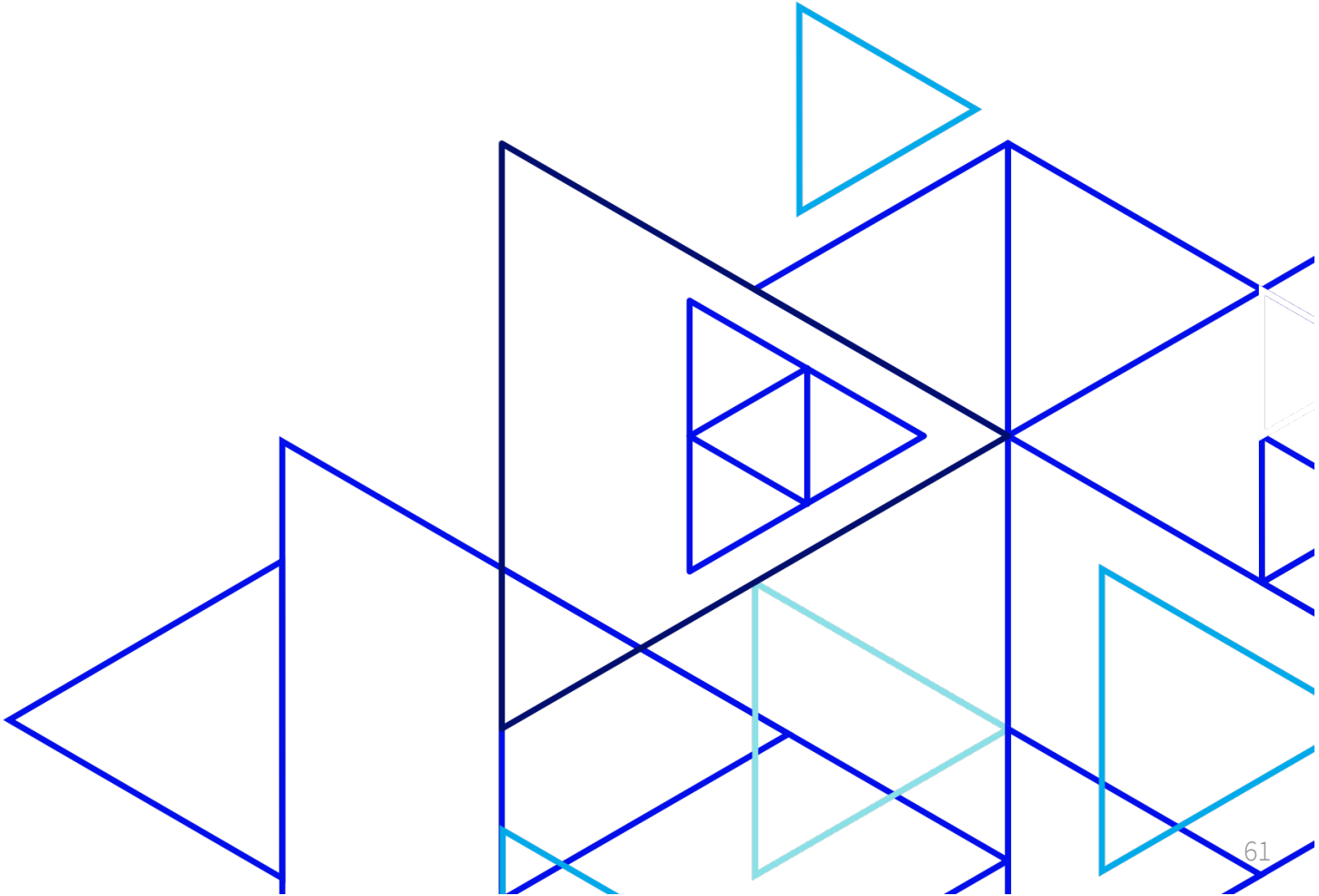
Transparent proxy



Transparent proxy at OVHcloud

- Runs on production since 2 years
- DBMS independent
- Hard to operate for DBAs

Conclusion



Percona Live Online
October 21, 2020

OVHcloud

Conclusion

- Distributed systems can be secure and traceable
- PgBouncer is great for pooling
- Transparent HAProxy is hacky
- Proxy Protocol support for PostgreSQL would be nice


OVHcloud
Thank you