initMAX

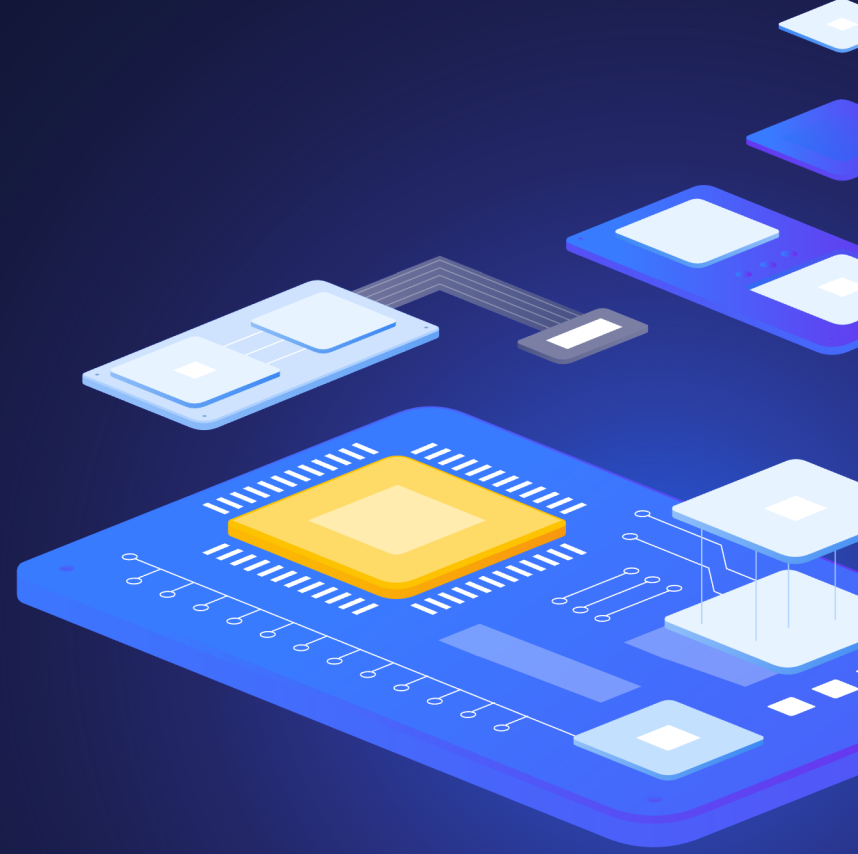
Webinar

PostgreSQL HA using Patroni

all your microphones are muted

ask your questions in Q&A, not in the Chat

use Chat for discussion, networking or applause



1

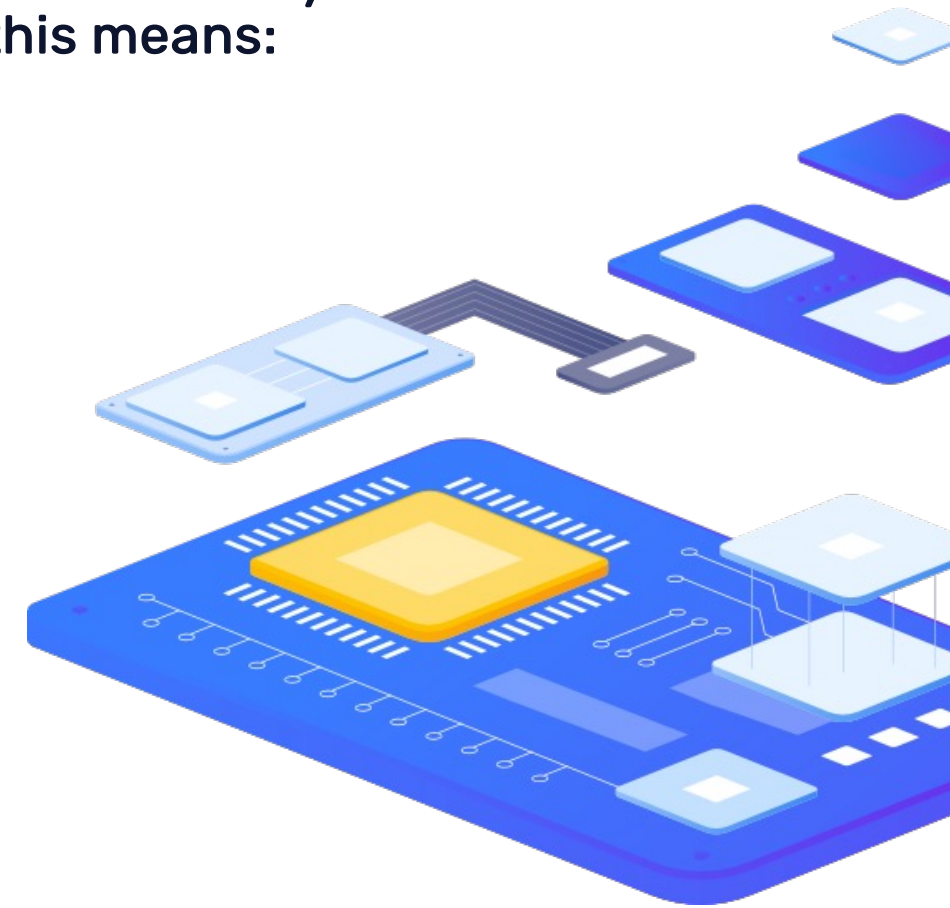
Introduction to PostgreSQL High Availability

initMAX

What is High Availability?

High Availability (HA) is the ability of a system to operate continuously without service interruption. For PostgreSQL databases, this means:

- Reducing the chance of failures
- Eliminating single points of failure (SPOF)
- Automatic detection of failures
- Automatic remediation to minimize impact



initMAX

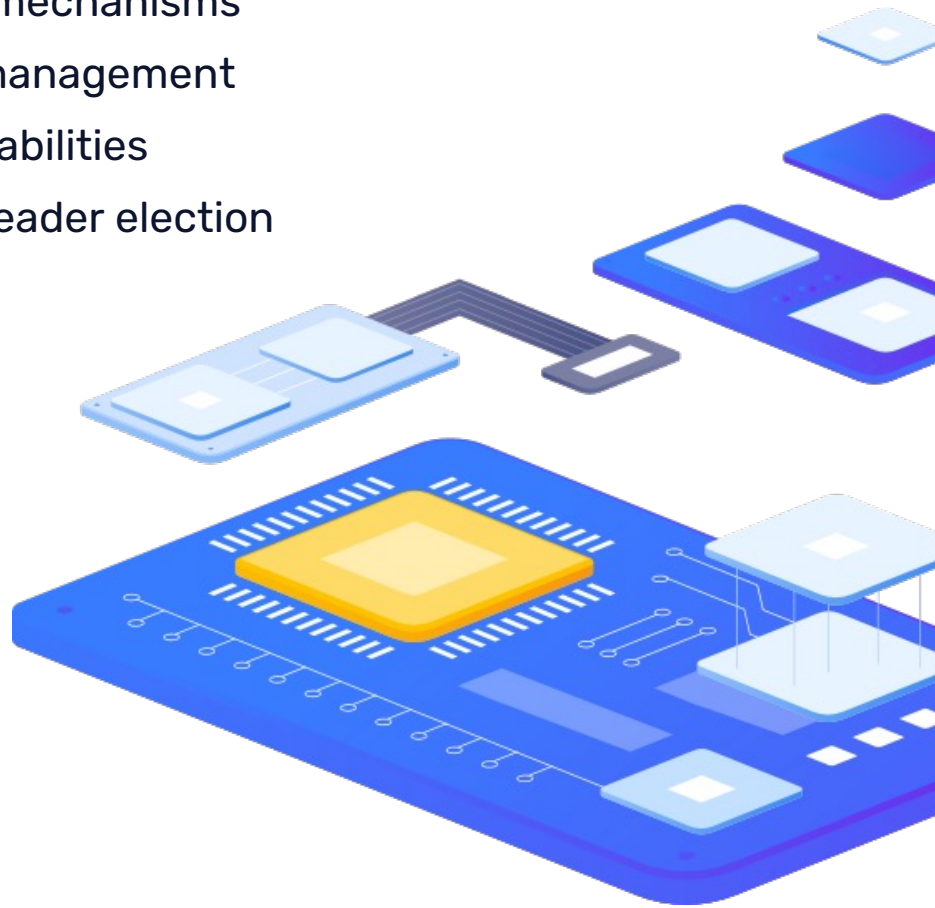
Why PostgreSQL Needs HA Solutions?

PostgreSQL's native features include:

- Streaming replication for data redundancy
- Point-in-time recovery
- WAL archiving

However, PostgreSQL lacks:

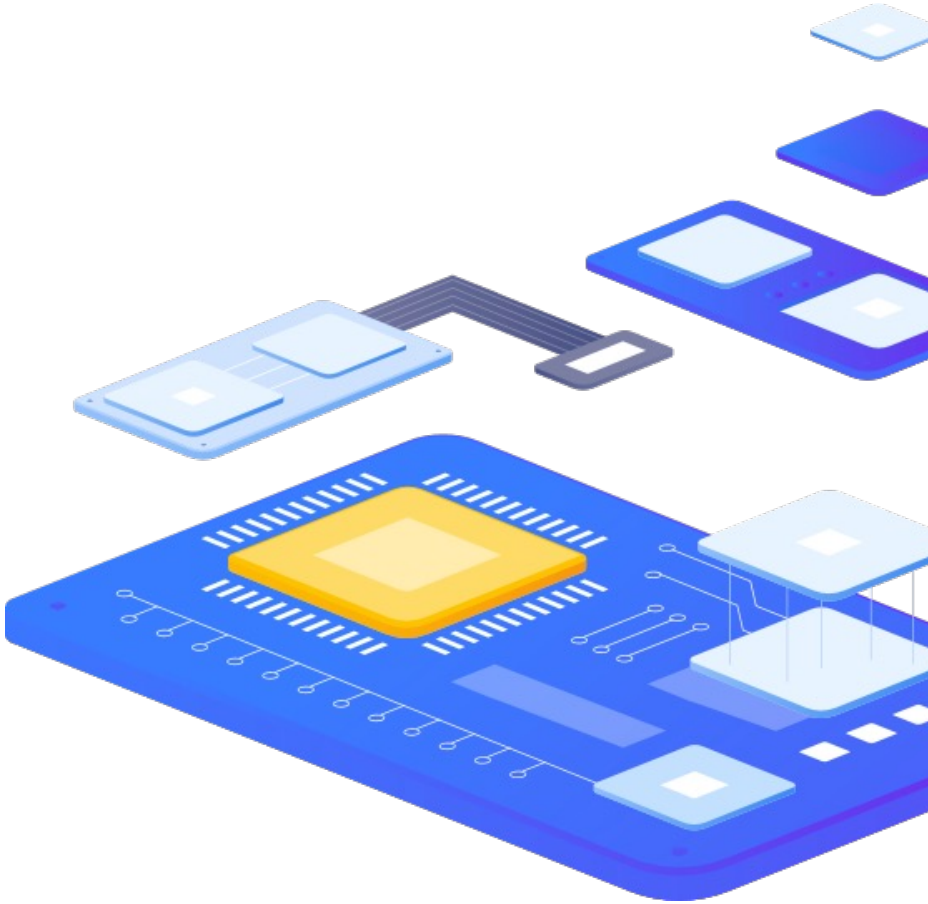
- Automatic failover mechanisms
- Advanced cluster management
- Load balancing capabilities
- Consensus-based leader election



initMAX

Key HA Requirements

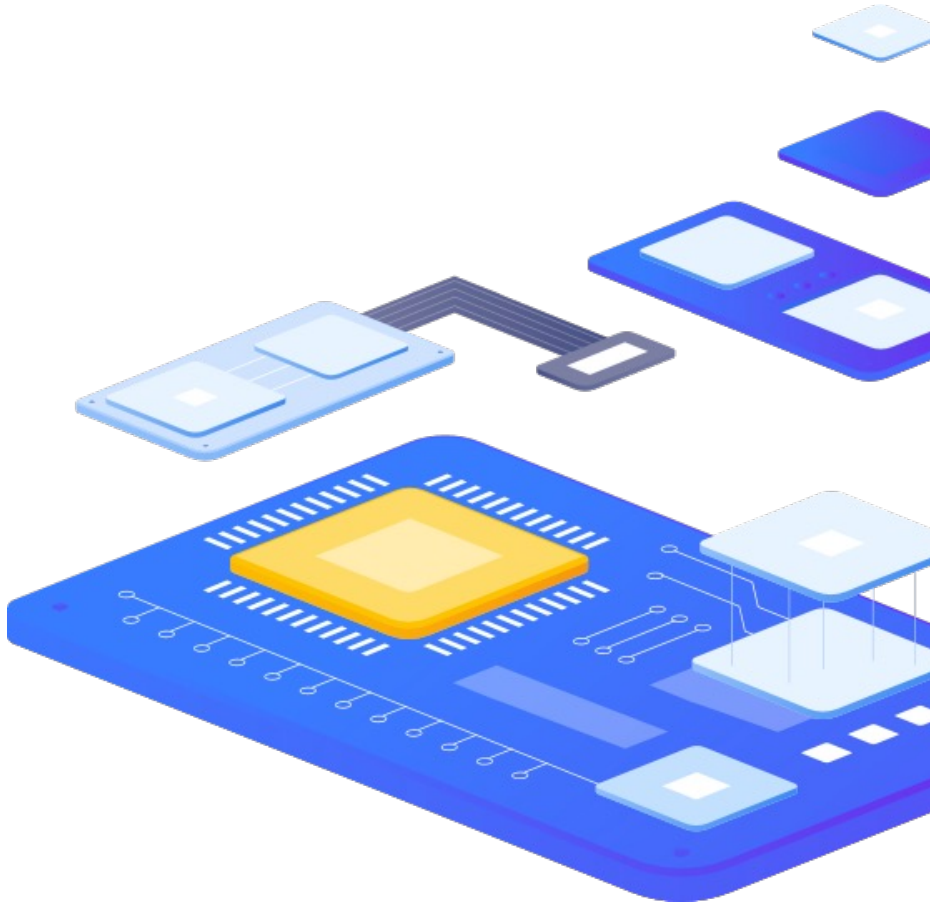
- **Replication:** Multiple copies of data across separate nodes
- **Switchover/Failover:** Planned and automatic primary role transfers
- **Connection Routing:** Intelligent traffic distribution
- **Backups:** Disaster recovery protection



initMAX

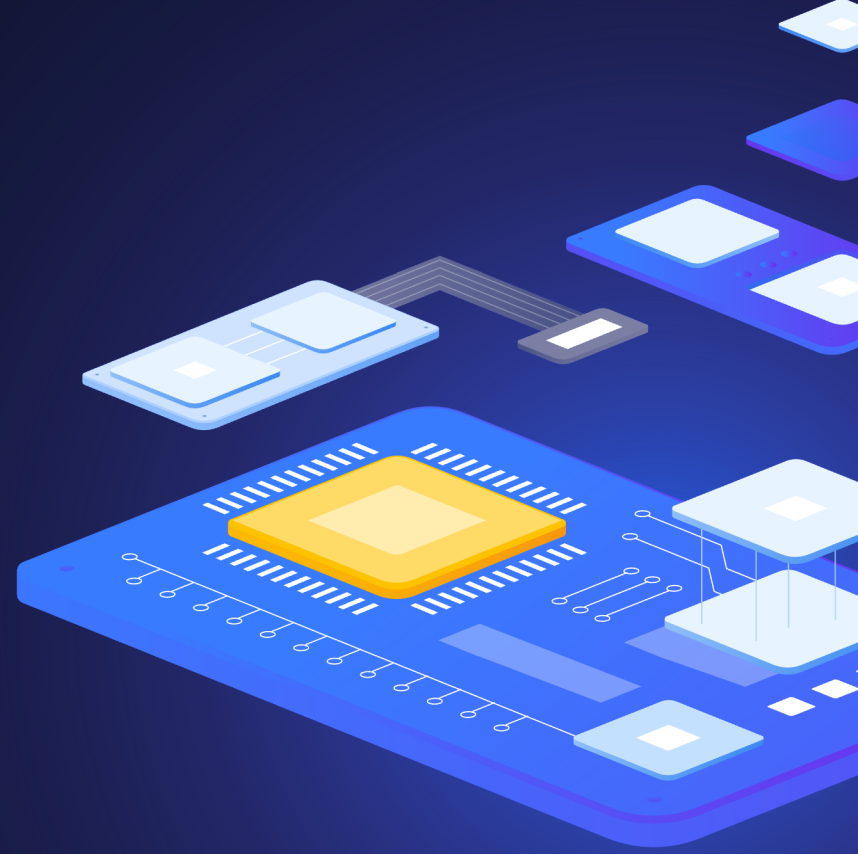
Business Impact of Downtime

- Customer loss and reputation damage
- SLA violations and penalties
- Revenue loss (estimated $5,500 per minute for large enterprises)
- Operational disruption



2

Overview of Patroni and Its Advantages

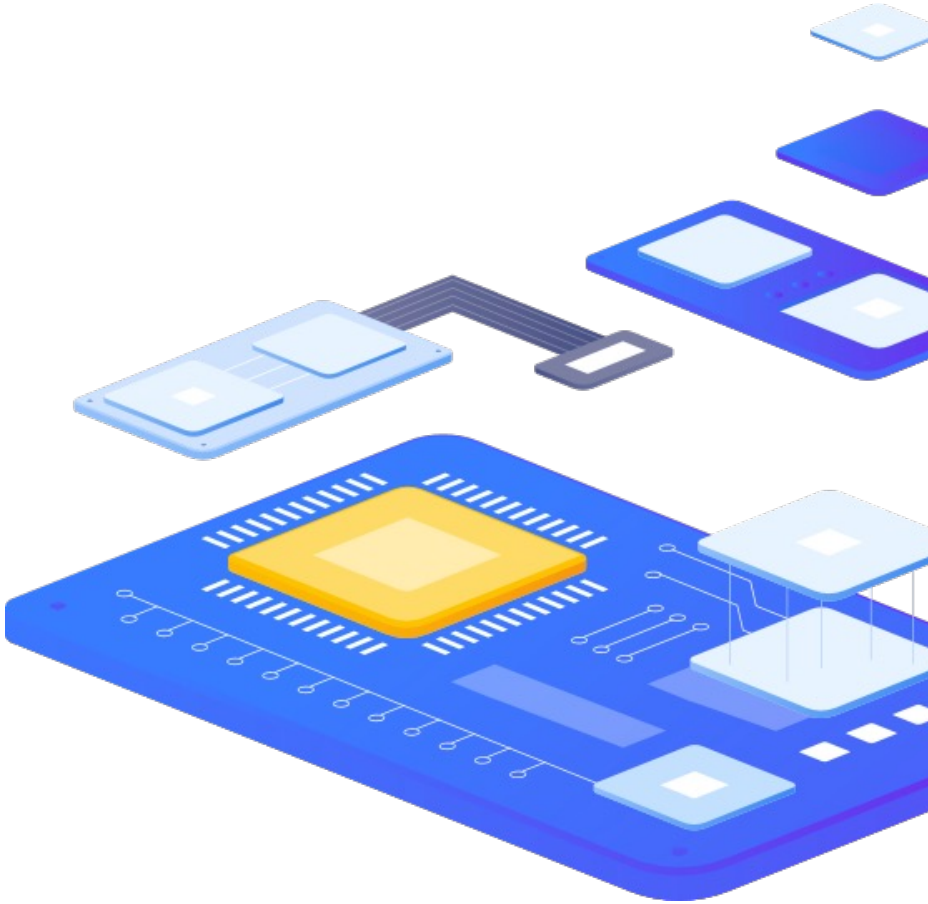


initMAX

What is Patroni?

Patroni is a Python-based template for creating high-availability PostgreSQL clusters using:

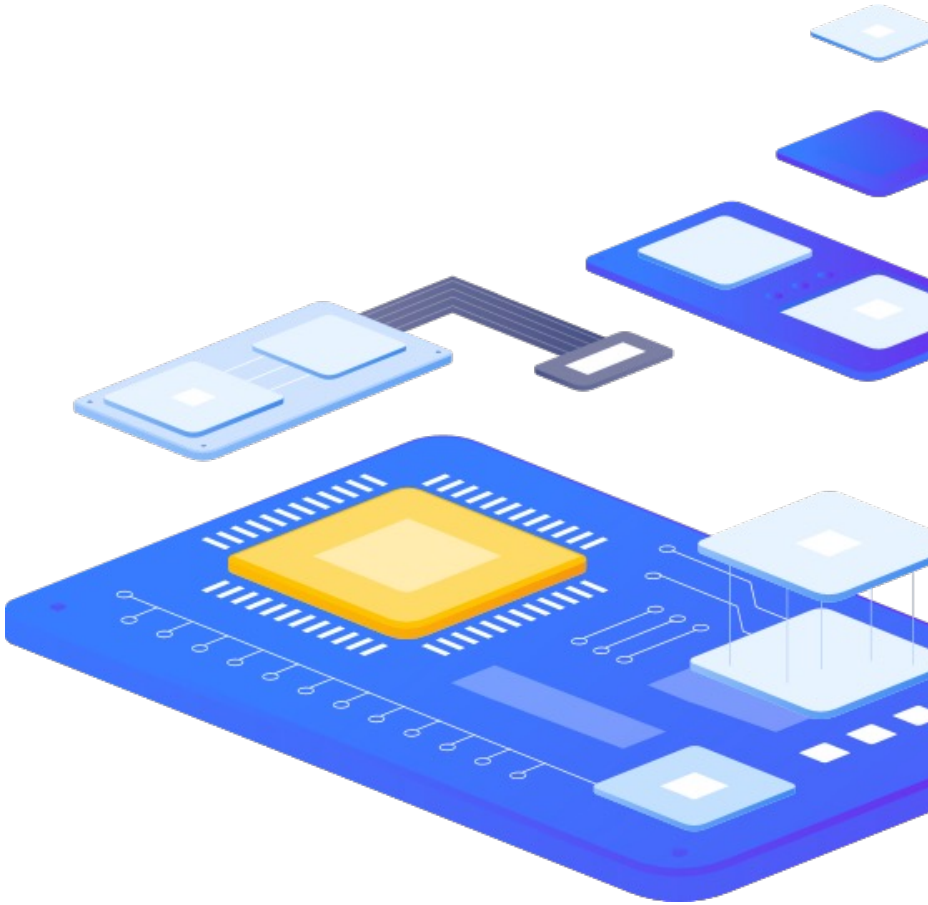
- PostgreSQL streaming replication
- Distributed consensus stores (etcd, Consul, ZooKeeper)
- REST API for management and monitoring



initMAX

Key Features

- Automatic leader election via distributed consensus
- Automatic failover with configurable policies
- Configuration management through YAML files
- REST API for monitoring and control
- Integration with load balancers and monitoring tools



initMAX

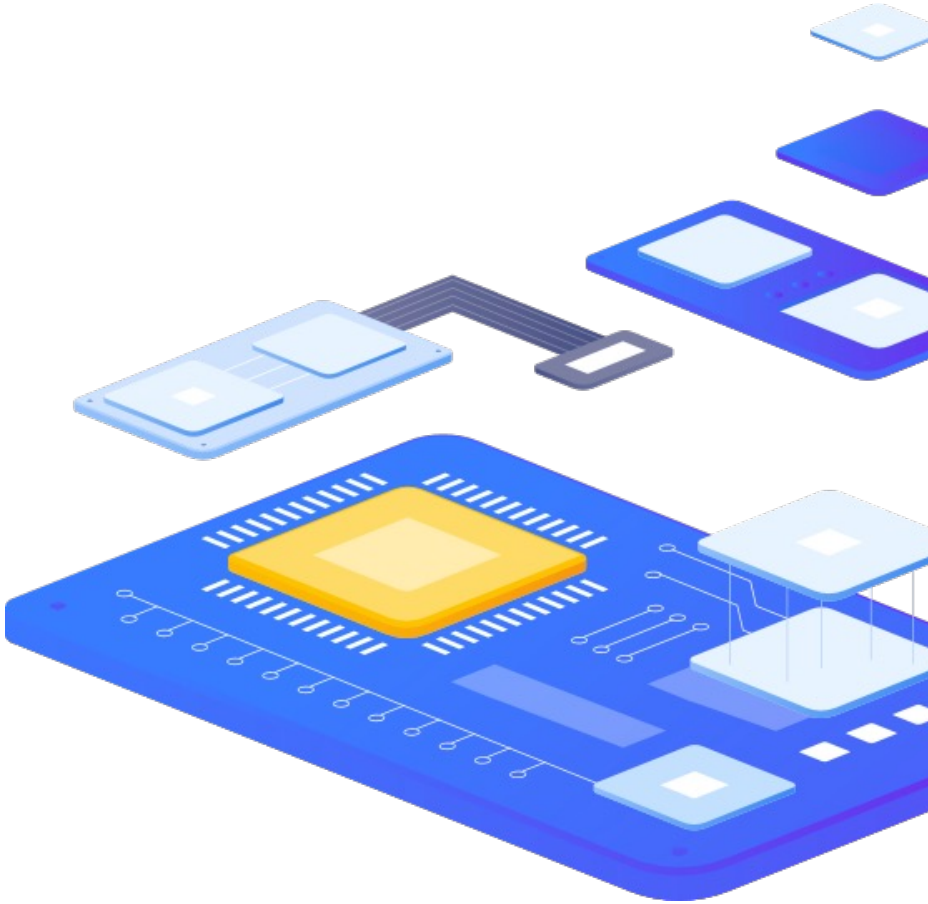
Patroni vs. Other HA Solutions

Feature	Patroni	Pacemaker/Corosync	Repmgr	Streaming Replication Only
Automatic Failover	✅	✅	✅	❌
Split-brain Protection	✅	✅	⚠️	❌
REST API	✅	❌	❌	❌
Cloud Native	✅	⚠️	⚠️	❌
Configuration Management	✅	⚠️	⚠️	❌
Learning Curve	Medium	High	Medium	Low

initMAX

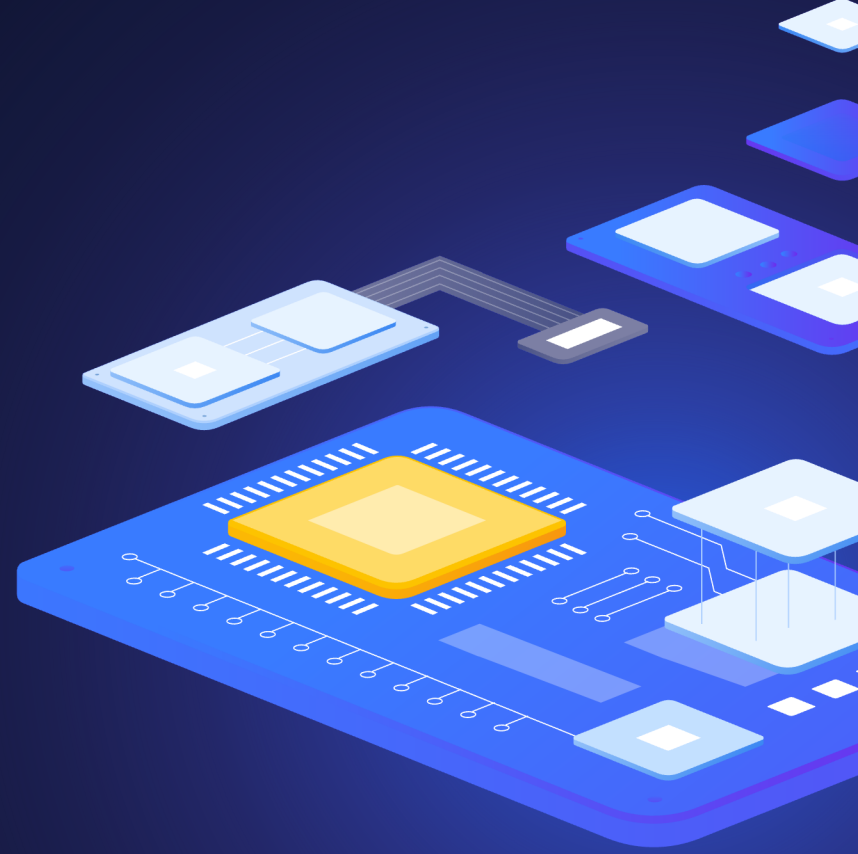
Advantages of Patroni

- **Consensus-based decisions** prevent split-brain scenarios
- **Cloud-native design** works well in containerized environments
- **Flexible configuration** via YAML and dynamic updates
- **Active community** and regular updates
- **PostgreSQL version compatibility** (9.3 to 17+)
- **Integration capabilities** with HAProxy, F5, Kubernetes, monitoring tools



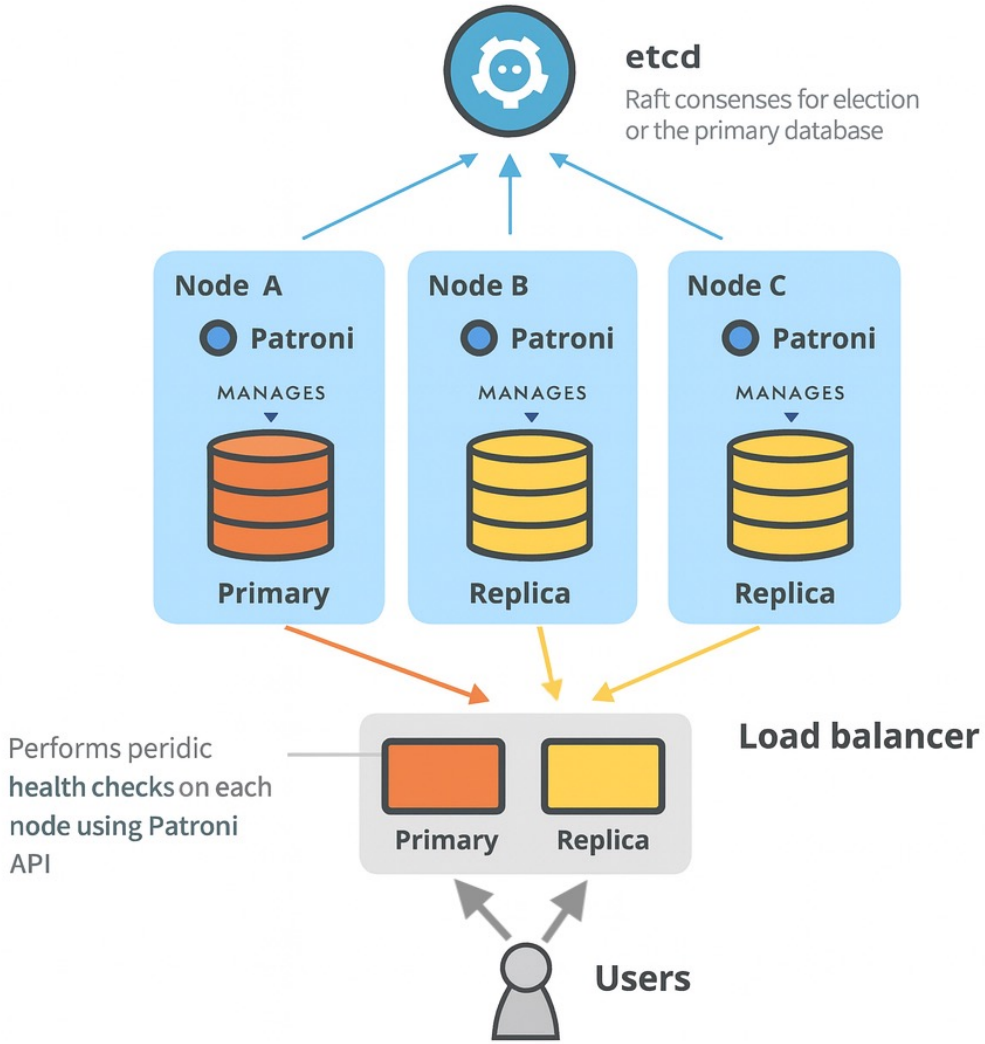
3

HA Cluster Architecture and Components



initMAX

Reference Architecture

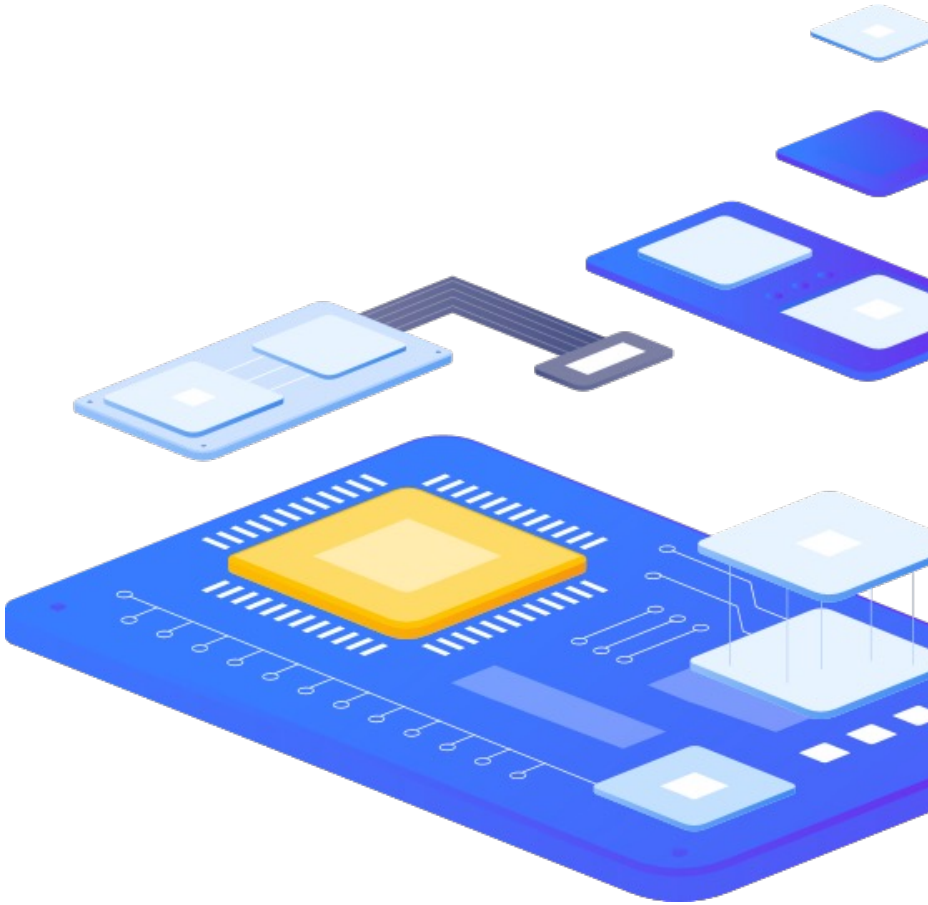


initMAX

Core Components

1. PostgreSQL Instances

- **Primary:** Accepts read/write operations
- **Replicas:** Handle read-only queries, ready for promotion
- **Minimum:** 2 nodes (1 primary + 1 replica)
- **Recommended:** 3+ nodes for better fault tolerance

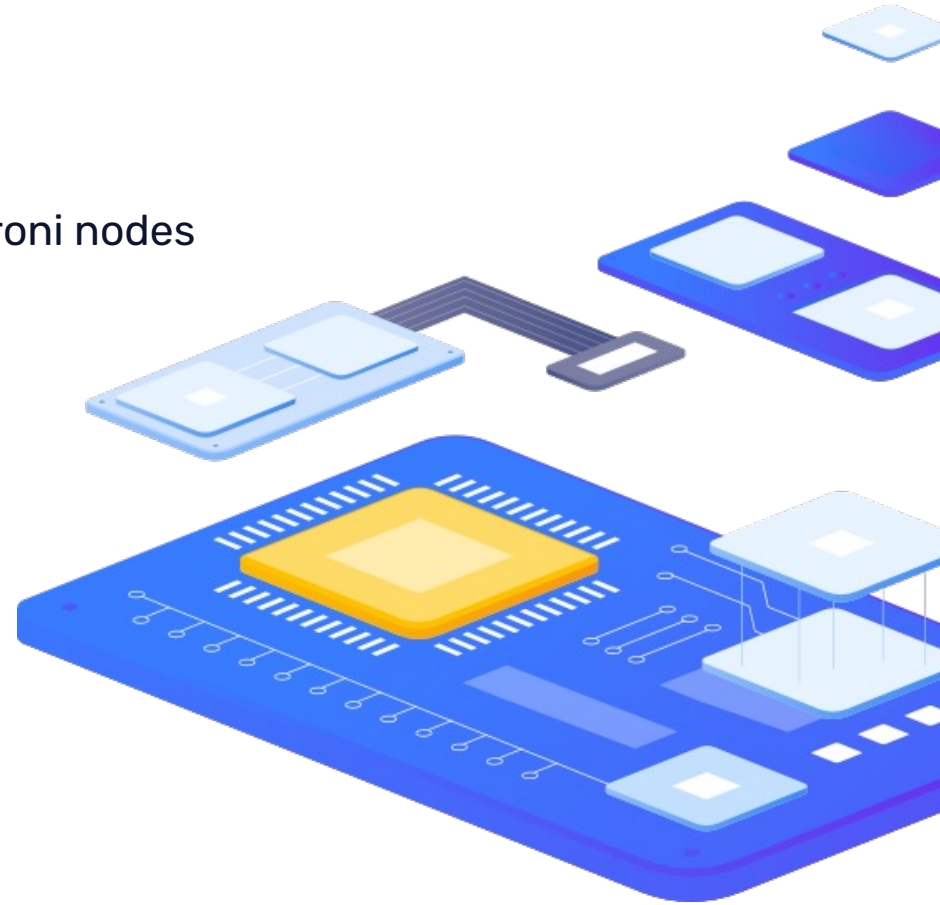


initMAX

Core Components

2. Patroni Agents

- **One agent per PostgreSQL node**
- **Manages:** Replication, failover, configuration
- **Communicates:** With DCS (Distributed Consensus Store) and other Patroni nodes
- **REST API:** Port 8008 (default) for monitoring



initMAX

Core Components

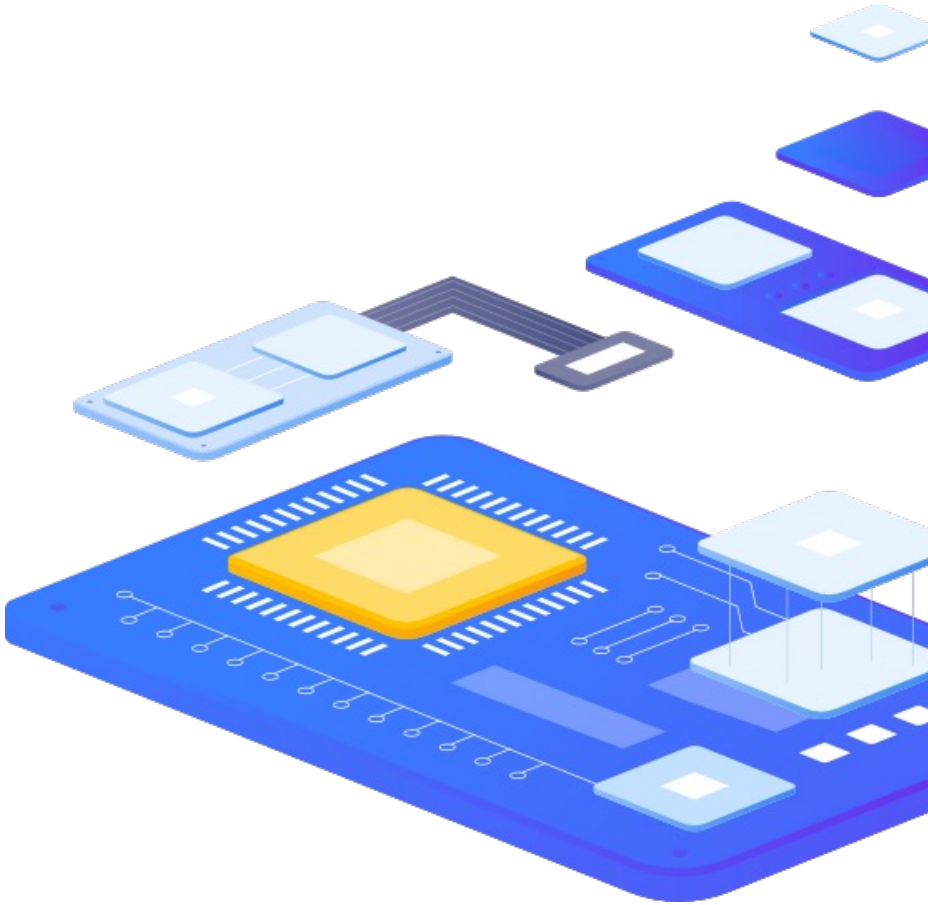
3. Distributed Consensus Store (DCS)

etcd (Recommended)

- Raft consensus algorithm
- Minimum 3 nodes for HA
- Stores cluster state and configuration

Alternatives

- Consul (with ACL support)
- ZooKeeper (Java-based)
- Kubernetes API (for K8s deployments)
- Pure Raft (deprecated from version 3.0!)

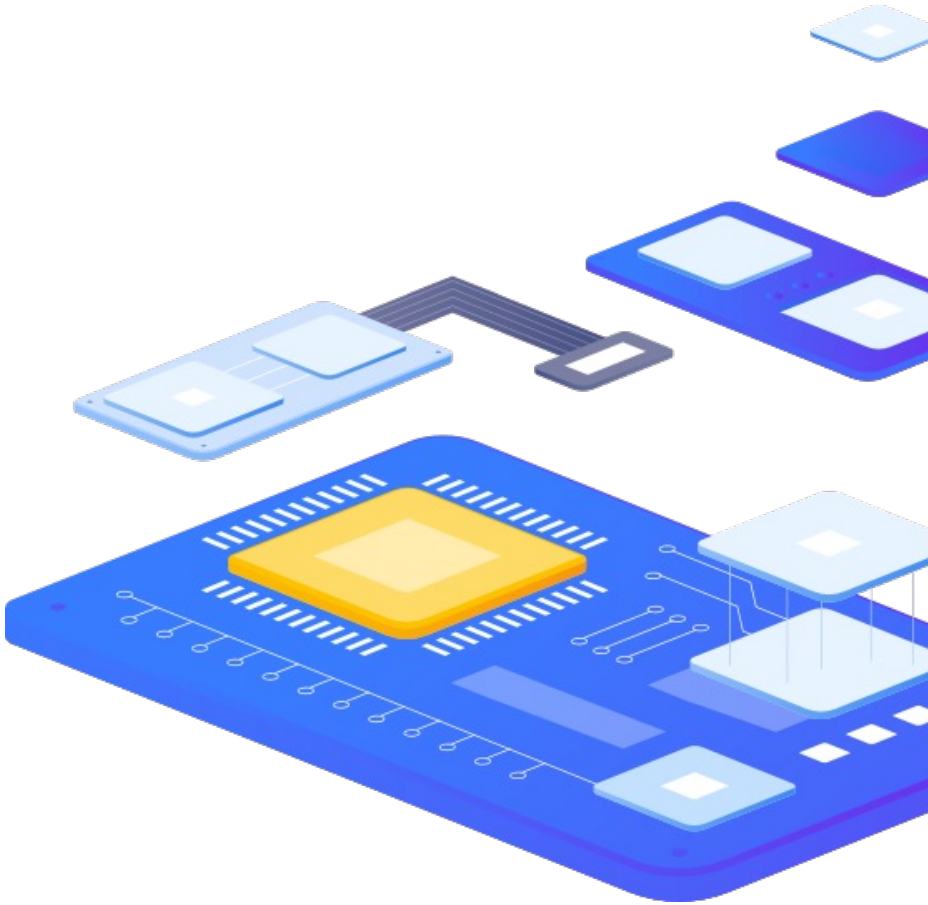


initMAX

Core Components

4. Load Balancer(F5, PgPool, libpg)

- **Routes connections** to appropriate nodes
- **Health checks** via Patroni REST API
- **Eliminates SPOF** when deployed redundantly
- **Supports** read/write splitting
- **Whenever possible** go with driver-based LB

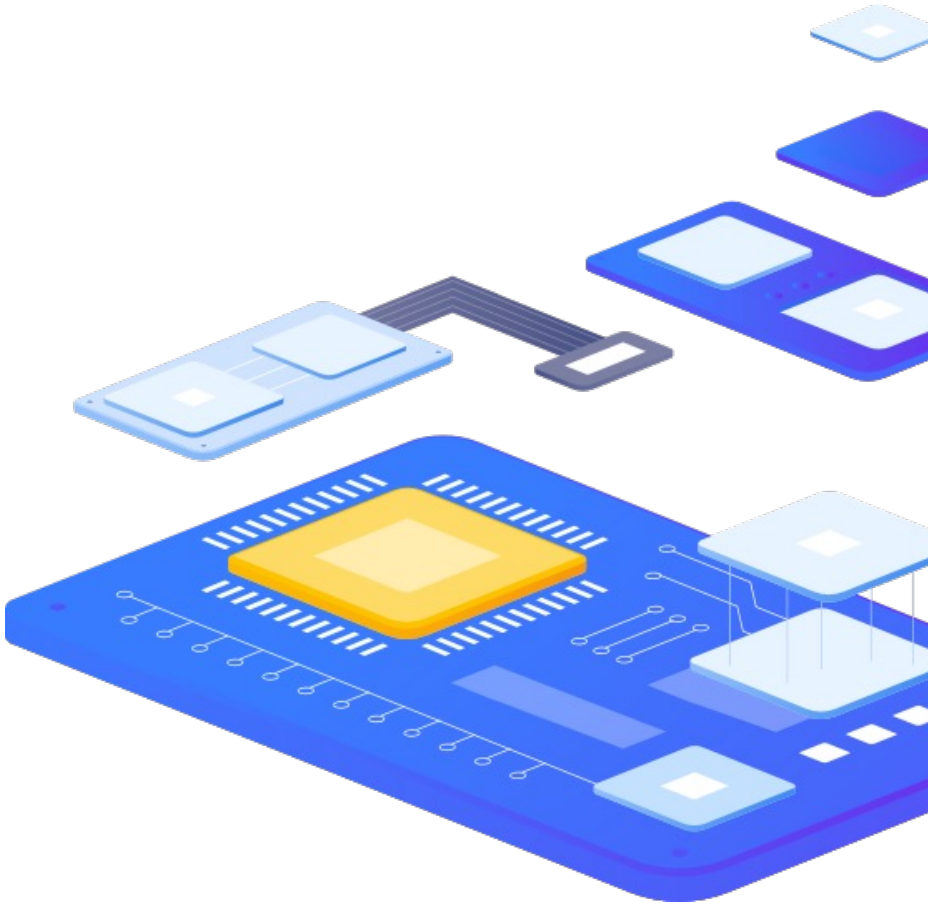


initMAX

Core Components

5. pgBackRest (Backup Solution)

- **Separate backup node** recommended
- **Point-in-time recovery**
- **WAL archiving**
- **Incremental backups**



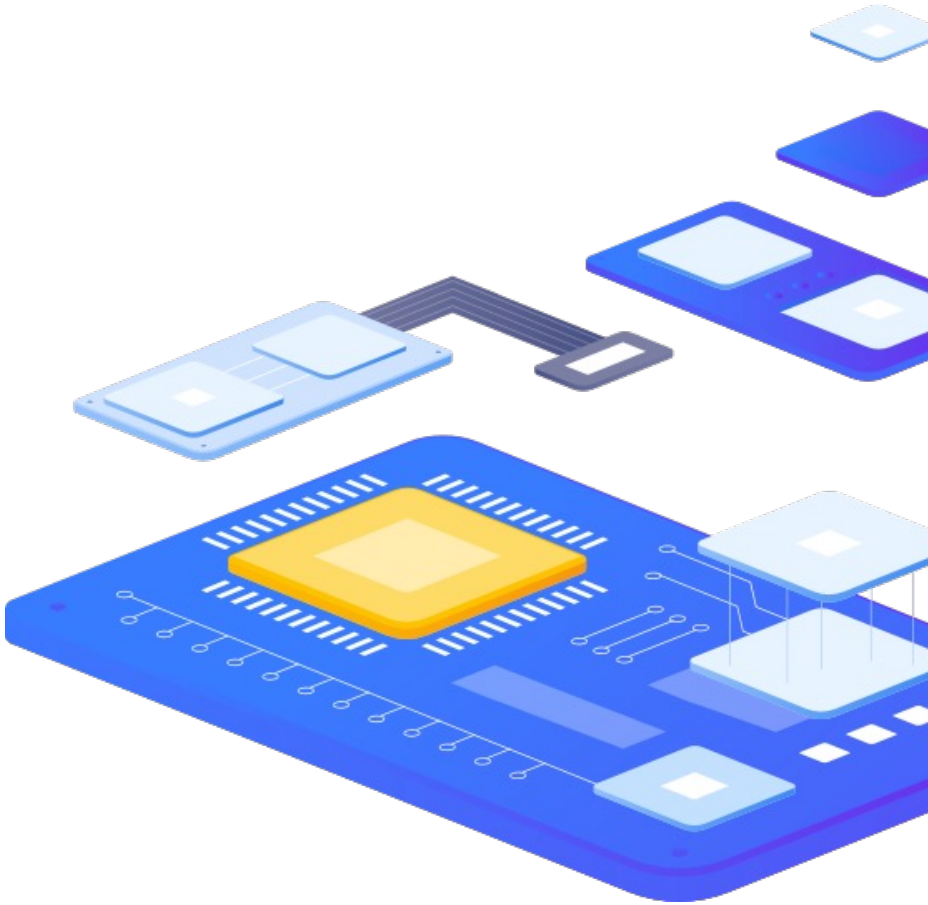
initMAX

Network Requirements

Ports

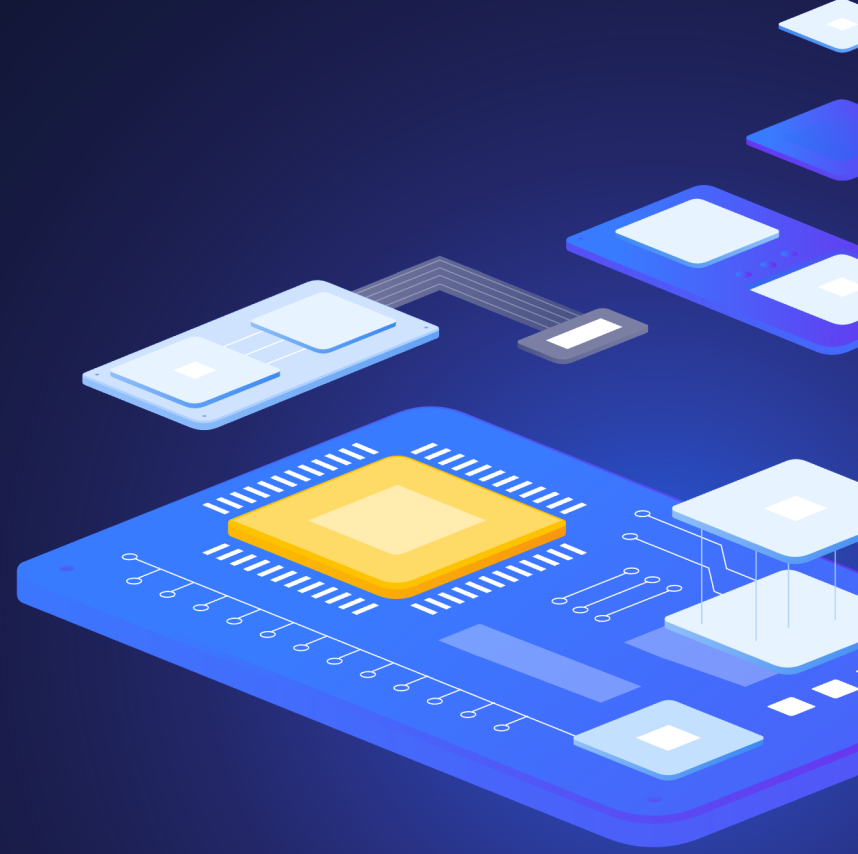
- PostgreSQL: Port 5432
- Patroni REST API: Port 8008
- etcd: Ports 2379 (client), 2380 (peer)
- HAProxy: Ports 5000 (write), 5001 (read)

Better to have separated network interface for replication



4

Installation and Initial Setup



initMAX

Installation and Initial Setup(EL9)

Add repository and disable os postgres repo

```
dnf install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-9-
x86_64/pgdg-redhat-repo-latest.noarch.rpm
dnf -qy module disable postgresql
dnf config-manager --set-enabled pgdg-rhel9-extras
```

Install packages including epel repository

```
dnf install –y epel-release postgresql17-server postgresql17-contrib etcd patroni-
etcd
```

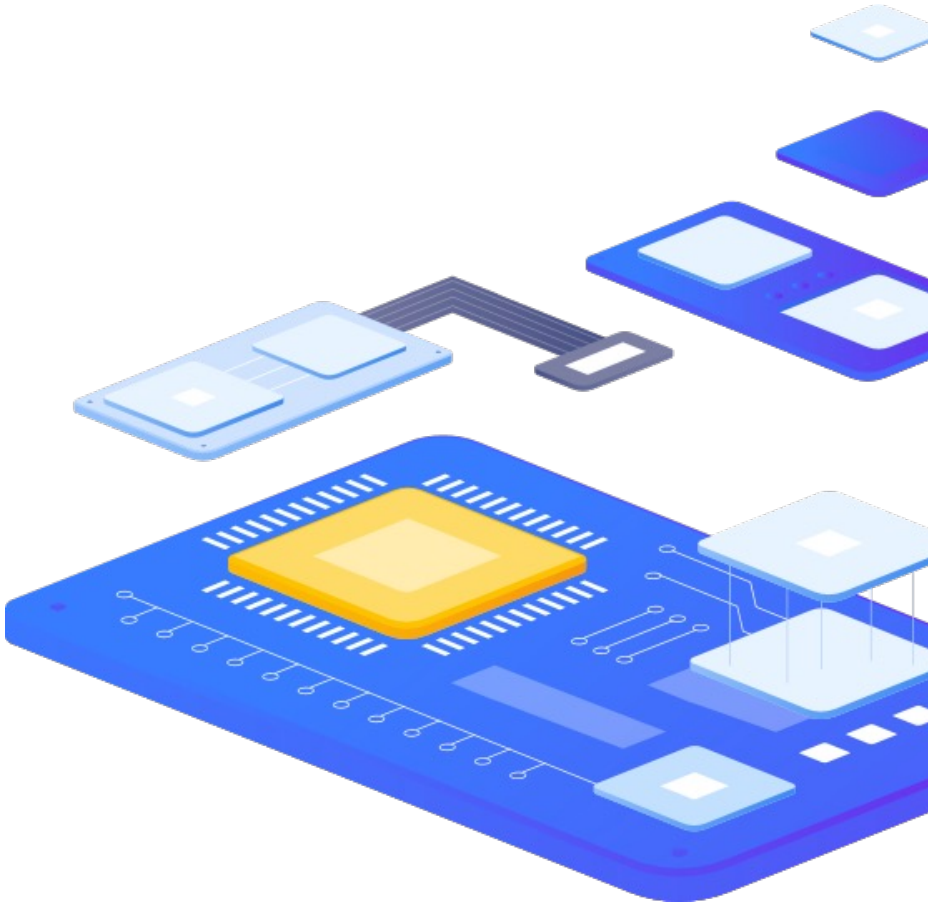
initMAX

etcd

Configure etcd – basic config

```
vim /etc/etcd/etcd.conf
ETCD_NAME=zabbixdb01
ETCD_LISTEN_PEER_URLS="http://10.225.21.44:2380"
ETCD_LISTEN_CLIENT_URLS="http://10.225.21.44:2379,http://127.0.0.1:2379"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://zabbixdb01.initmax.com:2380"
ETCD_INITIAL_CLUSTER="zabbixdb01=http://zabbixdb01.initmax.com:2380,
zabbixdb02=http://zabbixdb02.initmax.com:2380,zabbixdb03=http://zabbixdb03.init
max.com:2380"
ETCD_INITIAL_CLUSTER_STATE="new"
ETCD_INITIAL_CLUSTER_TOKEN="prod-zabbixdb"
ETCD_ADVERTISE_CLIENT_URLS="http://zabbixdb01.initmax.com:2379"
```

```
systemctl enable --now etcd
```

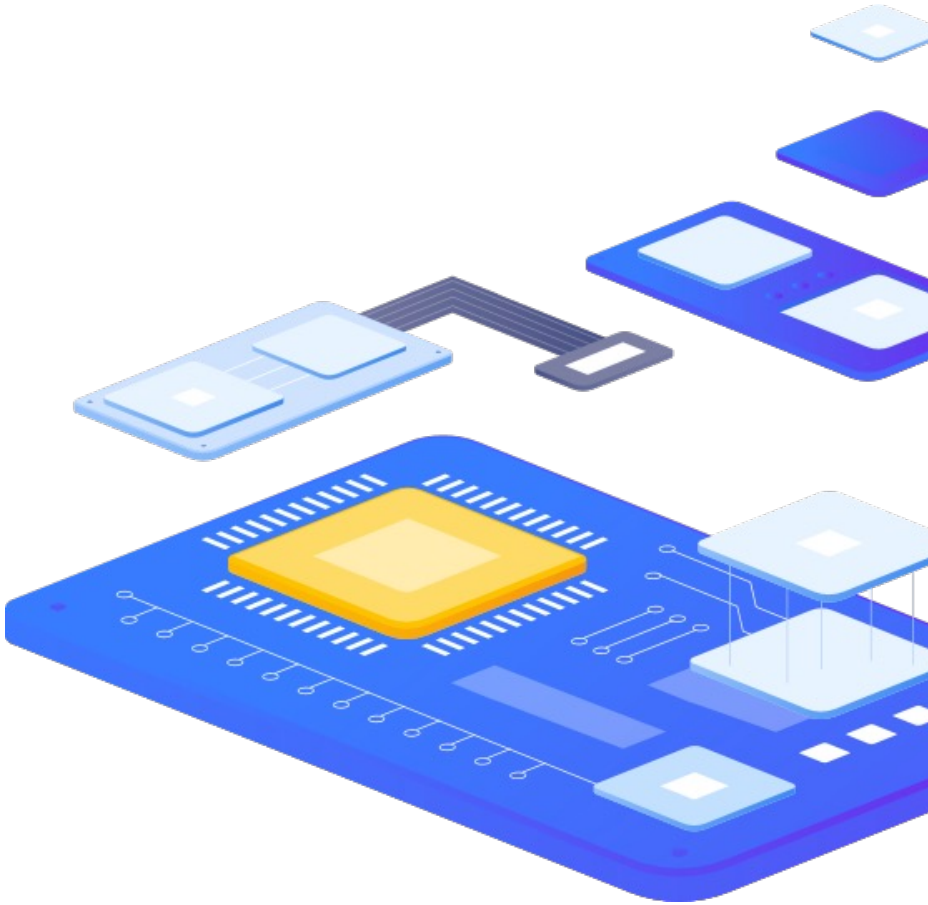


initMAX

Patroni

Configure Patroni

```
vim /etc/patroni/patroni.yml
scope: zabbixdb_cluster
name: zabbixdb01
log:
  ...
restapi:
  ...
etcd3:
  ...
bootstrap:
  ...
post_bootstrap:
  ...
postgresql:
  ...
watchdog:
  ...
tags:
  ...
```

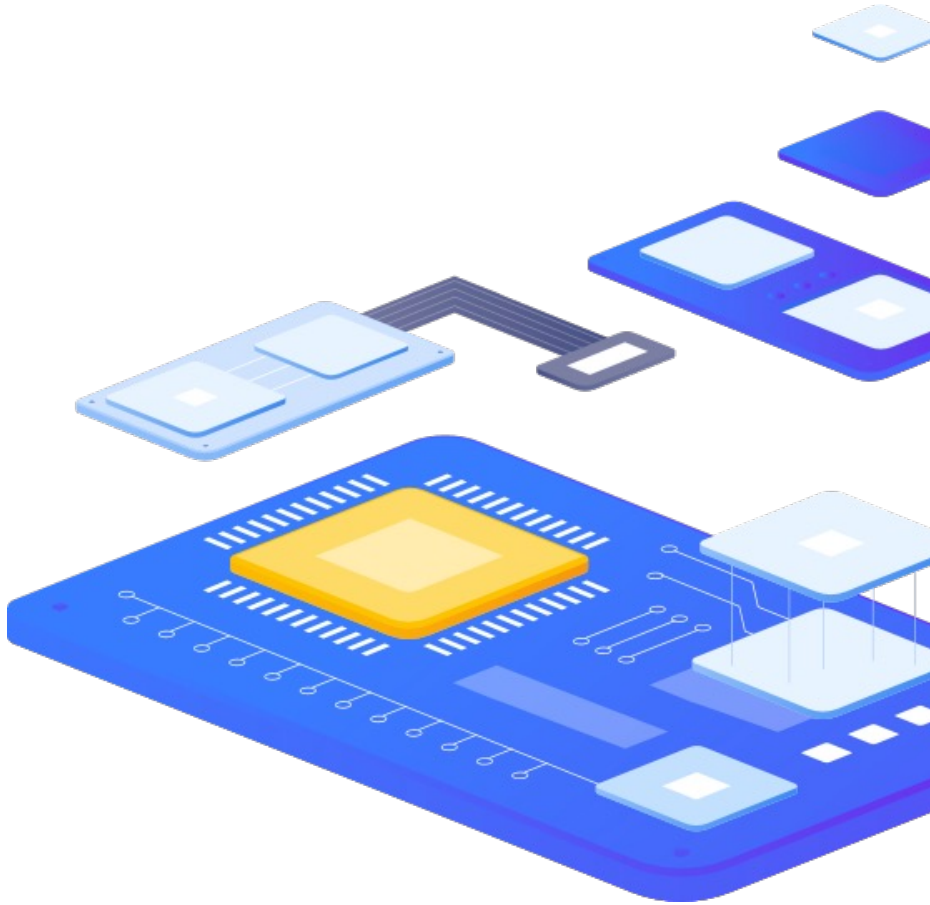


initMAX

Patroni

Log section

```
vim /etc/patroni/patroni.yml
scope: zabbixdb_cluster
name: zabbixdb01
log:
  dir: /var/log/patroni
  level: INFO
  traceback_level: ERROR
  max_queue_size: 1000
  file_num: 10
```

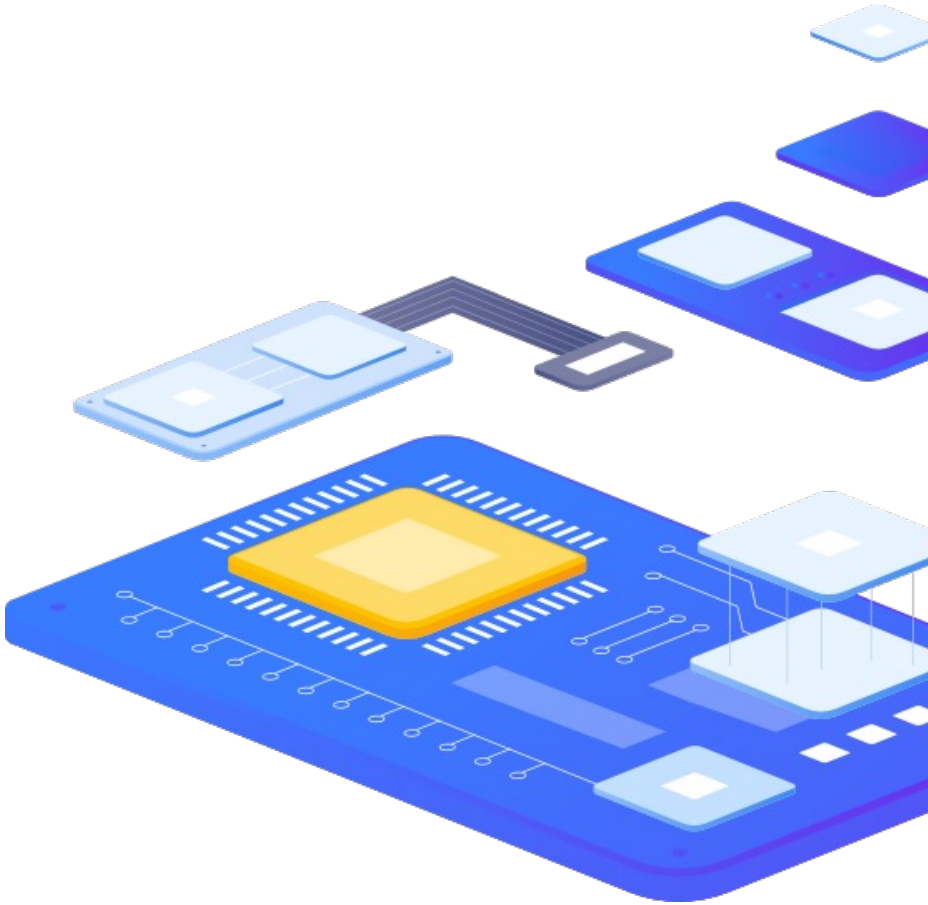


initMAX

Patroni

restapi section

```
vim /etc/patroni/patroni.yml
scope: zabbixdb_cluster
name: zabbixdb01
restapi:
  listen: 0.0.0.0:8008
  connect_address: zabbixdb01.initmax.com:8008
  authentication:
    username: admin
    password: <redacted>
```

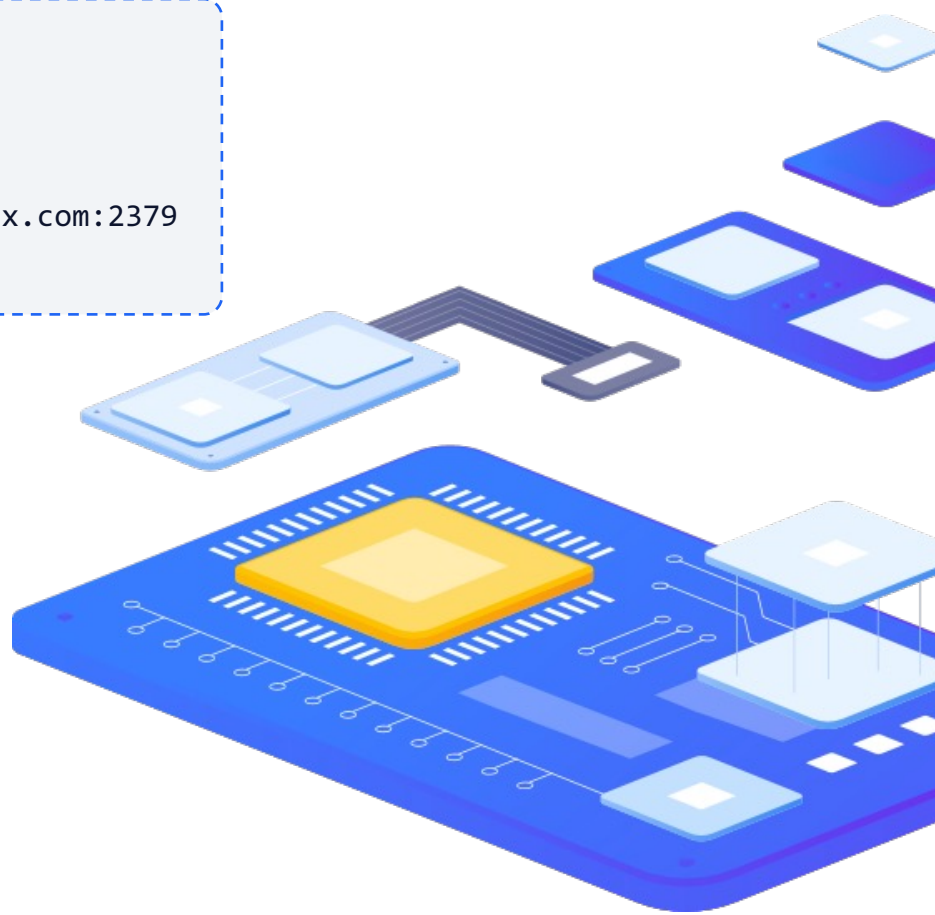


initMAX

Patroni

etcd3 section

```
vim /etc/patroni/patroni.yml
scope: zabbixdb_cluster
name: zabbixdb01
etcd3:
  hosts: zabbixdb01.initmax.com:2379,zabbixdb02.initmax.com:2379,zabbixdb03.initmax.com:2379
  protocol: http
```

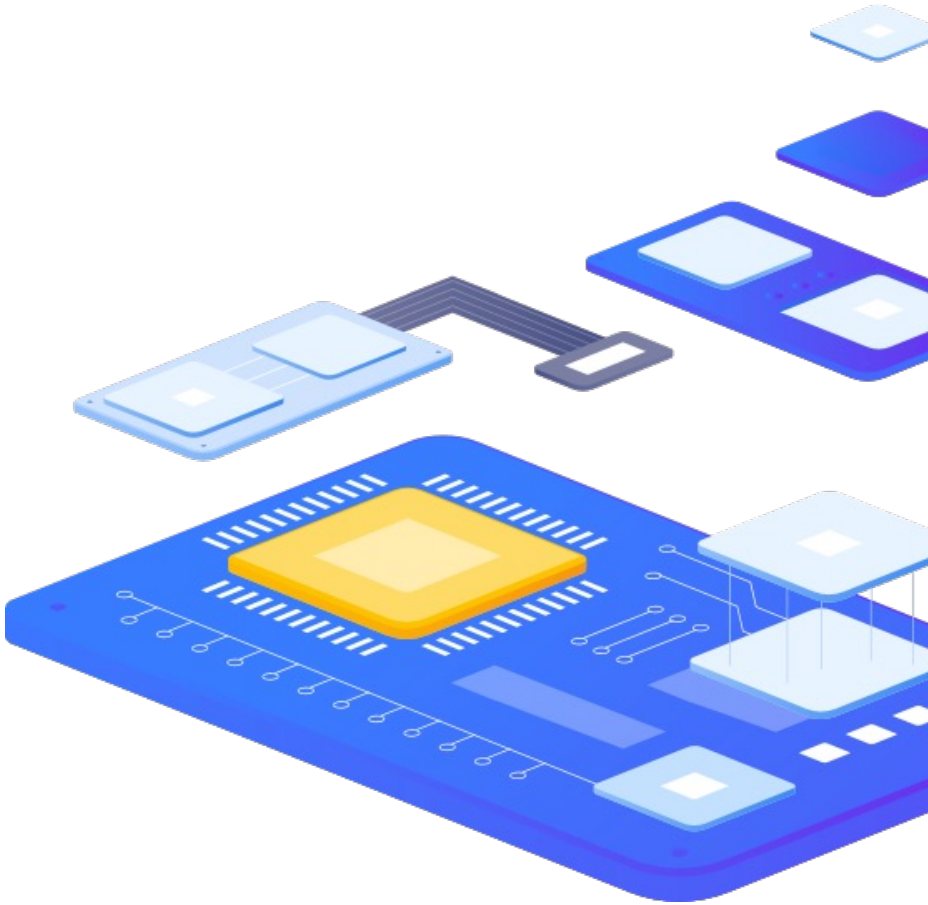


initMAX

Patroni

bootsrap section

```
vim /etc/patroni/patroni.yml
scope: zabbixdb_cluster
name: zabbixdb01
bootstrap:
  dcs:
    ttl: 20                             # Leader key TTL (seconds)
    loop_wait: 5                        # Check interval (seconds)
    retry_timeout: 5                    # DCS operation timeout
    maximum_lag_on_failover: 1048576    # Max lag for failover candidate (bytes)
    failsafe_mode: 'on'                 # Postgres may continue to run when DCS is down
    synchronous_mode: quorum            # Synchronous replication mode
    postgresql:
      use_pg_rewind: true
      remove_data_directory_on_diverged_timelines: true
      remove_data_directory_on_rewind_failure: true
      use_slots: true
      use_unix_socket: true
      parameters:
        max_connections: '160'
        ...
  initdb:
    - encoding: UTF8
    - data-checksums
    - auth-host: scram-sha-256
    - locale: "cs_CZ.UTF-8"
    - locale-provider: "libc"
  pg_hba:
    - local all all  trust
    - host all all 127.0.0.1/32 trust
    - ...
```

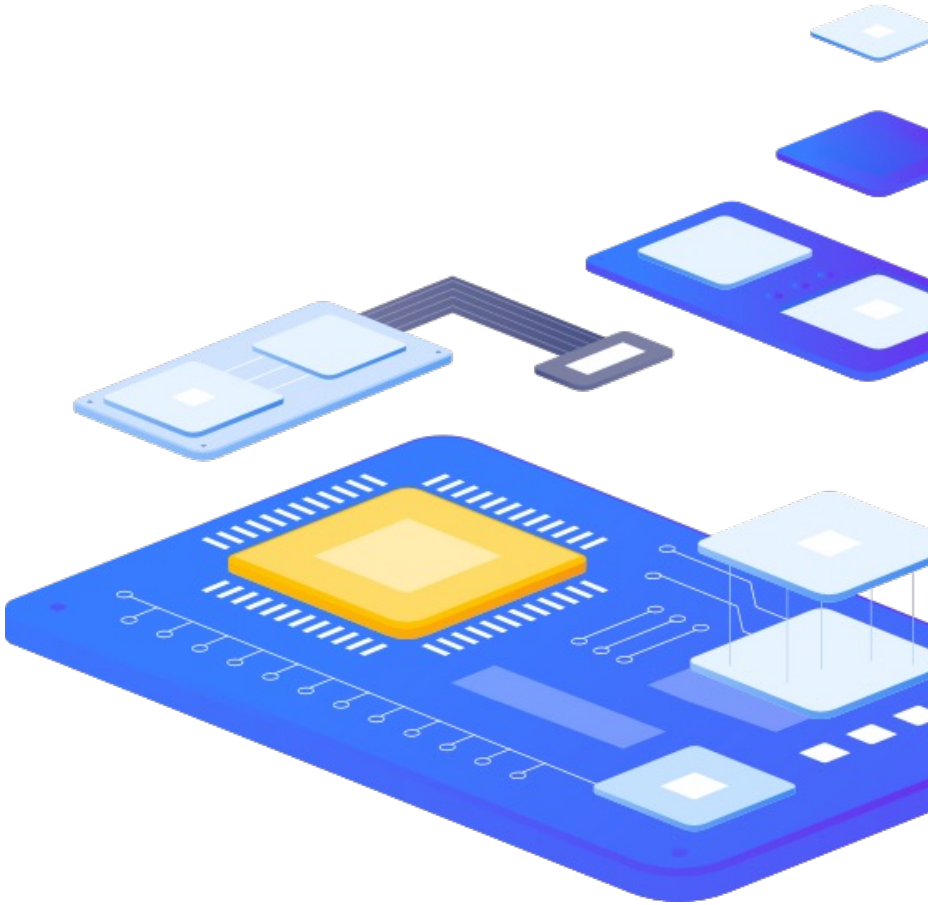


initMAX

Patroni

post_bootsrap section

```
vim /etc/patroni/patroni.yml
scope: zabbixdb_cluster
name: zabbixdb01
post_bootstrap:
  - type: sql
    command: |
      CREATE USER postgres WITH PASSWORD '<redacted>' SUPERUSER;
      CREATE USER repl WITH PASSWORD '<redacted>' REPLICATION;
      CREATE USER rewind_user WITH PASSWORD '<redacted>';
```

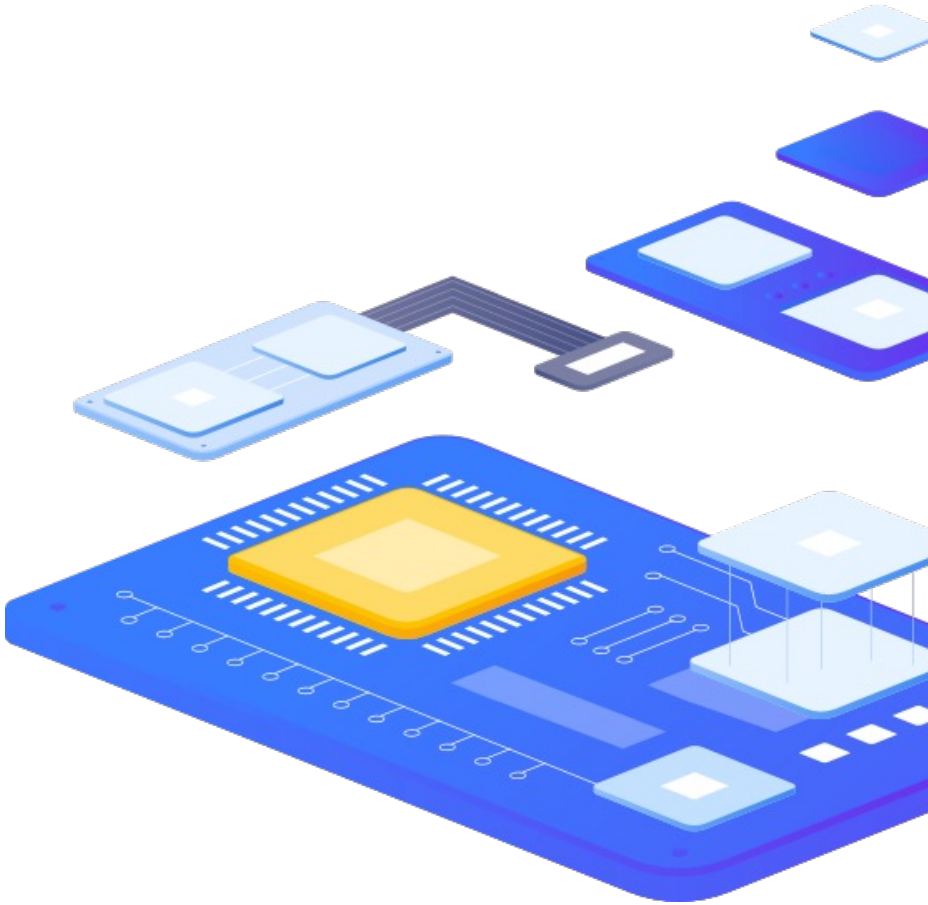


initMAX

Patroni

postgresql section

```
vim /etc/patroni/patroni.yml
scope: zabbixdb_cluster
name: zabbixdb01
postgresql:
  listen: 0.0.0.0:5432
  connect_address: zabbixdb01.initmax.com:5432
  data_dir: /data/patroni
  config_dir: /data/patroni
  bin_dir: /usr/pgsql-17/bin
  authentication:
    replication:
      username: repl
      password: <redacted>
    superuser:
      username: postgres
      password: <redacted>
    rewind:
      username: rewind_user
      password: <redacted>
  parameters:
    unix_socket_directories: '/var/run/postgresql'
  create_replica_methods:
    - basebackup
  basebackup:
    checkpoint: 'fast'
```

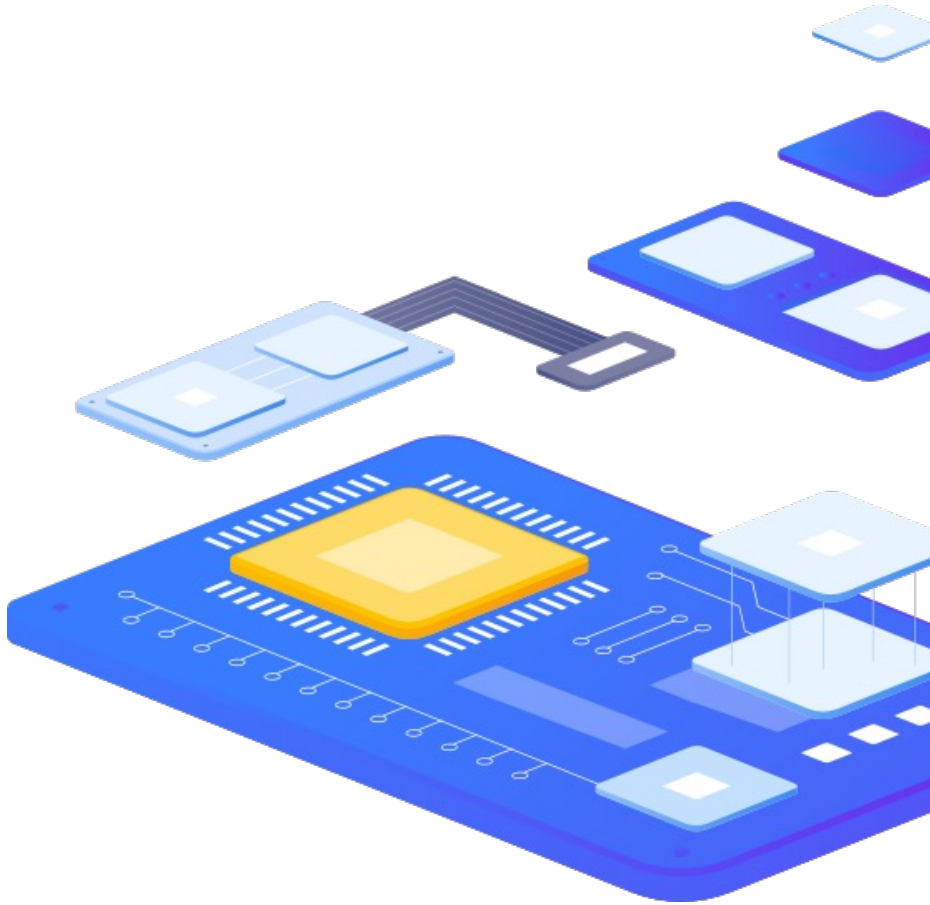


initMAX

Patroni

watchdog section

```
vim /etc/patroni/patroni.yml
scope: zabbixdb_cluster
name: zabbixdb01
watchdog:
  mode: required
  device: /dev/watchdog
  safety_margin: 4
```

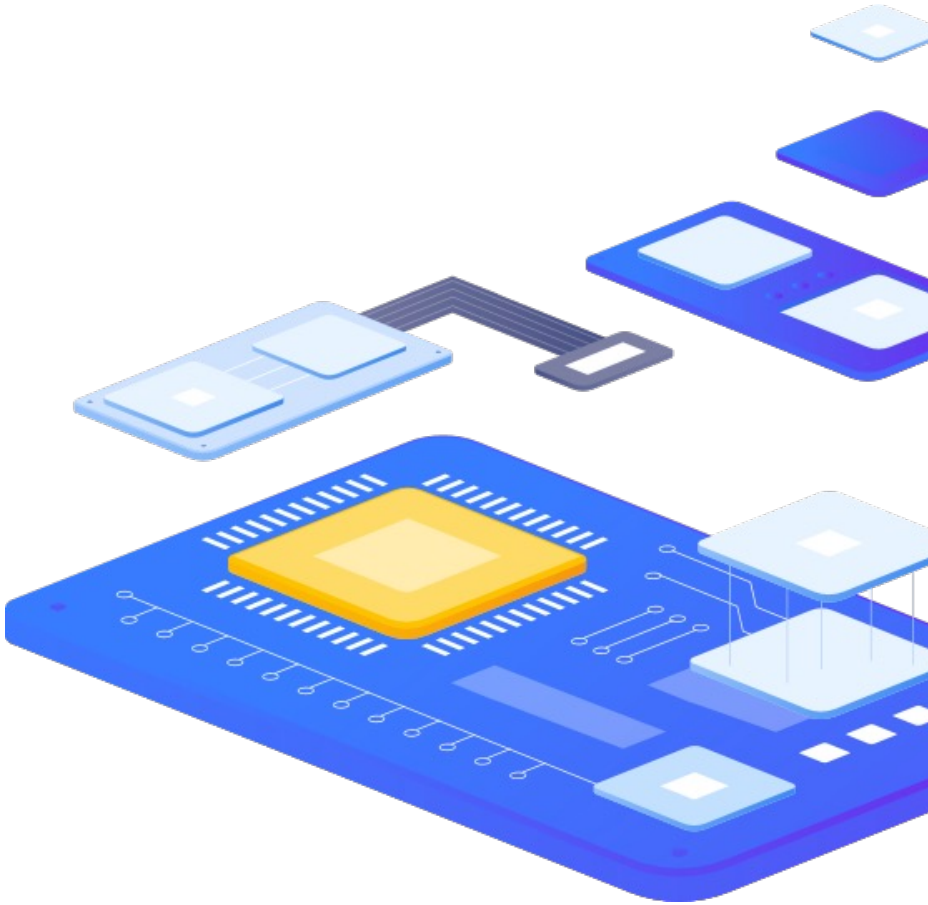


initMAX

Patroni

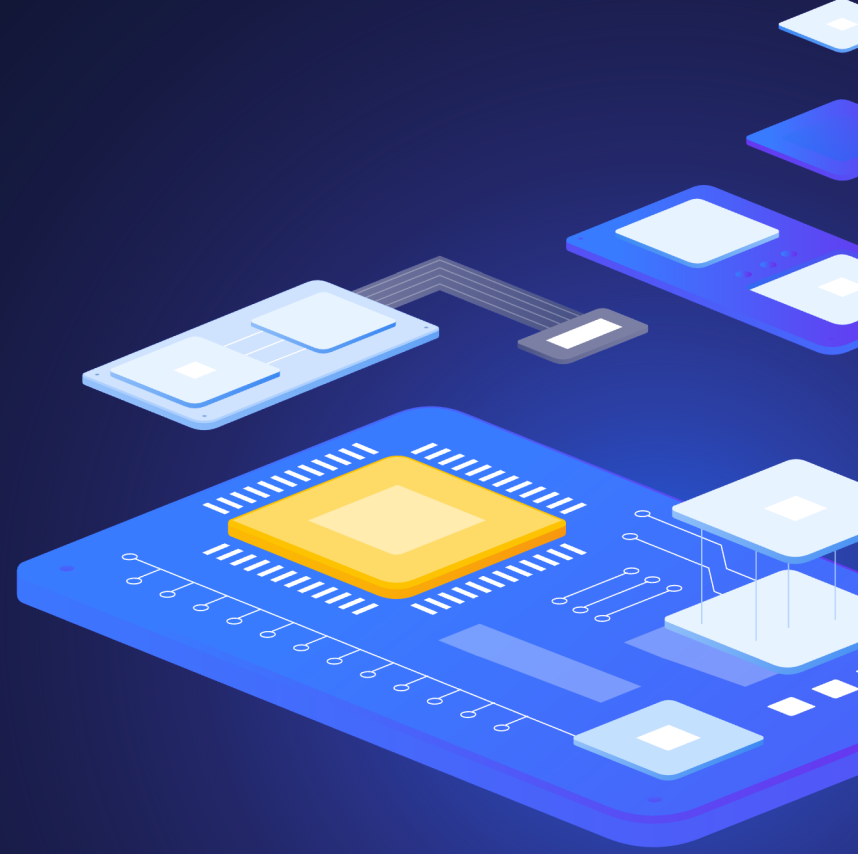
tags section

```
vim /etc/patroni/patroni.yml
scope: zabbixdb_cluster
name: zabbixdb01
tags:
    nofailover: false
    failover_priority: 1
    noloadbalance: false
    clonefrom: true
    nosync: false
```



5

Configuration Management

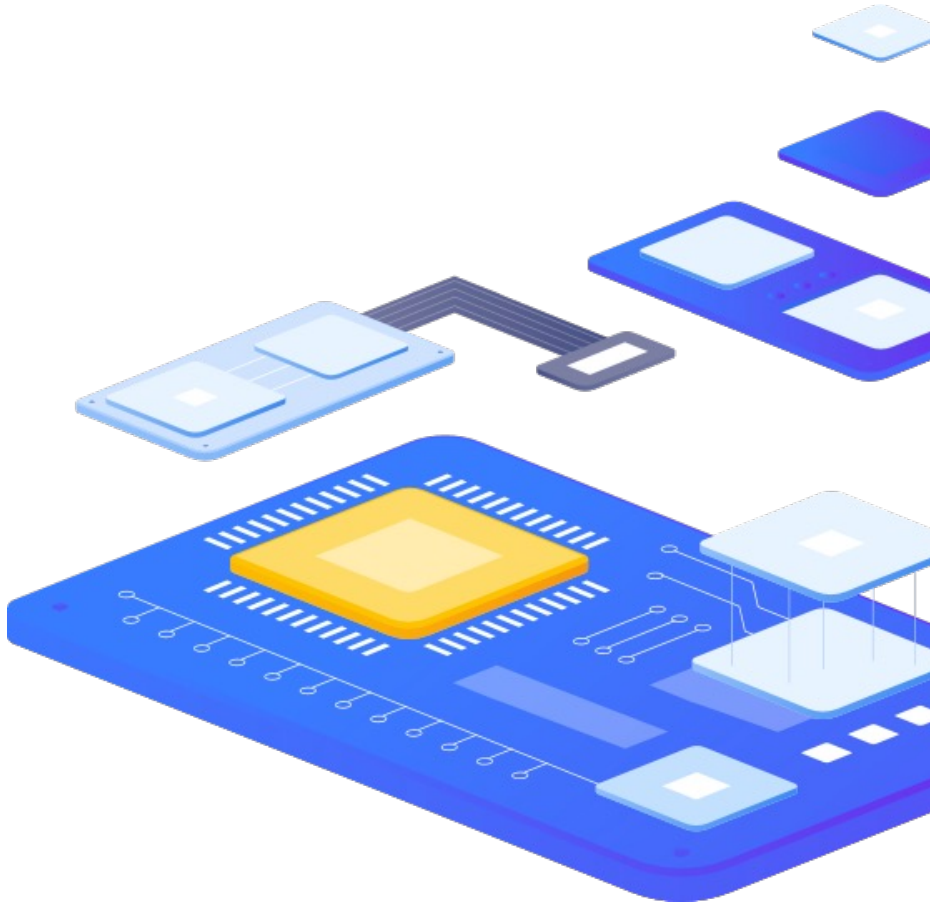


initMAX

Configuration Hierarchy

Patroni uses a hierarchical configuration system:

- **Environment variables** (highest priority)
- **Local YAML configuration file**
- **Dynamic configuration in DCS** (lowest priority)



initMAX

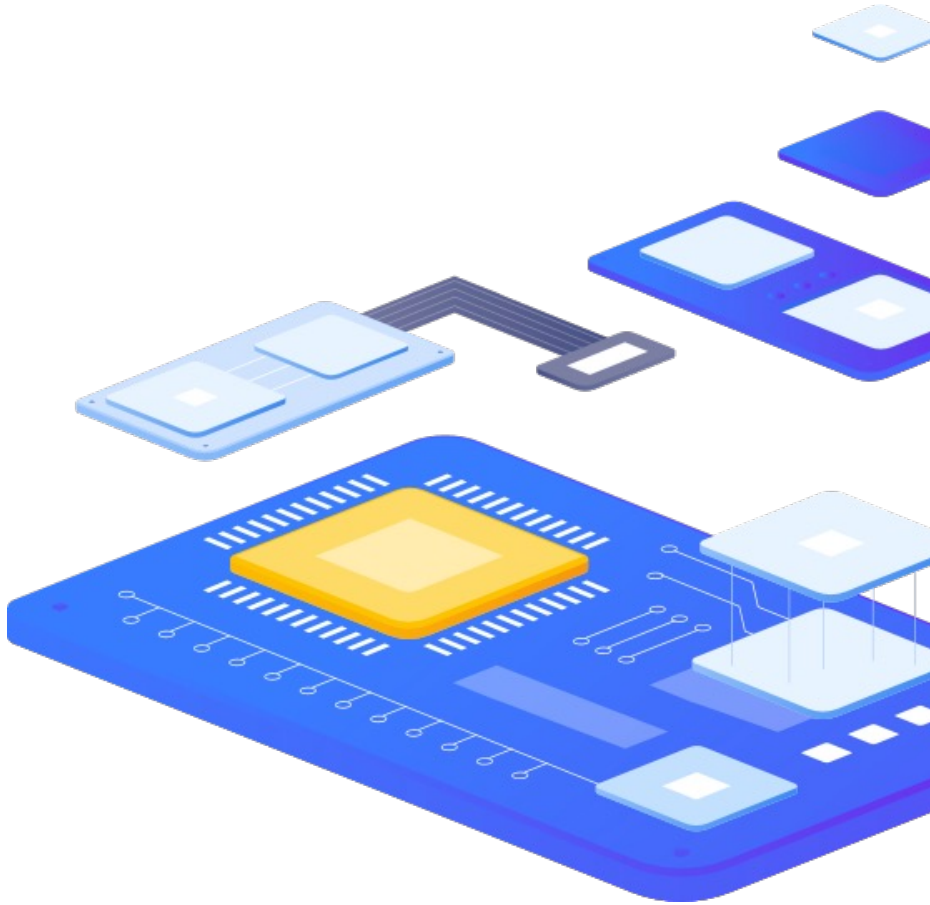
Dynamic Configuration with patronictl

Show Current Configuration

```
patronictl show-config
```

Edit Configuration Interactively

```
patronictl edit-config
```



initMAX

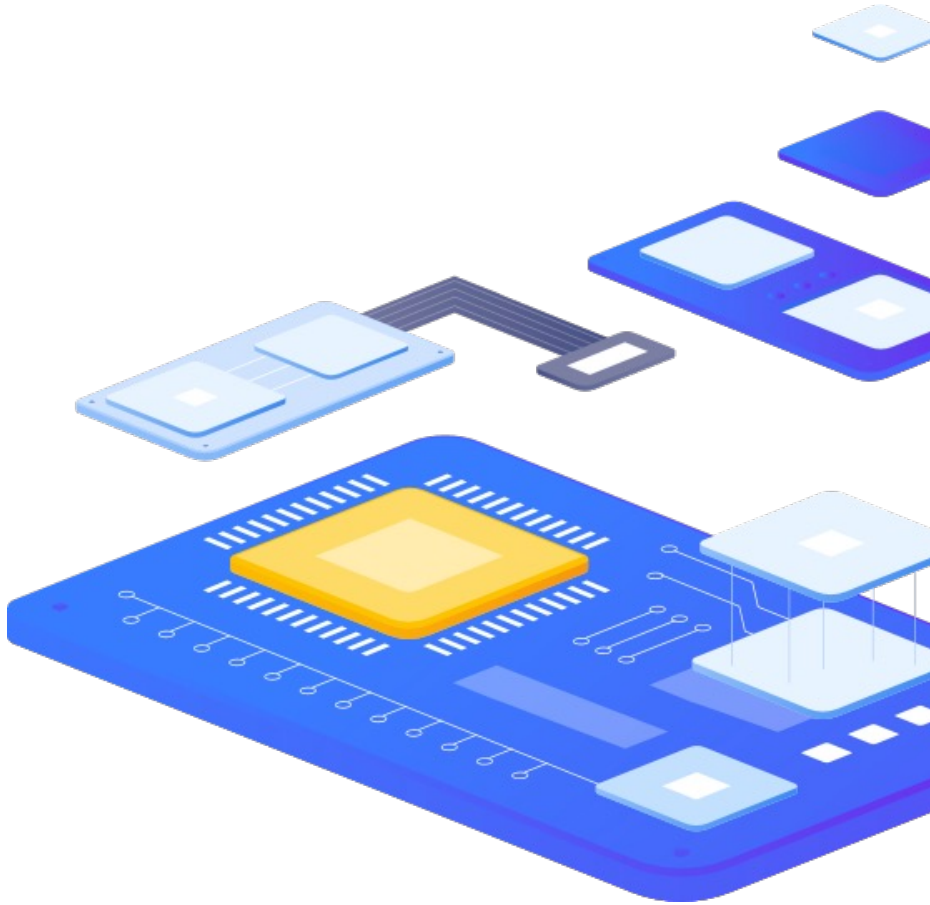
Check Cluster Status

List cluster members

```
patronictl list
```

Extended Cluster Information

```
patronictl list -e
```



initMAX

Cluster management

Reinitialize replica

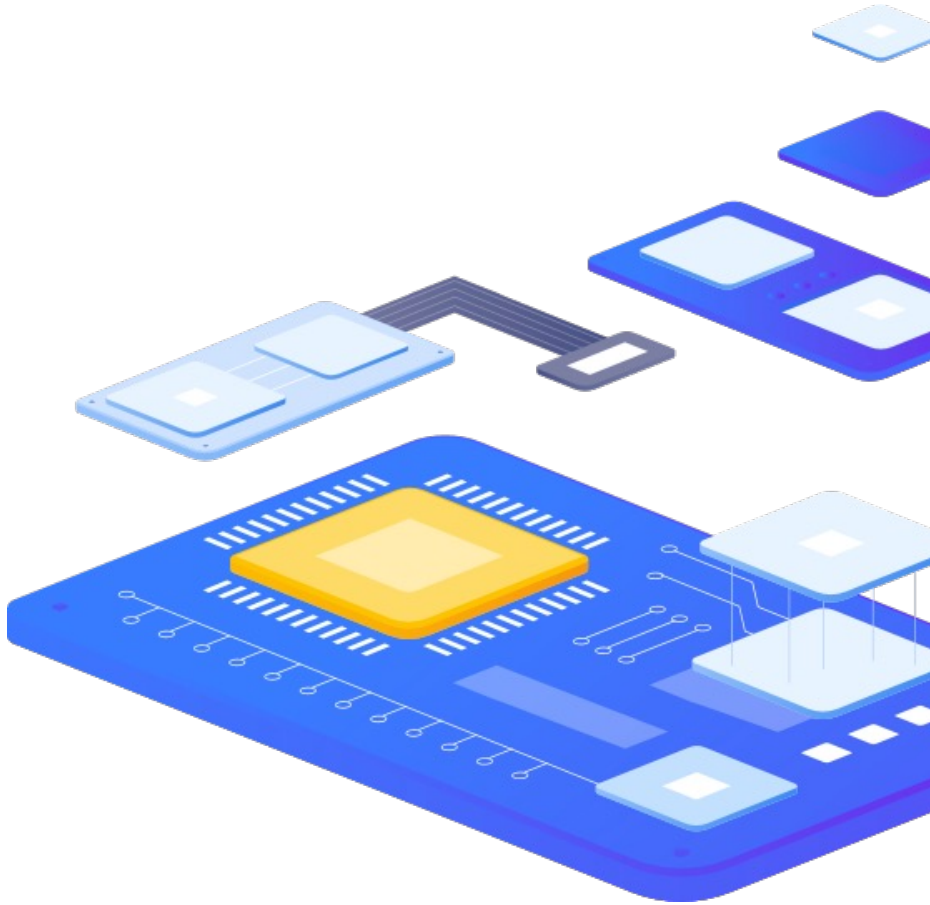
```
patronictl reinit
```

Show log of failover and switchover events

```
patronictl history
```

Perform a switchover (on healthy cluster)

```
patronictl switchover
```



initMAX

Cluster management

Maintenance mode (disables automatic failover)

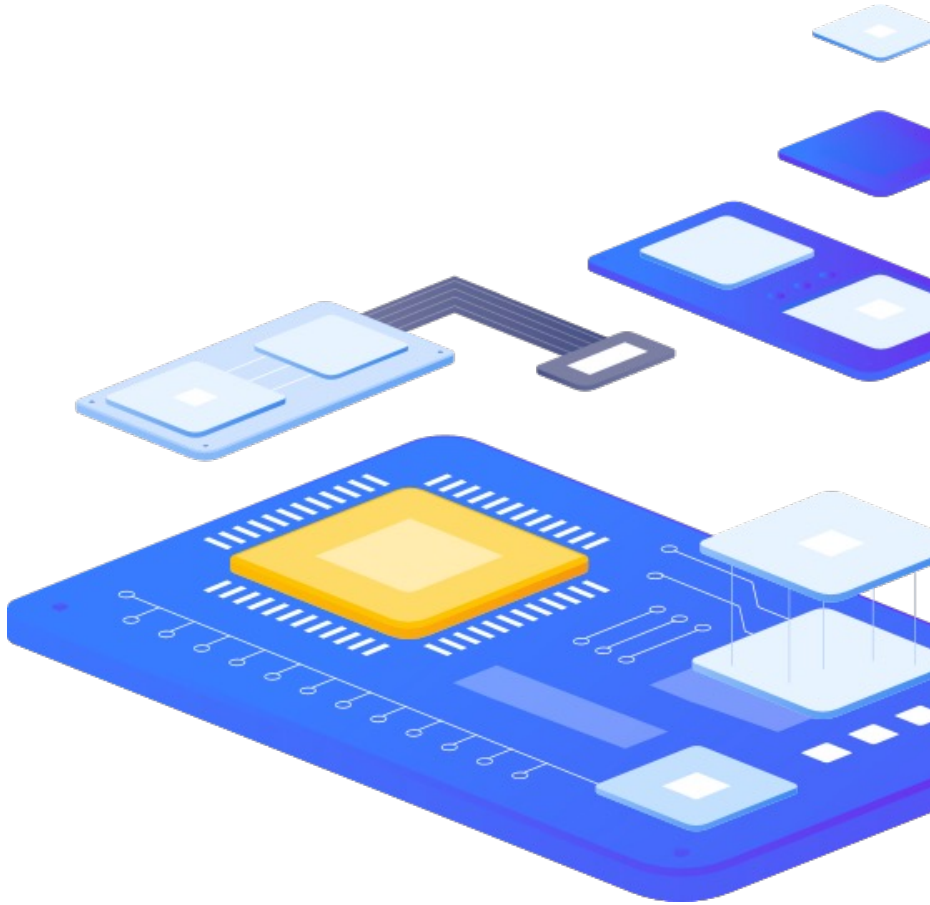
```
patronictl pause
```

Exit maintenance mode (enables automatic failover)

```
patronictl resume
```

Wipe cluster information from the DCS

```
patronictl remove
```



initMAX

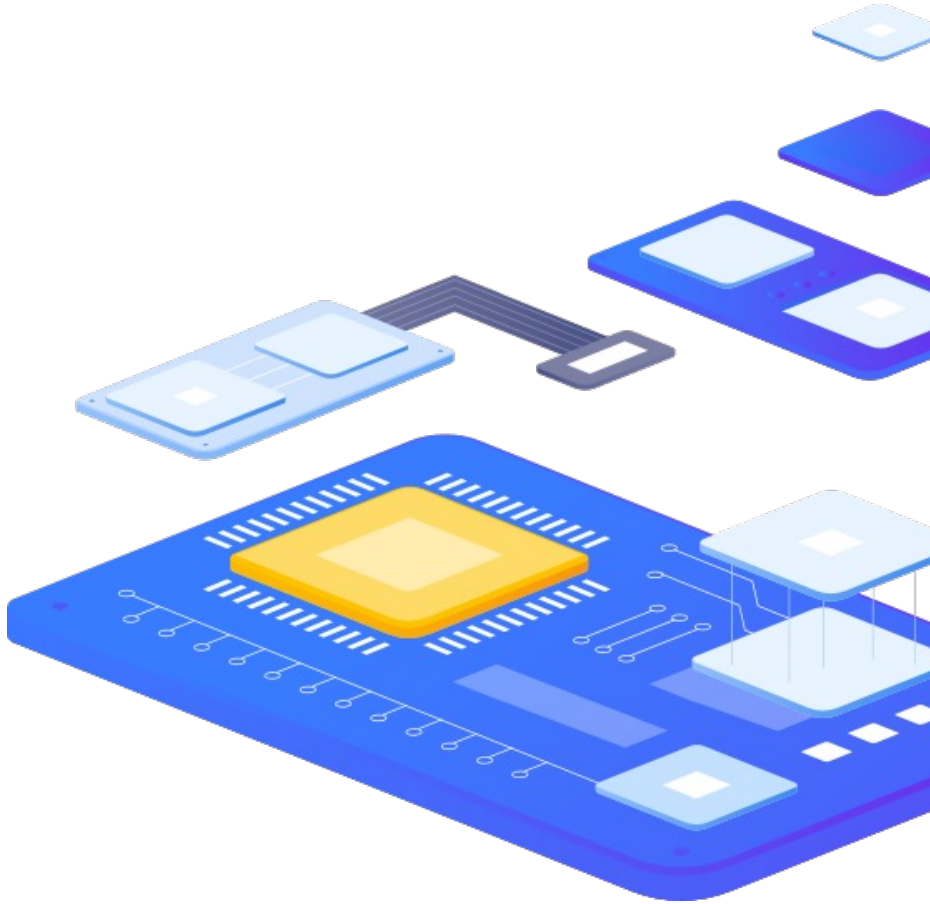
Cluster management

Reload of local configuration

```
patronictl reload
```

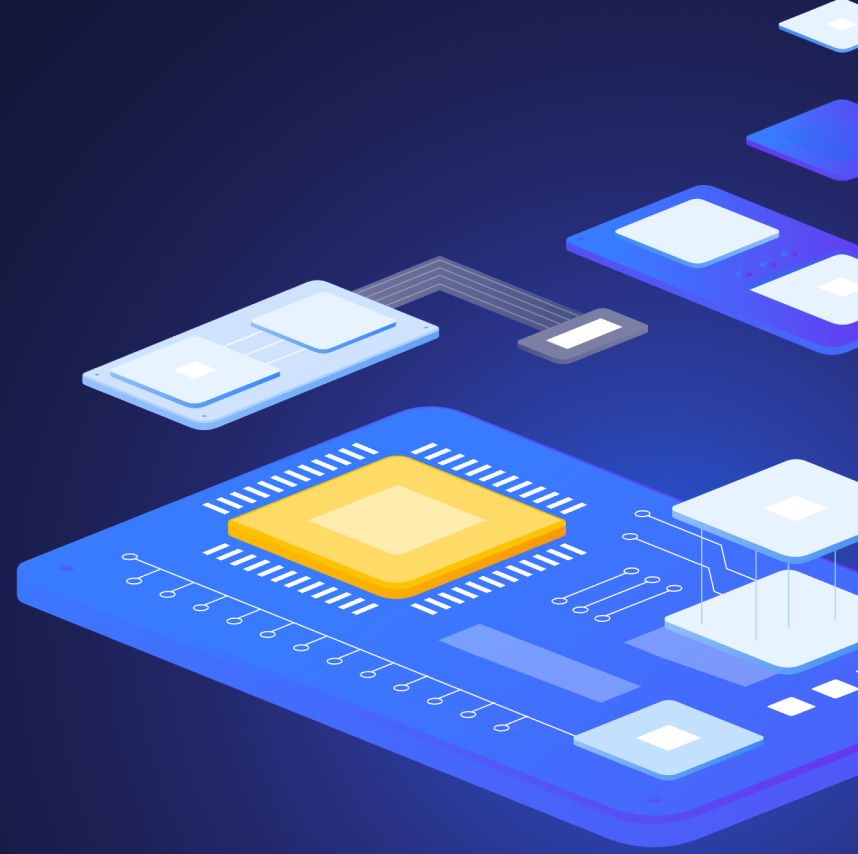
Restart Postgres instance

```
patronictl restart
```



6

API



initMAX

API calls for LB integrations

Primary node

```
curl -o /dev/null -s -w "%{http_code}" http://zabbixdb01.initmax.com:8008/leader
```

Replica node

```
curl -o /dev/null -s -w "%{http_code}" http://zabbixdb01.initmax.com:8008/replica
```

Explanations for HTTP status codes:

200 OK – The node is healthy and matches the endpoint condition

(e.g., /leader → this node is the Primary; /replica → this node is a Replica).

503 Service Unavailable – The node is reachable but doesn't meet the condition or is not ready yet

(e.g., hitting /leader on a replica).

initMAX

API call for monitoring

Example call on endpoint /patroni

```
curl -s http://zabbixdb01.initmax.com:8008/patroni | jq .
```

Data example:

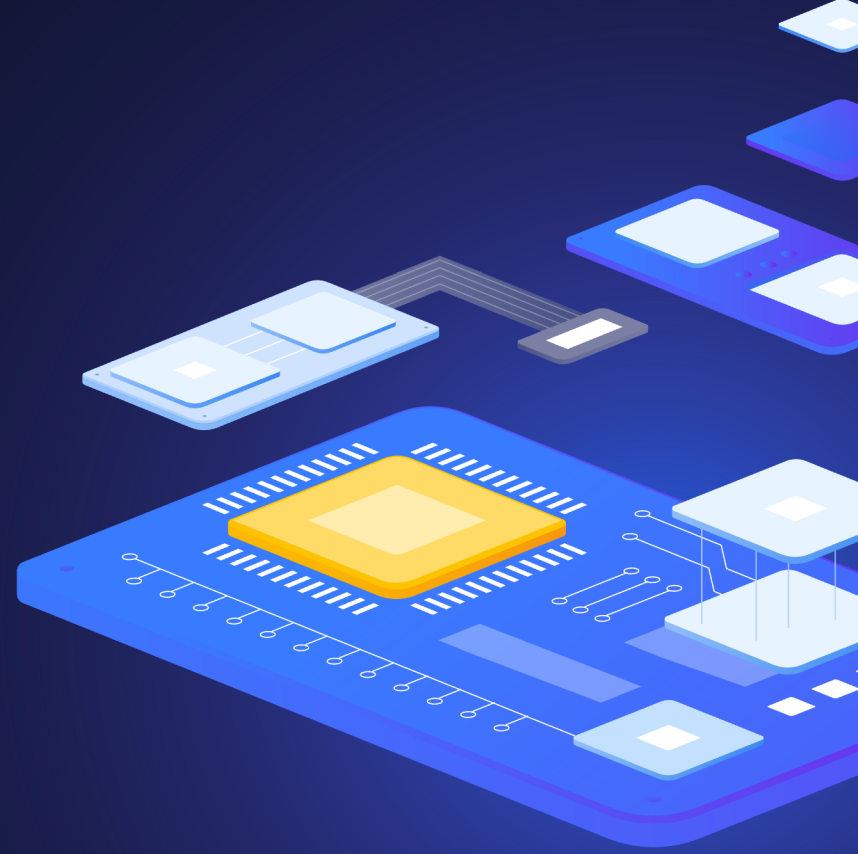
```
{
  "state": "running",
  "role": "primary",
  "timeline": 12,
  "xlog": {
    "location": 123456789
  },
  "patroni": {
    "version": "4.1.0",
    "scope": "zabbixdb_cluster"
  },
  ...
}
```

Key fields explained:

- **state –** whether PostgreSQL is up and running (running, stopped, …).
- **role –** shows if this node is primary or replica.
- **timeline –** timeline number, must be consistent across nodes.
- **xlog –** WAL location and lag → crucial for replication health.
- **version –** Patroni agent version running on the node.
- **scope –** the cluster name (important when multiple Patroni clusters exist).

initMAX

Questions?



PostgreSQL HA using Patroni

Certified training and Support

Basic certified courses

SQL Basics

This training has been designed for people who want to get familiar with SQL. You will learn how SQL works and how to write proper SQL statements using practical examples that will be useful for your daily work.

4 DAYS

€ 2,000

Excluding VAT

Requirements: None
Available online: Yes
Certification: Yes

Course dates:

On request | 13.–16. 1. 25 | 12.–15. 5. 25 | 3.–6. 11. 25

REGISTER

More info and dates

Introduction to PostgreSQL

This workshop has been designed for people who want to get familiar with SQL and PostgreSQL. You will learn how to use PostgreSQL and how to write proper SQL statements.

4 DAYS

€ 2,000

Excluding VAT

Requirements: None
Available online: Yes
Certification: Yes

Course dates:

On request | 23.–26. 9. 24 | 27.–30. 1. 25 | 26.–29. 5. 25 | 24.–27. 11. 25

REGISTER

More info and dates

Advanced certified courses

PostgreSQL Professional

This course provides a deep insight into advanced PostgreSQL topics like indexing, storage parameters, optimization, replication, monitoring and many more.

3 DAYS

€ 1,500

Excluding VAT

Requirements: None
Available online: Yes
Certification: Yes

Course dates:

On request | 7.–9. 10. 24 | 10.–12. 2. 25 | 9.–11. 6. 25 | 8.–10. 12. 25

REGISTER

More info and dates

PostgreSQL – administration and performance tuning

This course is perfectly suitable for database administrators and sysadmins, dealing with topics related to administration and performance tuning.

4 DAYS

€ 2,000

Excluding VAT

Requirements: None
Available online: Yes
Certification: Yes

Course dates:

On request | 21.–24. 10. 24 | 24.–27. 2. 25 | 23.–26. 6. 25

REGISTER

More info and dates

PostgreSQL – High Availability & Patroni

This course is intended for PostgreSQL users who are interested in fully automating high availability operations. The course first gives an overview of the general high availability landscape in PostgreSQL clusters and then focuses on installation, configuration and fully automated operation of very popular (open source) cluster manager "Patroni".

3 DAYS

€ 1,500

Excluding VAT

Requirements: advanced knowledge of PostgreSQL and OS Linux
Available online: Yes
Certification: Yes

Course dates:

On request | 4.–6. 11. 24 | 10.–12. 3. 25 | 8.–10. 9. 25

REGISTER

More info and dates

initMAX

Certified training and Support

Expert certified courses

PostgreSQL in Kubernetes

This course provides an introduction to Kubernetes itself and to Kubernetes resources, which are needed in order to manage PostgreSQL.

3 DAYS

€ 1,500

Excluding VAT

Requirements: None
Available online: Yes
Certification: Yes

Course dates:

On request | 18.–20. 11. 24 | 24.–26. 3. 25 | 22.–24. 9. 25

REGISTER

More info and dates

Introduction to PostGIS

This course provides an introduction to PostGIS and its most important features and capabilities. Along with gaining theoretical knowledge, participants will work with real-world datasets to deepen and reinforce their practical skills.

3 DAYS

€ 1,500

Excluding VAT

Requirements: advanced knowledge of PostgreSQL
Available online: Yes
Certification: Yes

Course dates:

On request | 2.–4. 12. 24 | 7.–9. 4. 25 | 6.–8. 10. 25

REGISTER

More info and dates

PostgreSQL – Migration tutorial

This training has been designed for people who want to switch to PostgreSQL.

4 DAYS

€ 2,000

Excluding VAT

Requirements: advanced knowledge of PostgreSQL
Available online: Yes
Certification: Yes

Course dates:

On request | 16.–19. 12. 24 | 22.–25. 4. 25 | 20.–23. 10. 25

REGISTER

More info and dates

initMAX

Certified training and Support

DBA – PostgreSQL

PostgreSQL is a powerful open-source object–relational database management system (ORDBMS). It is used for application development, data warehousing, analysis and other data-intensive tasks. Key features of PostgreSQL include a powerful engine, support for advanced data types and indexing methods, and support for stored procedures and triggers written in various programming languages, including PL/pgSQL, Tcl, and Python. Furthermore, PostgreSQL supports multiversion concurrency control (MVCC), allowing multiple users to access the same data simultaneously without conflicts, and offers robust data integrity and security support.

I AM INTERESTED IN THIS SERVICE →

Precise database management backed by experience

Reliable monitoring and notification system

Stability, availability and scalability

initMAX

Contact us:

Phone:	+420 800 244 442
Web:	https://www.initmax.cz
Email:	tomas.hermanek@initmax.cz
LinkedIn:	https://www.linkedin.com/company/initmax
Twitter:	https://twitter.com/initmax
Tomáš Heřmánek:	+420 732 447 184