CONTENTS

How To Choose a Redundancy Plan To Ensure High Availability

Published on November 9, 2013

MySQL Security Scaling PostgreSQL Conceptual

By Justin Ellingwood

Introduction

In the first part of this article, you explored different solutions for backing up data on your VPS.

This article will explore the idea of redundancy. Redundant data is not a backup, but can give you a failover in case your primary method of accessing data becomes unavailable.

How you choose to implement replication on your system mostly depends on how you

This might seem similar to a mirrored RAID array, and in some ways, it is. The difference lies in where the replication takes place. With RAID, the redundancy takes place below the application level. The RAID card or RAID software manages the physical storage devices and presents the application with a single apparent device.

DRBD, on the other hand, is configured in a completely different way. In a DRBD set up, each hardware stack is completely mirrored. The application interface is also mirrored. This means that an application failure can be dealt with because there is a completely separate machine with a copy of the data to work with. If your first server has a power supply failure, the second server will still operate effectively.

SQL Replication

If your important data is stored in an SQL database (MySQL, MariaDB, PostgreSQL, etc.), you can take advantage of some built-in replication features. These can be used to provide a failover system in case the main server goes down.

Master-Slave Replication

The most basic kind of SQL replication is a Master-Slave configuration. In this scenario, you have a main database server, which is referred to as the “master” server. This server is responsible for performing all writes and updates. The data from this server is copied continuously to a “slave” server. This server can be be read from, but not written to.

This s allows you to distribute the reads across multiple machines, which can dramatically improve your application’s performance.

While this performance increase is an advantage, one of the main reasons you may set up master-slave replication is for handling failover. If your master server becomes unavailable, you can still read from your slave server. Furthermore, it is possible to convert the slave into a master server in the event that your master is offline for an extended period of time.

Master-slave replication is, in fact, one area where we begin to see how redundancy and backups can complement each other. In a master-slave configuration, you can replicate data from the master to the slave. You can then temporarily disable replication to maintain a consistent state of information on the slave. From here, you can back up the database using whatever backup mechanism is appropriate.

To learn more about how to configure MySQL master-slave replication, click here. To learn about how to accomplish master-slave replication with PostgreSQL, follow this link.

Master-Master Replication

A second form of replication is called Master-Master replication. This configuration allows both servers to have “master” abilities. This means that each server can accept writes and updates and will transfer the changes to the opposite server. This configuration inherits the advantages of the master-slave setup, but also benefits from increased write performance if the writes are properly distributed by a load balancing mechanism.

This also means that, in the event that one server goes down, the other is still up and ready to accept requests. While the configuration is more complicated, the failover in the event of a problem is less complicated than the master-slave redundancy, because the slave database does not need to transform into the master.

This configuration can also be combined with a backup mechanism if you take one of the master servers offline. You must maintain a static database for backups to function correctly, so you have to ensure that no data is being modified or written to until after the backup is complete.

For more information about how to configure master-master replication, click here.

Distribution as a Redundancy Alternative

Distributed systems offer many of the advantages of traditional redundant setups.

You already looked at mirrored RAID levels above (RAID 1). Another common RAID level is RAID 5. This RAID distributes data across a number of drives and also writes the parity of th[illegible]ta across each drive. This means that any kind of transaction can be “rebuilt” from c[illegible]ning the parity information on the other drives if a single drive dies.

This provides a different way of distributing data across disks, and can handle a single disk failure as well.

Distributed systems do not have to be hardware based. There are also quite a few databases and other software solutions designed with distributed data as a key feature.

An example of this is Riak, a distributed database. Riak nodes are all the same. There is no master-slave relationship between the different pieces. Objects stored in the database are replicated, so there is traditional redundancy in that respect.

However, each node does not each hold the entire database. Instead the data is distributed among the nodes in an even manner. The objects that are replicated are placed on different nodes to allow for availability in case of hardware failures.

Another great example of this concept implemented in a database is Cassandra. It is based on the same principles as Riak, but is implemented in a different way.

Conclusion

As you can see, there are many options for redundancy, each with their advantages and faults. It mainly depends on what problems you are trying to anticipate and which parts of your system you consider a failure unacceptable. As you can imagine, a combination of some of these techniques will always yield a more comprehensive safety net.

You should also be able to see by this point that redundant setups are not a substitute for backups. The need for redundant systems to be always operational and always mirroring changes can easily allow data to be destroyed from all points of your configuration.

For applications where data integrity and access are essential, both redundancy and backups are invaluable. Proper implementation will ensure that your product is always available to your users and that important data will never be lost.

About the authors

Justin Ellingwood Author

Still looking for an answer? Ask a question

Search for more help

Was this helpful? Yes No

Comments

6 Comments

Leave a comment...

This textbox defaults to using **Markdown** to format your answer.

You can type **!ref** in this text area to quickly search our full set of tutorials, documentation & marketplace offerings and insert the link!

Sign In or Sign Up to Comment

Dennis Akinwonjowo • November 8, 2022

Concerning the master slave set up. Is replication automatic like the RAID 1? Or do you at a point enable replication? Cause if the former is true then the master slave would inherit the disadvantages of RAID 1 which loses data once deleted from the master DB.

Reply

Kamal Nasser • January 19, 2014

@pkseet: All DigitalOcean servers run hardware RAID with different RAID levels, but in each case it requires multiple drive failures at the same time for any data loss to occur.

Show replies Reply

pkseet • January 17, 2014

Aren't the disks on DO configured with RAID 1 or 5?

Reply

Kamal Nasser • November 13, 2013

@Samuel: While we do not support RAID (since we offer virtual private servers), all of the hypervisors are on RAID setups.

Reply

theguycalledsam • November 12, 2013

I'd actually pay more if you offered it! *hint hint*

Reply

theguycalledsam • November 12, 2013

You mention RAID:

1. It's impossible on DigitalOcean(sadly)
2. Why don't you support it? D:

Reply

Popular Topics

Ubuntu

Linux Basics

JavaScript

Python

MySQL

Docker

Kubernetes

All tutorials →

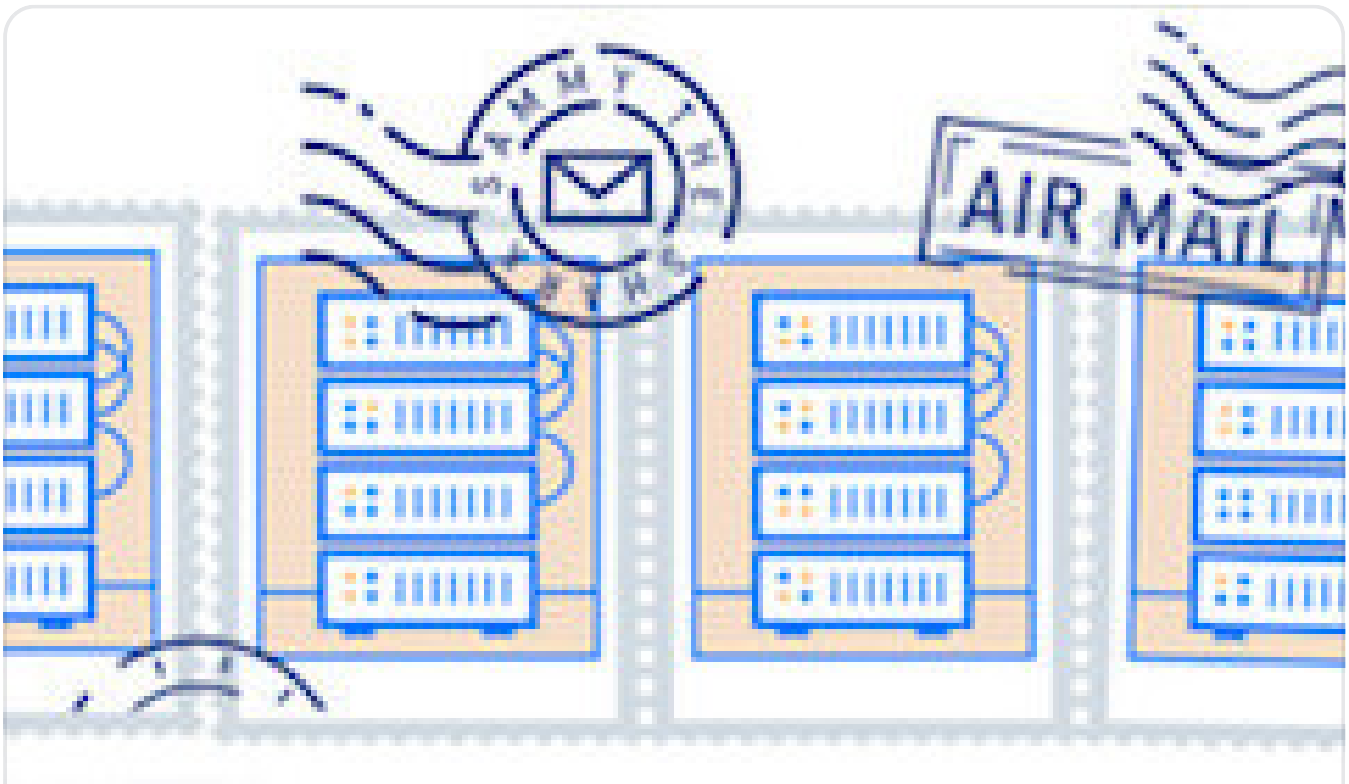
Talk to an expert →

Congratulations on unlocking the whale ambience easter egg! Click the whale button in the bottom left of your screen to toggle some ambient whale noises while you read.

Thank you to the Glacier Bay National Park & Preserve and Merrick079 for the sounds behind this easter egg.

Interested in whales, protecting them, and their connection to helping prevent climate change? We recommend checking out the Whale and Dolphin Conservation.

Reset easter egg to be discovered again / Permanently dismiss and hide easter egg



Get our biweekly newsletter

Sign up for Infrastructure as a Newsletter.

Become a contributor

Get paid to write technical tutorials and select a tech-focused charity to receive a matching donation.

Learn more →

Featured on Community

Kubernetes Course Learn Python 3 Machine Learning in Python

Getting started with Go Intro to Kubernetes

DigitalOcean Products

Cloudways Virtual Machines Managed Databases Managed Kubernetes

Block Storage Object Storage Marketplace VPC Load Balancers