Planning for ODM disaster recovery on OpenShift

By Pierre-Andre Paumelle

https://community.ibm.com/community/user/automation/blogs/pierre-andre-paumelle1/2022/07/13/planning-for-odm-disaster-recovery-on-openshift

Introduction

Operational Decision Manager (ODM) empowers business users and developers to collaborate when they automate an organization's business policies. ODM automates the decision-making process and governs future policy updates. Execution of the business rules (decision services at run time) scales out in clusters of servers running on bare metal, virtual machines, or containers.
ODM standard is a decision-making platform composed of 2 main servers, both of which use a database to store application data and state:

- Decision Server Console enables the deployment of new versions of a decision service and notifies all the servers to pick up the latest version. (Production environment)
- Decision Center is a web application to author and manage the business rules. (Authoring Environment)

Note, if you need to restart Decision Server Rules or Decision Center, you can start them from the databases without losing in-flight data.
As these applications are expected to be highly available as they are running on OpenShift (business is now globally less tolerant of downtime), a disaster can have a devastating effect if they are down for any time. A disaster can be anything that puts an organization's operations at risk, from a cyberattack, to equipment failures, to natural disasters.
OpenShift is in charge of High Availability of ODM and your applications.
Preparing for a disaster and recovering from such an event, in a reasonable amount of time, needs you to evaluate the resistance of your company's hardware and software, networking equipment, power, and connectivity.
The process of planning for disaster recovery (DR) must include some testing and may involve a separate physical site for restoring operations.

Why plan for disaster recovery

DR planning aims to protect an organization from the effects of negative events. The goal of DR is for a business to continue operating as closely to normal as possible following a disaster, and to quickly resume mission-critical functions. It's really important to hold an uninfected, or "last known good" copy of your systems, completely separated from your live systems. Backups can help to protect your organization against disasters. However, having backups is not always good enough in today's cybercrime environment; you need to test the recovery procedures and they need to be as efficient as possible.
Recovery is a major concern because of recent datacenter issues, like the fire in a datacenter of OVH cloud.
Recovery point objective (RPO) and recovery time objective (RTO) are two influential measurements in DR and downtime. While implementing a thorough DR plan isn't a small task, the potential benefits are significant.

Terms fundamental to disaster recovery

- **Redundancy**: The provision of additional or duplicate systems or equipment if an operating part or system fails.

- **High Availability (HA)**: Ensures the system can continue to process work within one location after routine single component failures.

- **Continuous Operations**: Ensure the system is never unavailable during planned activities. For example, if the application is upgraded to a new version, it is done in a way that avoids downtime.

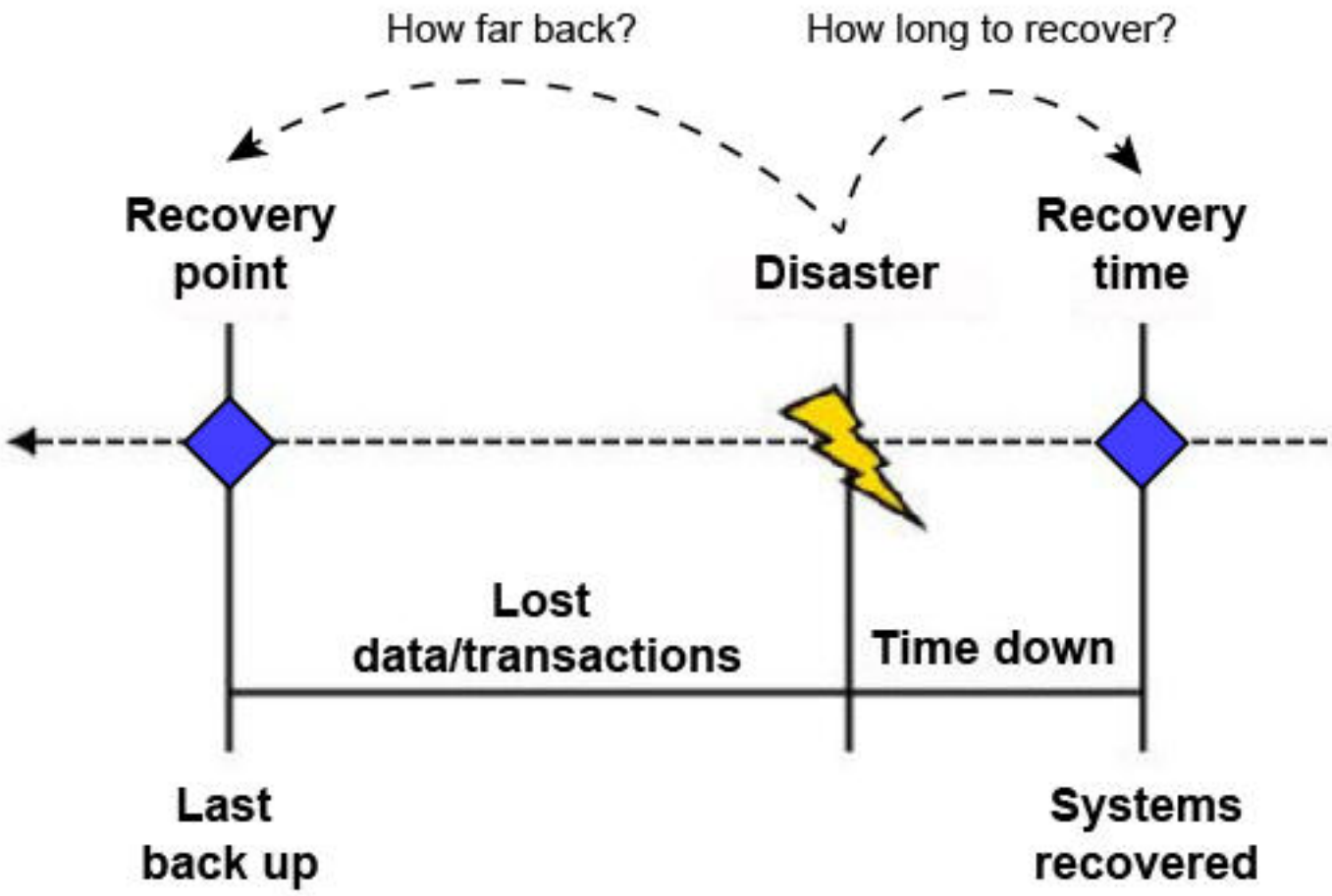
- **Continuous Availability (CA)** : High availability coupled with continuous operations. No tolerance for planned downtime and little unplanned downtime as possible. Note that while achieving CA almost always requires an aggressive DR plan, they are not the same thing.

- **Disaster Recovery (DR)**: Ensures the system can be brought back up at another location and can process work after an unexpected catastrophic failure at one location. Often multiple single failures are considered catastrophic. There may or may not be significant downtime as part of a disaster recovery. This environment may be substantially smaller than the entire production environment, as only a subset of production applications demands DR.

- **Recovery Point Objective (RPO):** Data recovery and acceptable data loss. RPO is the maximum age of files that an organization must recover from backup storage for normal operations to resume after a disaster. The recovery point objective determines the minimum frequency of backups. For example, if an organization has an RPO of four hours, the system must back up at least every four hours.

- **Recovery Time Objective (RTO)**: Service recovery with little to no interruption. RTO is the maximum amount of time, following a disaster, for an organization to recover files from backup storage and resume normal operations. In other words, the recovery time objective is the maximum amount of downtime an organization can handle. If an organization has an RTO of two hours, it cannot be down for longer than that.

The following diagram shows the relationship between the two measurements:



Service Levels (SLAs): Includes a clear set of conditions that define the availability requirements of the system. The requirements take into account:

- Components of the system: A system has many pieces and business aspects. How do the requirements differ?
- Responsiveness and throughput requirements: 100% of requests aren't going to work perfectly 100% of the time.
- Degraded service requirements: Does everything have to meet the responsiveness requirements ALL the time?
- Dependent system requirements: What are the implications if a system on which you depend is down?
- Data loss
    - Application data
    - Application state (is this critical in a disaster?)
- Maintenance. Change occurs, how does that affect availability?
- The unimaginable happens, then what?

Clusters and data centers

Multiple clusters or data centers can eliminate a single point of failure (SPOF). A SPOF is a part of a system that, if it fails, stops the entire system from working; a SPOF can be a cluster or an entire data center. In both cases, having multiple points increases reliability and insurance against disaster.
However, a word of warning! It is possible to construct a network so that latency is NOT an issue under normal conditions. But WANs are less reliable than LANs; and much harder to fix! It is also important to understand that network interdependency between data centers means that the data centers are not independent. Outages in one data center can impact clusters in another data center if the clusters aren't aligned to the data center boundaries.
In a typical DR scenario, an organization can recover and restore its technology infrastructure and operations when its primary data center is unavailable. DR sites can be internal or external. Companies with large information requirements and aggressive RTOs are more likely to use an internal DR site, which is typically a second data center.
External sites can be hot, warm, or cold. An outside provider can own and operate an external DR site.

- Hot site: A fully functional data center with hardware and software, personnel, and customer data, typically staffed around the clock.
- Warm site: An equipped data center that does not have customer data. An organization can install additional equipment and introduce customer data following a disaster.
- Cold site: Has infrastructure to support IT systems and data, but no technology until an organization activates DR plans and installs equipment.

Data center option 1: Full "Active/Active"

Two active data centers is the hardest and most costly option to achieve. Active/Active has the following characteristics:

- Both centers serve requests for the same applications.
- Requires shared application data.
    - Application data consistency is a prerequisite to any other planning.
    - Simultaneous reads/writes requires geographic synchronous disk replication. For example, IBM High Availability Geographic Cluster (HAGEO) or Sun Cluster Geographic Edition.
- Expectation of continuous availability and transparent fail-over.
    - Requires sharing application state.
    - Expectation is seldom realized.
    - Outage of one data center stops disk writes in both, and this is no longer "transparent".

Note, the disk replication you employ for application data and state, software updates, and application maintenance must be maintained independently to ensure isolation (and availability).

Data center option 2: “Active/Passive”

Two data centers, where one serves requests and the other is idle. Active/Passive has the following characteristics:

- Easier than Active/Active.
- User and application state synchronization is less critical.
- Asynchronous replication is likely sufficient.
- Lower cost for network and hardware capacity.

From a capacity perspective one data center is being under-utilized. Typically, the idle data center does not incur S/W license charges. If you don’t pay for S/W licenses is the cost and underutilization a concern?
WebSphere and ODM licenses are available for hot, warm, and cold external sites:

- Hot: Processing requests means a license is required. DB2 and MQ require hot licenses for replication.
- Warm: Started but not processing requests means a license is not required.
- Cold: Installed but not started means a license is not required.

Data center option 3: Hybrid “Active/Active” (Partitioned by applications)

Two data centers that run Decision Server (Production environment), where both serve requests, but run different applications or new application tests. For example, a specialized data center to run decision services for one country, and another specialized data center to run different decision services for another country. Hybrid Active/Active has the following characteristics:

- No shared application state, no shared application data.
- If you run the same application in both sites in the event of a disaster, there will be a loss of service.
    - Users failover from one data center to the other.
    - Asynchronous replication sufficient.
- Be aware of the implications of losing 50% of your system capacity should one data center go out of service.

Provides most of the benefits of full “Active/Active” without the cost and complexity.

Prioritization using tiers and business objectives

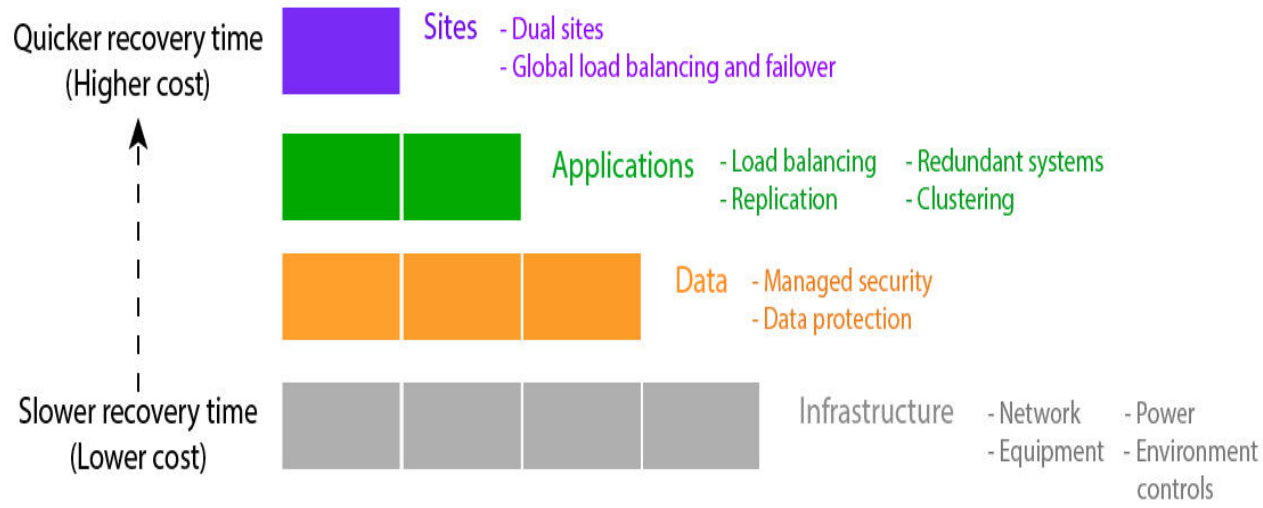
In order to consider the data center options:

1. Group your business needs and associated applications into tiers. Tier 0 represents the least amount of off-site recover ability and tier 6 represents the most. Tier 6 has minimal to zero data loss and recovery is instantaneous (no downtime or perhaps just a few minutes of down time). While the ability to recover improves with the next highest tier, costs also increase.
2. Place each application into a tier based on the hard/soft dollar impact on the organization.
3. Categorize by RPO and RTO. Tier 5 may have an RTO of 24 hours, then 48 to 72 hours, then perhaps 72 to 96 hours, and so on.

The recovery time objective (RTO) provides a guideline on how quickly the system must be able to accept traffic after the disaster. It is stating the obvious but shorter times require progressively more expensive techniques. For example, a tape backup and restore is relatively inexpensive, whereas a fully redundant fully operational data center is very expensive.
The recovery point objective (RPO) determines how much data you are willing to lose when there is a disaster. Again, it might be stating the obvious but the lower the data loss you can afford the higher the cost. For example:

- Restoring from tape is relatively inexpensive but you'll lose everything since the last backup.
- Asynchronous replication of data and system state requires significant network bandwidth to prevent falling far behind.
- Synchronous replication to the backup data center guarantees no data loss but requires very fast and reliable network and can significantly reduce performance.

Most RTO and RPO goals deeply impact application and infrastructure architecture and can't be done "after something has happened", when it is too late to change anything. For example, if you share data across data centers, your database and application design should avoid conflicting database updates and/or tolerate them. If application upgrades have to account for multiple versions of the application running at the same time this can impact user interface design, database layout, and so on.
Build a solid infrastructure to mitigate risk, and then add technologies to protect your data and scale your applications. The following diagram shows the infrastructure foundation and the building blocks you can use to reduce your recovery time.



Extreme RTO and RPO goals tend to conflict with one another. For example, using synchronous disk replication of data gives you a zero RPO, but that means the second system can't be operational, which raises RTO.
Trying to achieve a zero RTO AND a zero RPO is mutually exclusive!

Testing

An essential part of any plan is knowing who approves a recovery. An automated site fail-over is a bad idea, because triggering DR is very expensive and doing it if the situation does not warrant it just makes matters worse. However, one big challenge is detecting when a disaster has happened. It takes time to determine you are in a disaster state and trigger disaster procedures. While you are deciding if the system is down, you are probably missing your SLA.
An organization should have a schedule for testing its disaster recovery policy and be aware of how intrusive it is. Use the results from the tests to update the DR plan.

Recommended starting point for ODM on OCP

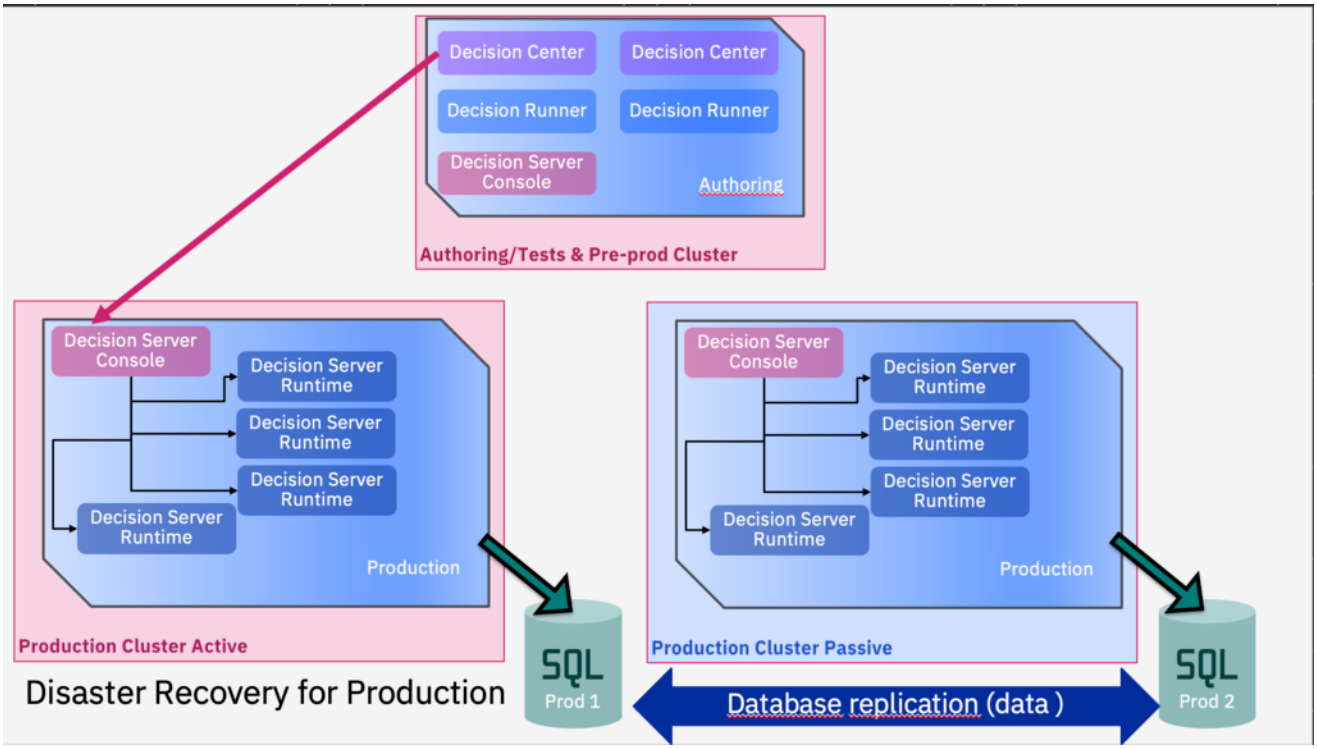
To begin with, if you are unsure of where to start, base your DR scenario on the Active/Passive option. The Active/Active option is harder to achieve. The replication of the database and LDAP can be an asynchronous replication or a backup/restore process depending on your DR objectives and service level.
In the following figures the LDAP replication is implicit and not visible.
The definition of the ODM environments is detailed in the CP4BA ODM topologies on OpenShift article

Production environment Disaster Recovery solutions

The production environment contains the runtime of ODM only.

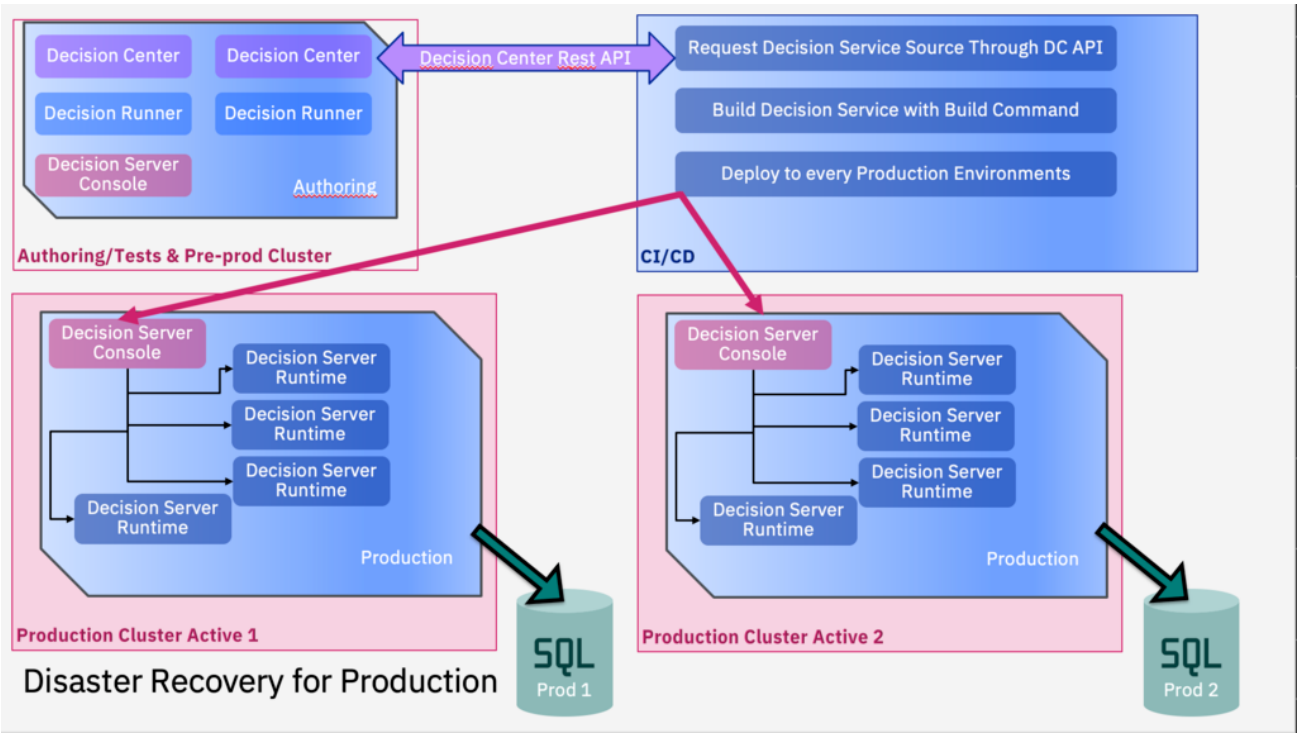
Active/Passive solution

In this Active/Passive solution, the Disaster recovery is based on database replication.
In this case the Passive cluster needs some specific ODM operation to become active.
You must **restart the Decision Server Console and Decision Server Runtimes** pods to ensure that the Decision Server is up to date when you will execute the rulesets.
Otherwise, the **ruleset executed could be outdated**.
The deployment of the ruleset from authoring to production is done inside Decision Center.



Active/Active solution

In this architecture the deployment is managed by a CI/CD (Continuous integration and continuous delivery) to ensure that all production environments are up to date. This CI/CD uses Decision Center REST API to get the ruleset artifacts. It builds and deploys the rulesets everywhere. If an environment is not available, the deployment should be done as soon as the environment becomes available.
Every production environment has a separate database and should be on separate datacenter and geographies.
All configurations are always ready.
This solution could be used for an Active/Active or Active/Passive architecture for production.
This architecture minimizes the clusters dependencies as we are removing the database replication.
The number of datacenter is not limited by a complex replication technology.
The network latency between datacenter has no impact on this architecture.

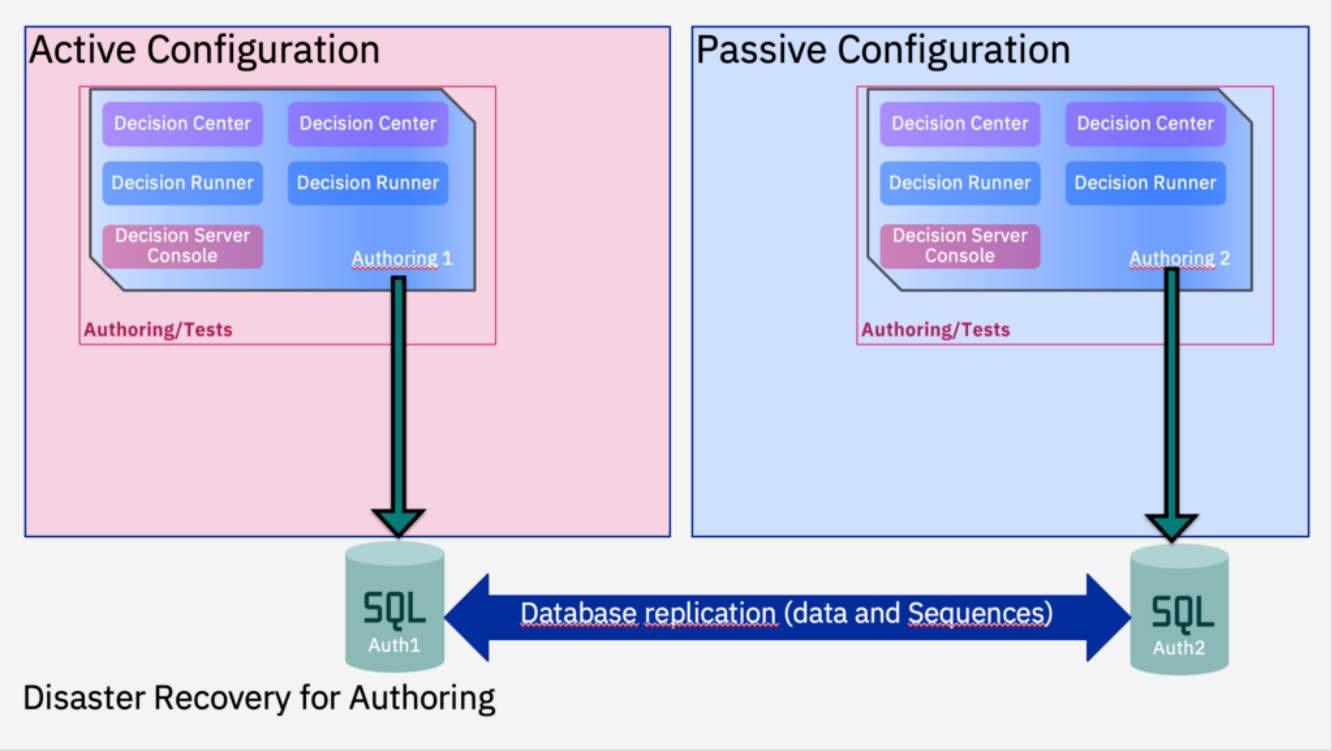


Authoring environments Disaster Recovery solution

For Authoring environment only Active/Passive solution is supported

Active/Passive solution

We build several Authoring environments with replicated database on several geographies.
The Disaster Recovery architecture is Active/Passive.
The replication is insured by the database which is the source of truth in Decision Center



Warning: This architecture is based on the database **replication** mechanism of **data and sequences.**

Conclusion

For Authoring, only Active/Passive architecture is supported and depends on database replication.
For Production, Active/Active and Active/Passive architectures are supported.
The Active/Active architecture implies the usage of CI/CD to ensure the deployments of the ruleset on every environment.