



SEARCHSTAX

CLOUD DEPLOYMENT ARCHITECTURES

for Apache Solr



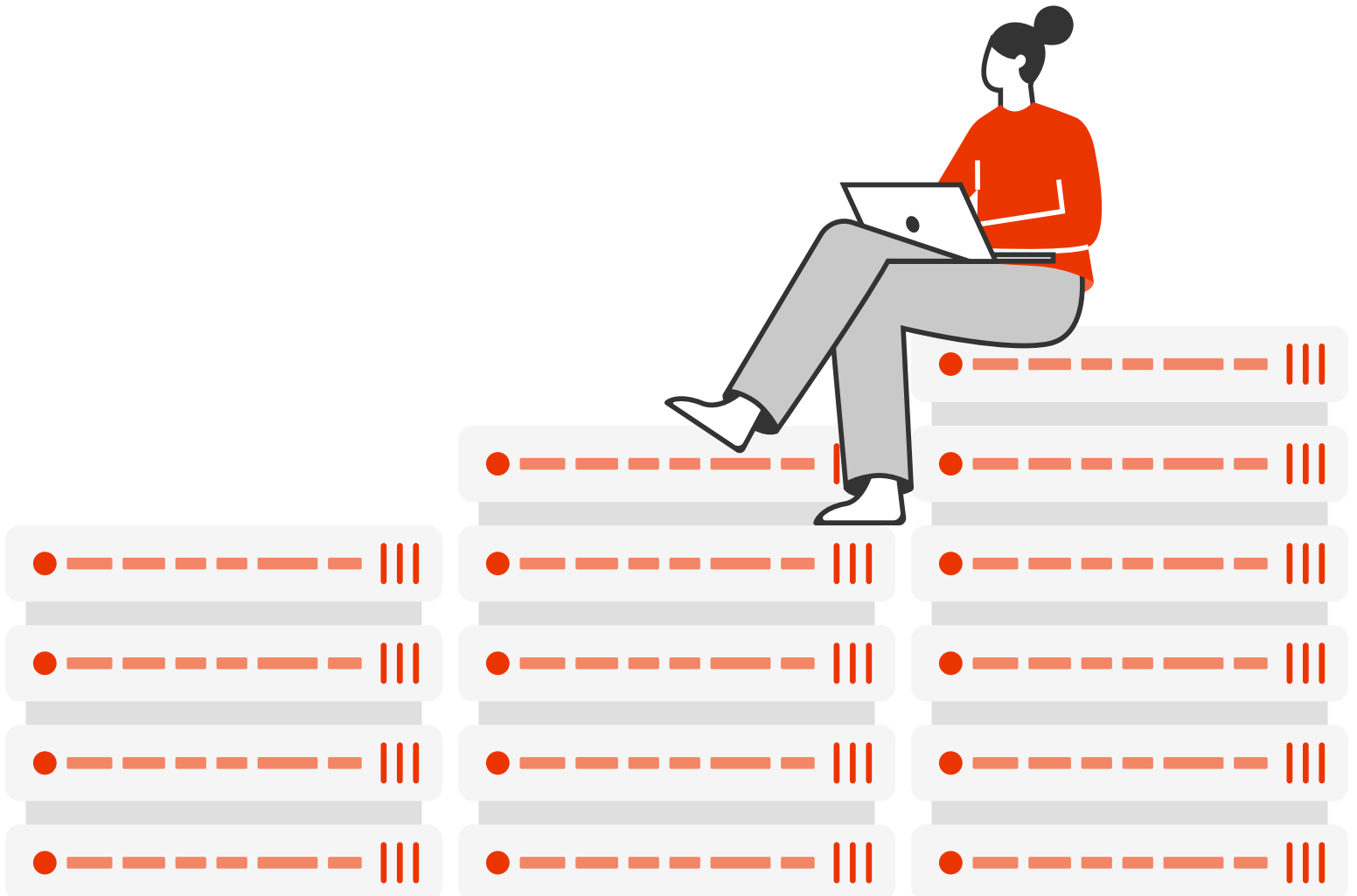
TABLE OF CONTENTS

- 1 **Introduction**
- 2 **Local Solr**
- 3 **Simplest Cloud Architecture: One Solr Plus One Zookeeper**
- 4 **SearchStax Dedicated Cluster: Two Solr with Three Zookeepers**
- 5 **Disaster Recovery with Duplicate Clusters**
- 6 **Hot Disaster Recovery with Vanity DNS & Automatic Failover**
- 7 **Warm Disaster Recovery: Minimal Remote System**
- 8 **Cross Data Center Replication (CDCR)**
- 9 **Virtual Network Peering**
- 10 **Conclusion**

INTRODUCTION

At SearchStax, we have a unique perspective on Solr systems architectures. With over fifteen years of hands-on experience hosting hundreds of SolrCloud deployments, we are familiar with Solr deployments ranging from simple out-of-the box implementations to enterprise mission-critical applications where high-availability, performance, failover, disaster recovery and security are important business requirements.

This whitepaper shows how a simple proof-of-concept system evolves into an enterprise-class armor-studded foundation to support your company's Solr-based search applications. Along the way, we demonstrate how we build a Solr infrastructure that serious enterprise organizations can trust to run the most important parts of their business, including options for disaster recovery, vanity DNS and virtual network peering.

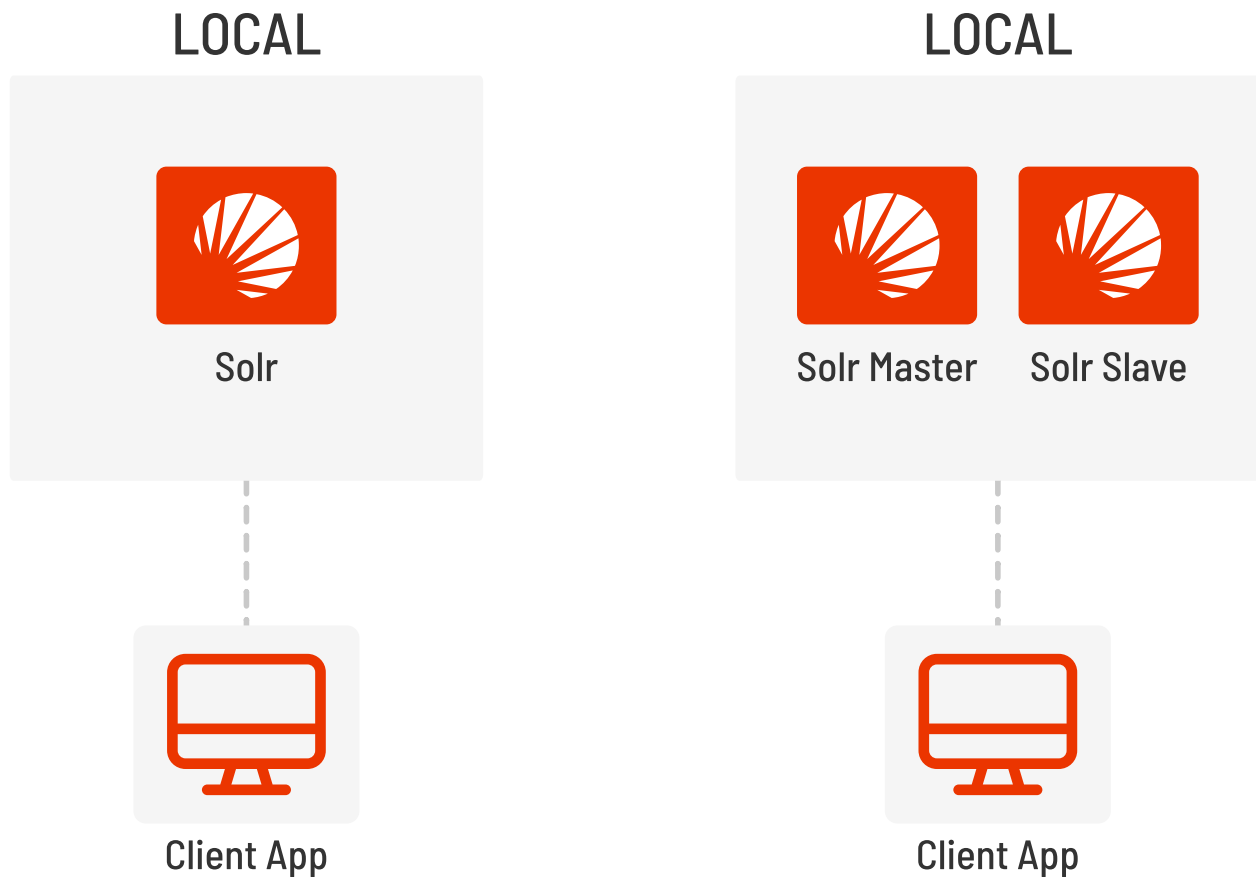


Most developers interested in building a Solr-based search experience will start by downloading Solr from the Apache Software Foundation and will install it on their local machine. Once Solr is running, you will connect it to a model of your search application with some representative data and see how it works.

In the following diagram, we illustrate a typical local project where the “Client Application” is a Sitecore project. As we manage many Solr deployments utilized by a variety of app platforms, this is a typical example but it is also applicable to other CMS tools like Adobe Experience Manager, Drupal, and custom applications.

If your project is headed for the Cloud, where nodes can dynamically expand or contract, then Solr Master/Slave architecture can be quite hard to maintain and wrestle with as it provides little support inside Solr to understand distributed architecture and dynamic discovery.

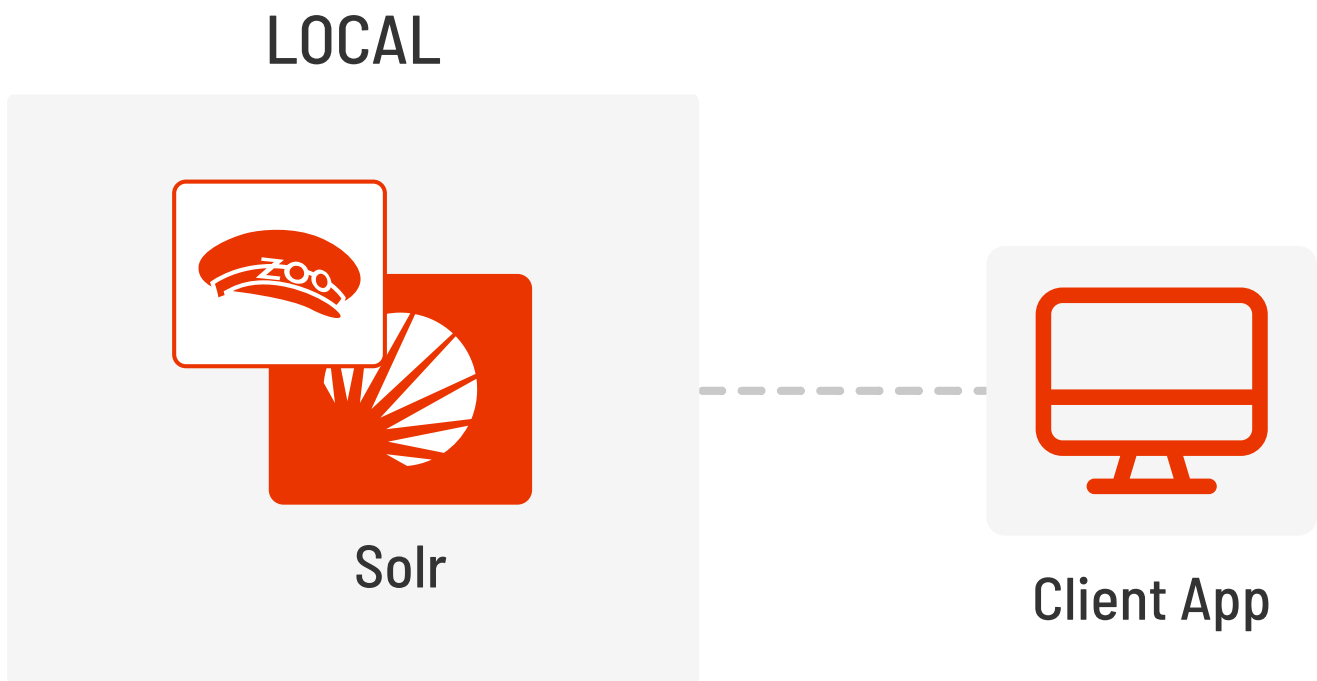
A local installation is a great first step to learn Solr because it gives you full control over all the configuration and index files at a time when you know the least about how to use them. One of the initial big configuration steps is to adapt your Solr schema to fit your legacy data, and that is much easier to learn on a local system.



SolrCloud is the Solr feature that lets you set up a cluster of Solr servers combining fault tolerance and high availability. SolrCloud is managed through Apache Zookeeper which is an open-source distributed coordination service that helps you manage deployments in the cloud. The simplest cloud architecture places Solr and Zookeeper on the same computer.

Zookeeper does two things for SolrCloud:

1. Zookeeper provides an interface for uploading configuration files to Solr, and acts as a central repository for all the configurations utilized by various Solr nodes. Since it's advised to not telnet or ssh into or communicate with the Solr servers to edit the Solr configuration files directly, you must edit the config files locally and then upload them (or utilize Solr's APIs). One Zookeeper instance is sufficient for that role.
2. Zookeeper is also utilized by Solr to understand the state of the cluster and is consulted in promoting one of the nodes in the cluster as a leader, which is an essential part of any Solr cluster deployments.



SEARCHSTAX DEDICATED CLUSTER: TWO SOLR WITH THREE ZOOKEEPERS

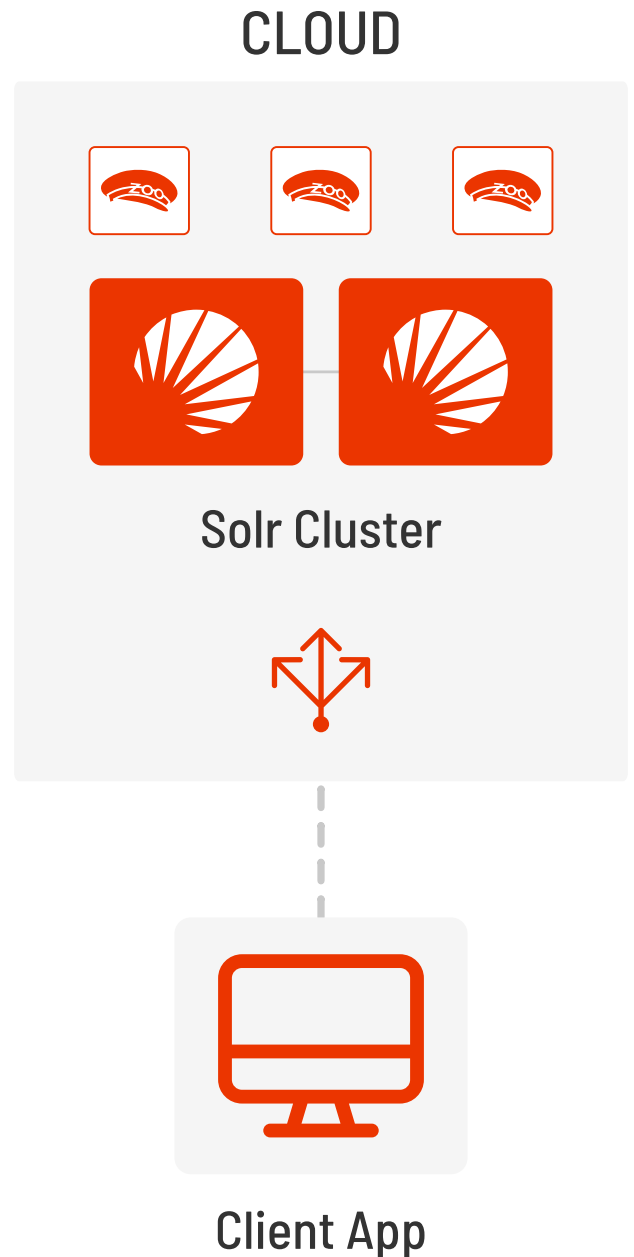
In 2019, SearchStax launched Dedicated Solr Cluster packages designed to be both reliable and economical.

This configuration loads Solr on two robust nodes and then sets up a Zookeeper ensemble on three tiny servers that are very inexpensive to operate. In the event of a Solr server crash, the Zookeeper ensemble can reset the leader replicas to the remaining Solr node and continue to serve both queries and index updates.

Often-asked questions about this architecture are:

- Are two Solr nodes really enough to provide high availability?
- What if the remaining node gets overloaded?
- What if the second node goes down, too?

That's where Disaster Recovery or DR is the answer to maintain high availability.



SolrCloud clusters can be easily extended by adding Solr nodes, which is the usual approach to high-availability. The more nodes serving queries in parallel, the less query latency.

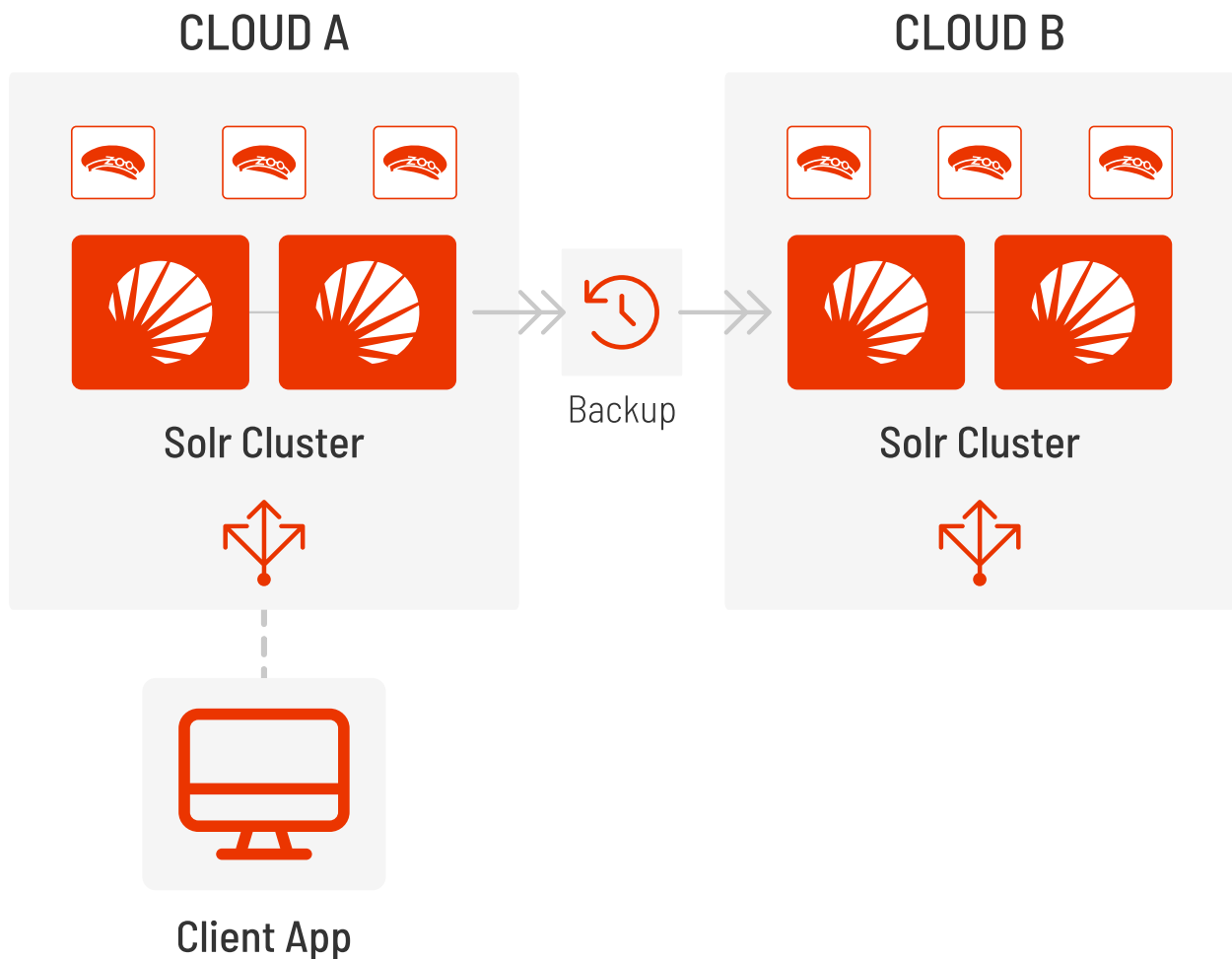
The secondary cluster is kept (almost) in sync with the primary by making periodic backups of the primary and restoring them on the secondary. Making a backup does not take the primary off line. The secondary cluster is briefly unavailable while the restore is in progress.

Backups can be scheduled on a daily basis (the usual practice) or more frequently as needed depending on your RTO and RPO policies.

The backup is stored in the same cloud region as the secondary cluster so you are still covered if an earthquake or hurricane takes out your primary Cloud region.

In this basic architecture, the Client App would have to be manually reconfigured before search traffic could resume. If you're using a Hot DR option, SearchStax can automate the reconfiguration step and significantly reduce the RPO.

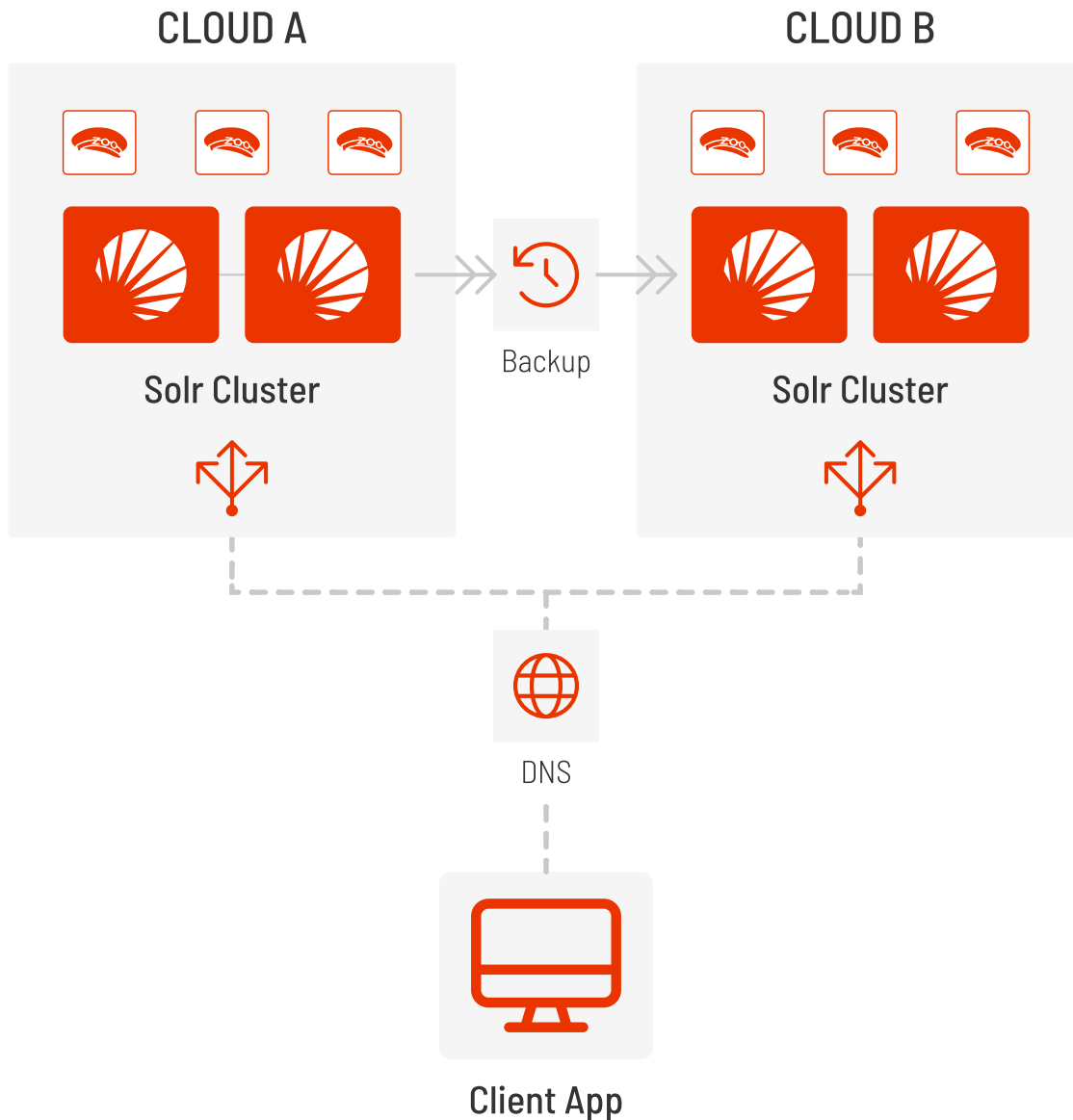
For failover, however, most clients prefer a complete duplicate cluster located in a different cloud region than the original or primary cluster. When something goes wrong with the primary, they redirect query traffic to the secondary (or "DR") cluster.



For premium clients with advanced service-level agreements, SearchStax can provide a Vanity DNS link to use as the Solr query endpoint to create a Hot Disaster Recovery with automatic failover option.

For instance, a problem in the primary cluster triggers a Heartbeat Alert and activates a Webhook that switches traffic to the secondary site. In practice, this alert will also send alarms and email messages to the account's owner and technical staff. In addition, SearchStax engineers are automatically notified and we always have at least one engineer on standby on a 24x7x365 basis.

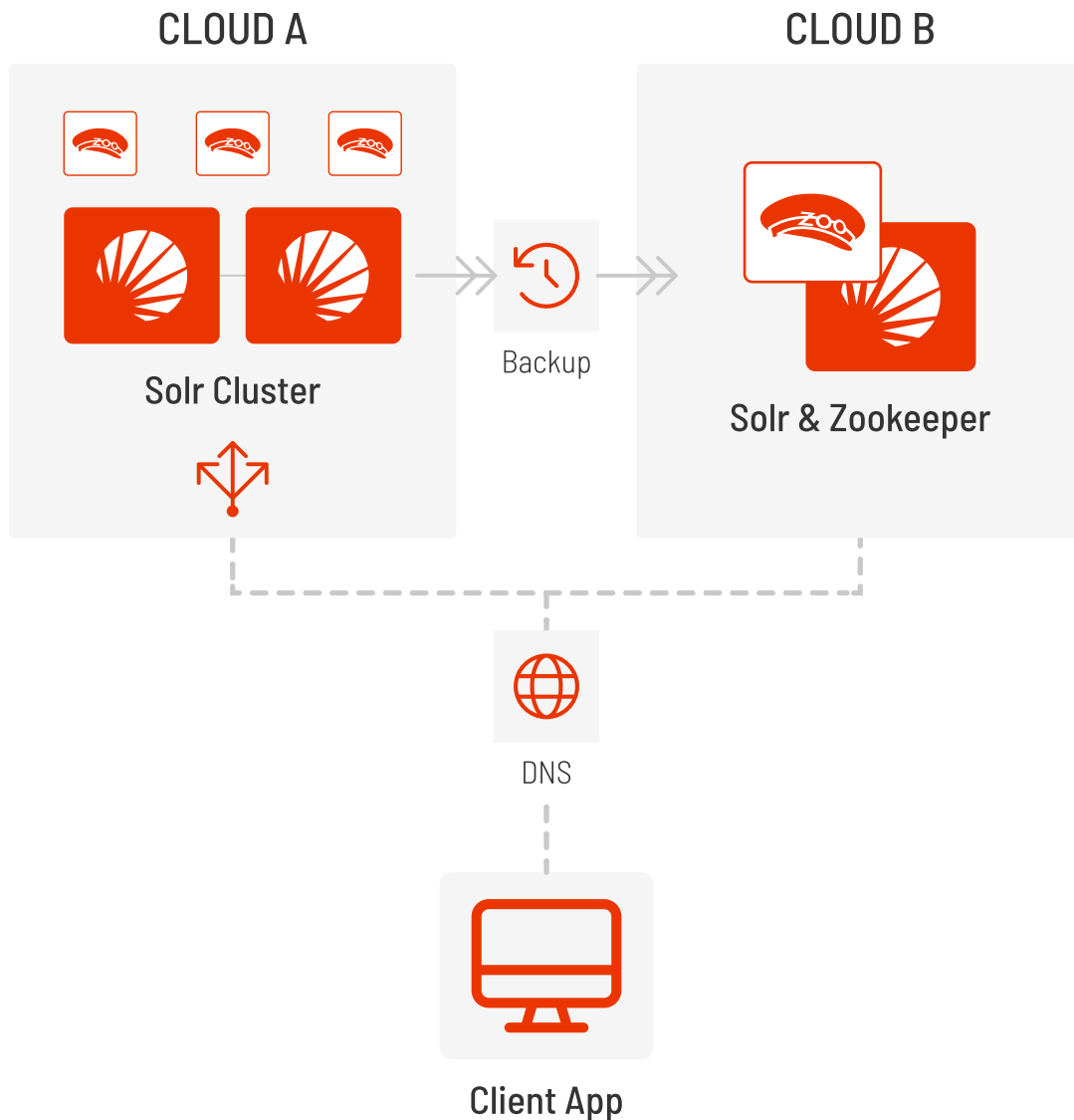
Rather than manually reconfigure the Client Application, we can redirect the Vanity DNS record. This takes only a few minutes to do manually, and can be automated using Alerts and Webhooks.



One issue with a Hot DR system is the expense of maintaining a complete secondary cluster. If your business requirements can tolerate slower query response times on a temporary basis, then a Warm DR solution with minimal downtime may be appealing to you.

In the event of a primary cluster failure, traffic automatically switches to the secondary system. Queries continue to be served, although maybe not as quickly as before. This option works for clients who can tolerate increased latency while the primary cluster is brought back online.

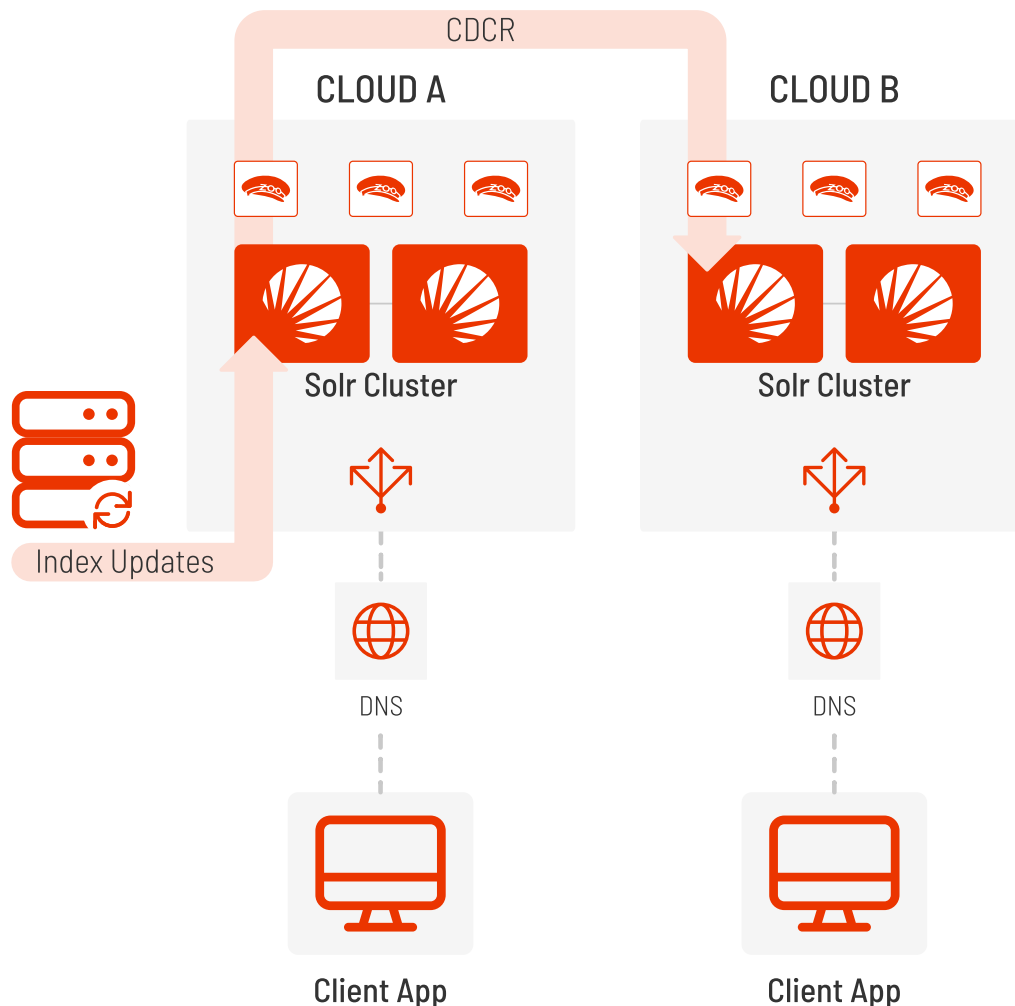
To this end, some clients equip the secondary site with a single Solr node and a co-resident Zookeeper. This is the same as the simplest SolrCloud deployment described above except that the server is sized to be robust enough to run a mature index.



A DR system that relies on a backup file has its limitations. It takes time to make a backup, and it takes time to restore it to the DR system. For clients who require almost-instantaneous synchronization across cloud regions, we turn to Cross Data Center Replication (CDCR). New documents propagate world-wide in seconds.

CDCR continuously synchronizes two (or more) Solr clusters in separate cloud regions in real time. Incoming documents are sent to a "Source" collection in Cloud A, where they are added to the index. Solr then pushes the same set of changes to a "Target" collection in Cloud B. In the meantime, both deployments are fully available for serving queries. In a typical system of this type, the "Target" indexes are only a few seconds behind the "Source" index, even though they may be on the far side of the world.

CDCR can take the place of the backup file in either Warm or Hot DR architectures mentioned above, achieving world-wide high-availability at the same time as near-instantaneous failover if one of the deployments (or cloud regions) should go offline. One feature of CDCR is that any of the deployments can serve as the Source (but only one at a time). If the Source deployment fails, one of the Target deployments can take over ingestion of new content. The system keeps right on running without interruption.



A final configuration is a Private Network which creates a highly-secure search environment for clients who must comply with high-security policies.

Depending on your cloud provider, virtual network peering is handled differently:

- **Amazon Web Services (AWS):** SearchStax can provide Solr instances either within your own Virtual Private Cloud (VPC) or in a separate VPC.
- **Microsoft Azure:** SearchStax can provide Solr instances in a separate Private Network.
- **Google Cloud:** SearchStax can provide Solr instances in a separate Virtual Private Cloud.

This illustration shows an Azure Private Network architecture with the VNet Peering Option deploying the Solr servers in a VNet.

This VNet communicates with your application VNet through VNet-peering which guarantees that all of the operational aspects of managing Apache Solr are handled by SearchStax engineers without directly accessing your VNet.

**Note that this architecture can be applied to any of the SolrCloud architectures described earlier in this white paper.*



CONCLUSION

Enterprise-level SolrCloud architecture may seem complex, but every component is there for a good reason. By following the evolution from the simplest Solr project to the most complex, every step is a simple and logical progression resulting in an iron-clad Solr architecture that is secure, performant and highly available.

SEARCHSTAX CLOUD

SearchStax Cloud is a fully-managed SaaS solution that automates, manages and scales Solr infrastructure in public or private clouds on AWS, Azure and Google Cloud.

Developers use SearchStax Cloud to:

- Implement faster with less risk
- Focus resources on building better search experiences
- Spend less time managing and supporting Solr

See how easy it is to deploy, manage and scale Solr in the cloud by taking advantage of our [Free 14-day trial](#).

To learn more about SearchStax Cloud, go to searchstax.com/managed-solr/

CONTACT US

[SearchStax Sales](#)

sales@searchstax.com

+1.844.973.2724