Monitoring

Objectives

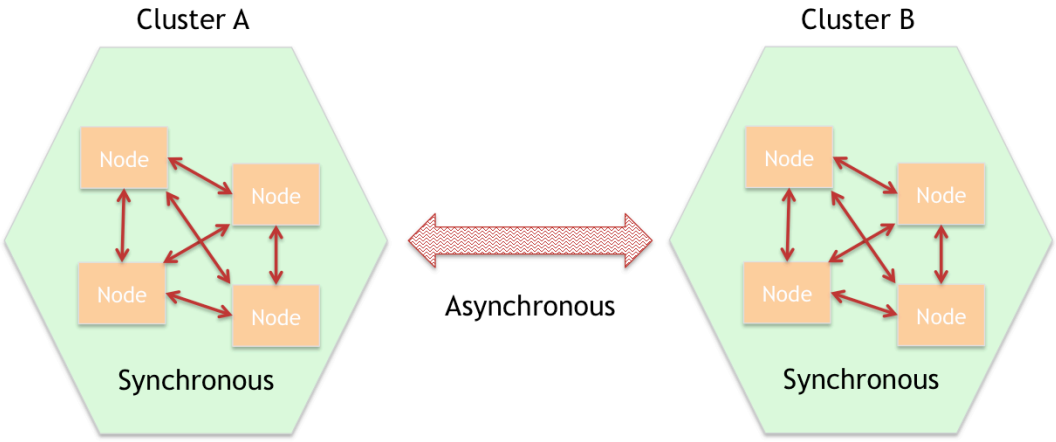
At the end of this module, you will be able to:

- Understand the basic XDR Architecture
- Configure XDR to join clusters

Cross Datacenter Replication (XDR)
Basic Architecture

XDR – High Level Architecture

XDR synchronizes data between different clusters. Note that while an Aerospike cluster typically replicates data synchronously within a cluster, XDR always works asynchronously.



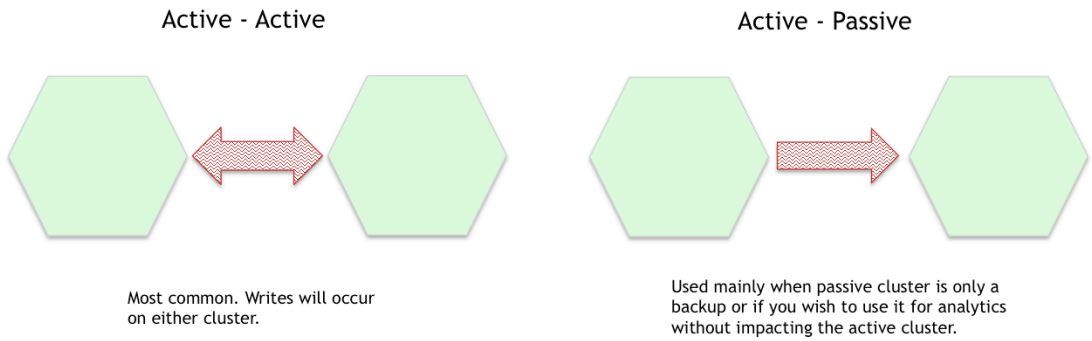
XDR allows for asynchronous replication of data between 2 clusters. Note that replication WITHIN a cluster is still synchronous.

The time lag between clusters will vary depending on settings and bandwidth, but the intent was to have replication on the order of a few seconds.

While normal deployments are for clusters to be in different data centers, they may also be within the same data center.

XDR – High Level Architecture

Administrators may choose from two different replication models:



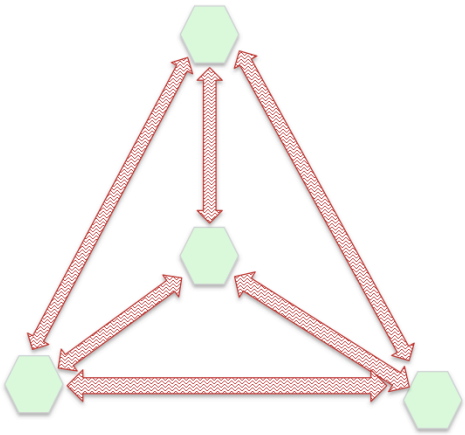
You may choose between active-active and active-passive modes.

If you are using active-passive for a hot backup, you may want to consider still using active-active. The main reason is that writes that occur on the backup cluster during an outage of the main cluster will then be queued to automatically sync back to the main cluster when normal operations resume.

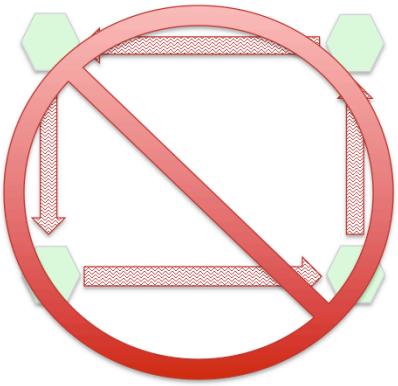
XDR – High Level Architecture

There is also a choice of replication geometries (combinations are possible). Generally, ring geometry is not recommended.

Star Geometry



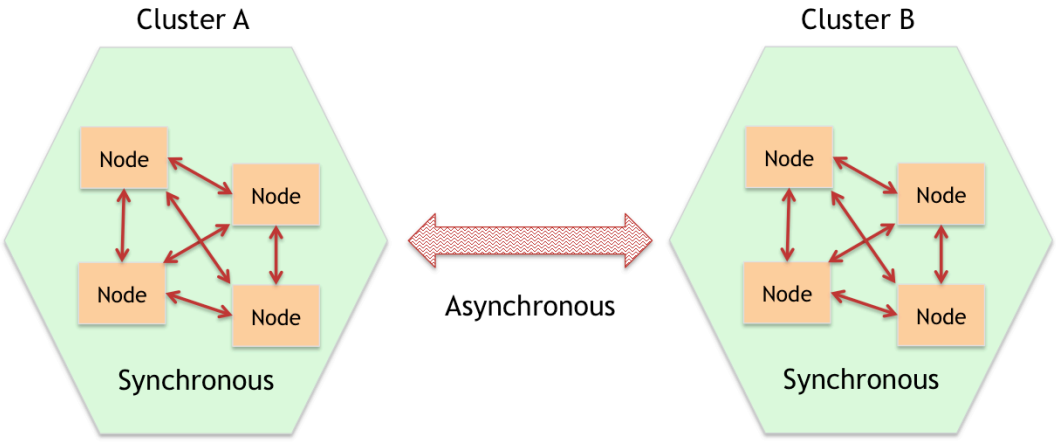
Ring Geometry



Ring geometries have the weakness that if a single cluster in the ring is out, synchronization is broken. Star geometries do not have this issue.

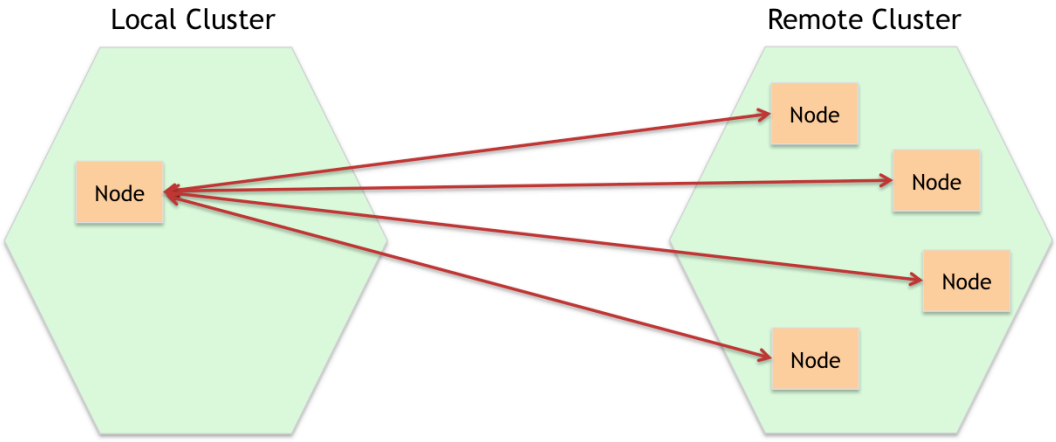
XDR – High Level Architecture

XDR synchronizes data between different clusters. Note that while an Aerospike cluster typically replicates data synchronously within a cluster, XDR always works asynchronously.



XDR – Lower Level Architecture

No matter which methods you choose, XDR works by putting an Aerospike client on each node in the local cluster that will transfer the data to the remote cluster. This architecture means you do not need to have the same cluster configuration in both nodes.

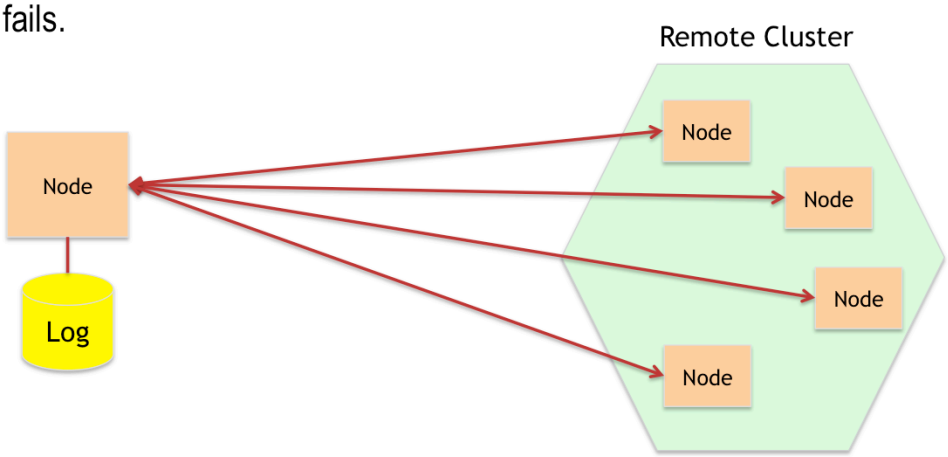


Every node in the local cluster will work to transfer data to the remote cluster. Each node ill act similarly to a standard Aerospike client. It must be given one or more seed nodes in the remote cluster, which it will use to learn about the other nodes in the cluster.

XDR – How It Really Works

At a low level, what XDR does on each node in the local cluster is to keep track of the hashes of all keys that have been written to in a log. XDR will then periodically read from the log and write the latest value for those hashes to the remote cluster.

The log keeps track of all master writes as well as replica writes in the event a node fails.



Cross Datacenter Replication (XDR) Configuring

XDR – Configuring Namespace

XDR must be enabled for each namespace that you wish to use.

- Basic XDR
- Networking
- Replication

XDR – Configuring Basic XDR

There are several parameters that are vital to the operation of XDR:

- Enable XDR
- Named pipe path
- Digest log
- XDR error log
- XDR pidfile
- Local node port
- XDR info port
- XDR stop writes

XDR – Enable XDR

Description	This setting will turn on XDR centrally
Context location	xdr
Config parameters (defaults)	enable-xdr
Notes	By dynamically changing this variable, you can turn XDR on or off without restarting the node.
Change dynamically	no
Best practices	If using XDR, set to "false".

This is the setting within the xdr context and is the master control for XDR on the node. Note that it is possible to turn on xdr on the namespaces, but for the main xdr setting to be off. In that event, no records will be sent to the remote cluster.

XDR – Named Pipe Path

Description	This parameters determines the XDR channel used for communication.
Context location	xdr
Config parameters (defaults)	xdr-namedpipe-path (/tmp/xdr_pipe)
Notes	The parameter is not generally altered. If using XDR, this parameter is required.
Change dynamically	yes
Best practices	Keep the default value.

XDR – Digest Log

Description	This is the location of the file that holds the hashes for all the keys that have been written.
Context location	xdr
Config parameters (defaults)	xdr-digestlog-path (/opt/aerospike/digestlog) [size]
Notes	If using XDR, this is required. You must have read/write access to the file and the size must be configured. For example, to set a 100 GB digest log file use: `xdr-digestlog-path /opt/aerospike/digestlog.dat 100G` Changing the filesize requires a restart of XDR.
Change dynamically	no
Best practices	The convention is to use: /opt/aerospike/digestlog Set the size of the digest log to 80 bytes/record for all the records you expect to have on the node. For example, if you have 100M objects, set the file size to at least 8 GB.

Note that the object count it total. XDR also keeps track of replica copies, so the storage of 80 bytes should be for all (master and replica) records on each node.

XDR – XDR Error Log

Description	This is where XDR keeps its error messages.
Context location	xdr
Config parameters (defaults)	xdr-errorlog-path (none)
Notes	When using XDR, this field is required. There is no default value. The default location is read/write for root or sudo only.
Change dynamically	no
Best practices	Use the path: /var/log/aerospike/asxdr.log

XDR – XDR Pidfile

Description	This is where the process ID (PID) of the XDR process is kept.
Context location	xdr
Config parameters (defaults)	xdr-pidfile
Notes	
Change dynamically	no
Best practices	Use the path: /varl/log/aerospike/asxdr.pid

XDR – Local Node Port

Description	This is the port of the local Aerospike node that will be used to get the data.
Context location	xdr
Config parameters (defaults)	local-node-port
Notes	When using XDR, this parameter is required. The standard port for this is 3000. Note that if the value has changed multiple times since the last shipment of the record, only the last value will be transmitted.
Change dynamically	no
Best practices	

You can increase the block size, but cannot decrease the block size.

XDR – XDR Info Port

Description	This is the port used by various tools to gather information on the health of XDR.
Context location	xdr
Config parameters (defaults)	xdr-info-port
Notes	When using XDR, this field is required
Change dynamically	no
Best practices	Port 3004 is standard

It is possible to telnet directly into the XDR port to issue commands directly. There are some simple instructions at:

http://www.aerospike.com/docs/operations/troubleshoot/xdr/

XDR – XDR Stop Writes

Description	This parameter controls whether or not the Aerospike will accept writes if XDR is down.
Context location	xdr
Config parameters (defaults)	stop-writes-noxdr (false)
Notes	
Change dynamically	yes
Best practices	Turning on this feature is dependent on business requirements. Most set this as false, meaning even if XDR is down, the database will continue to accept writes.

This feature is important if you want to make sure that local writes are always replicated to the remote cluster. This may come at the cost of availability, so be very careful with this setting.

Basic Configuration Example

```
...
xdr {
   enable-xdr          true
   xdr-namedpipe-path  /tmp/xdr_pipe
   xdr-digestlog-path  /opt/aerospike/digestlog 100G
   xdr-errorlog-path   /var/log/aerospike/asxdr.log
   xdr-pidfile         /var/run/aerospike/asxdr.pid
   local-node-port     3000
   xdr-info-port       3004
   stop-writes-noxdr   false
   ...
}
...
```

XDR – Network Configuration

There are several parameters that are vital to the network operation of XDR:

- Compression Threshold
- Datacenter

XDR – Compression Threshold

Description	Any record larger than the compression threshold will be compressed.
Context location	xdr
Config parameters (defaults)	xdr-compression-threshold
Notes	If you do not wish to compress the data, set this to 0.
Change dynamically	yes
Best practices	A value of 1000 is a good value.

Aerospike has found that making this setting too small can sometimes lead to increasing object size. 1000 bytes has been found to be a good compromise.

XDR - Datacenter

Description	For every remote datacenter you will need a separate datacenter subcontext.
Context location	xdr
Config parameters (defaults)	datacenter [datacenter_name]
Notes	There are special if the nodes in the remote datacenter have separate public and private IP addresses. Multiple datacenters can be specified with different names.
Change dynamically	no
Best practices	

You will need one of these definitions for every datacenter in your overall setup.

XDR Networking Example

```
...
xdr {
   ...
   xdr-compression-threshold 1000
   datacenter <datacenter-name> {
      dc-node-address-port <remote_node1> <remote_port>
      dc-node-address-port <remote_node2> <remote_port>
   ...
}
...
```

You need only specify a single remote port. XDR will learn about the others from the first one. Specifying the others is used as a backup in the event the first node is down.

XDR Networking Example (Public/Private IP)

```
...
xdr {
   ...
   xdr-compression-threshold 1000
   datacenter <datacenter-name> {
      dc-node-address-port <remote_node1> <remote_port>
      dc-node-address-port <remote_node2> <remote_port>
   ...
      dc-int-ext-ipmap <private_ip_1> <public_ip_1>
      dc-int-ext-ipmap <private_ip_2> <public_ip_2>
}
...
```

The parameter dc-int-ext-ipmap must be specified for ALL nodes in the remote cluster.

The public/private settings are only needed if you are communicating with a cluster that must have separate connections for public/private IP addresses.

XDR – Replication Configuration

There are several parameters that determine how XDR will replicate:

- ❑ Enable XDR
- ❑ Remote datacenter
- ❑ Set replication (optional)

XDR – Enable XDR

Description	This setting must be in the namespace stanza. It will enable XDR for just the namespace in question.
Context location	namespace
Config parameters (defaults)	enable-xdr
Notes	This setting is subject to the same enable-xdr setting in the XDR stanza. Unlike, the XDR context setting for enable-xdr, this value can be dynamically changed. If used without any set options, all data from the namespace will be replicated to the configured datacenters using the `xdr-remote-datacenter` parameter.
Change dynamically	yes
Best practices	

This setting does not override the setting in the xdr context.

XDR – Remote Datacenter

Description	Sets the remote datacenter to use for the particular namespace
Context location	namespace
Config parameters (defaults)	xdr-remote-datacenter
Notes	You may select more than one datacenter for replication. Simply add additional lines using the same xdr-remote-datacenter parameter.
Change dynamically	no
Best practices	This parameter is what is used to properly configure the replication geometry. Make sure you have properly mapped out how the replication will work.

XDR – Enable XDR For Sets

Description	This optional parameter controls whether or not replication will be filtered by sets.
Context location	xdr
Config parameters (defaults)	sets-enable-xdr (true)
Notes	You may choose to replicate via either blacklist or whitelist. Blacklist At the top level, set `sets-enable-xdr` to "true" and set "false" for each of the blacklisted sets with the parameter `set-enable-xdr`. Whitelist At the top level, set `sets-enable-xdr` to "false" and set "true" for each of the blacklisted sets with the parameter `set-enable-xdr`.
Change dynamically	no
Best practices	

Basic Configuration Example Replicate All

```
...
namespace <namespace_name> {
   enable-xdr          true
   xdr-remote-datacenter <dc-name-1>
   xdr-remote-datacenter <dc-name-2>
   ...
}
...
```

The remote datacenters must be defined in the `xdr` stanza

Basic Configuration Example Replicate Sets With Blacklist

```
...
namespace <namespace_name> {
   enable-xdr              true
   xdr-remote-datacenter <dc-name-1>
   xdr-remote-datacenter <dc-name-2>
   sets-enable-xdr       true
   set <setname-1> {
      set-enable-xdr     false # blacklisted
   }
   ...
}
...
```

Basic Configuration Example Replicate Sets With Whitelist

```
...
namespace <namespace_name> {
   enable-xdr                true
   xdr-remote-datacenter <dc-name-1>
   xdr-remote-datacenter <dc-name-2>
   sets-enable-xdr         false
   set <setname-1> {
      set-enable-xdr       true # whitelisted
   }
   ...
}
...
```

Cross Datacenter Replication (XDR) Troubleshooting

XDR – Configuring

XDR

Summary

At the end of this module, you will be able to:

- Understand the basic XDR Architecture
- Configure XDR to join clusters