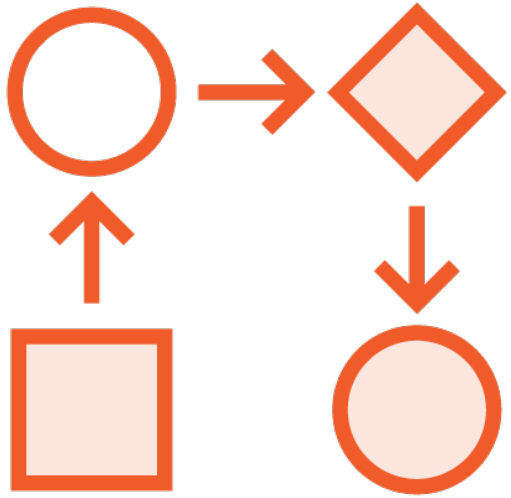
Advanced Networking and Distributed Availability Groups

Russ Thomas
DATA ARCHITECT

@sqljudo

Module Prerequisites

Comfort with prior course modules and concepts



Hands on experience working with availability groups

SQL Server 2019 General Availability

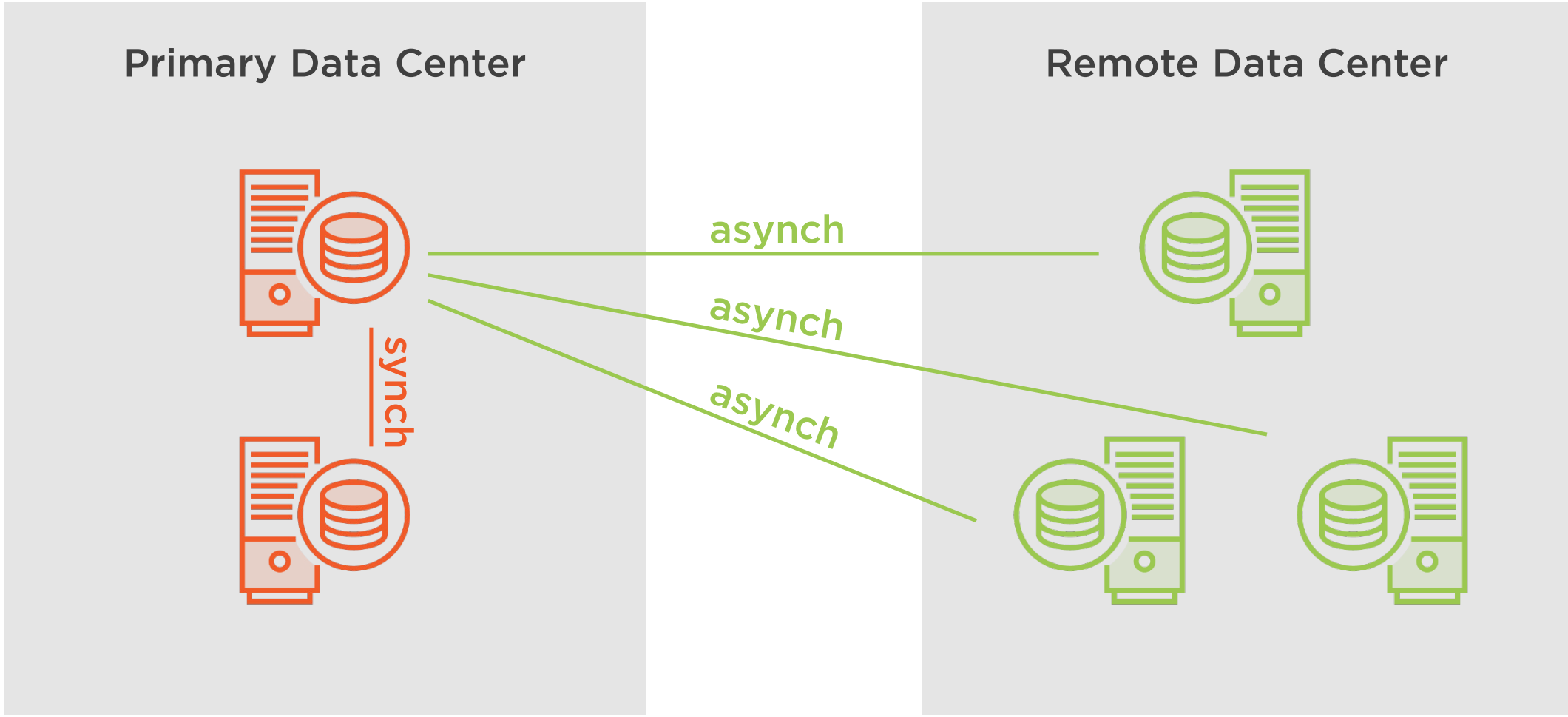
Module Topics

Scaling Availability Groups with Distributed Availability Groups

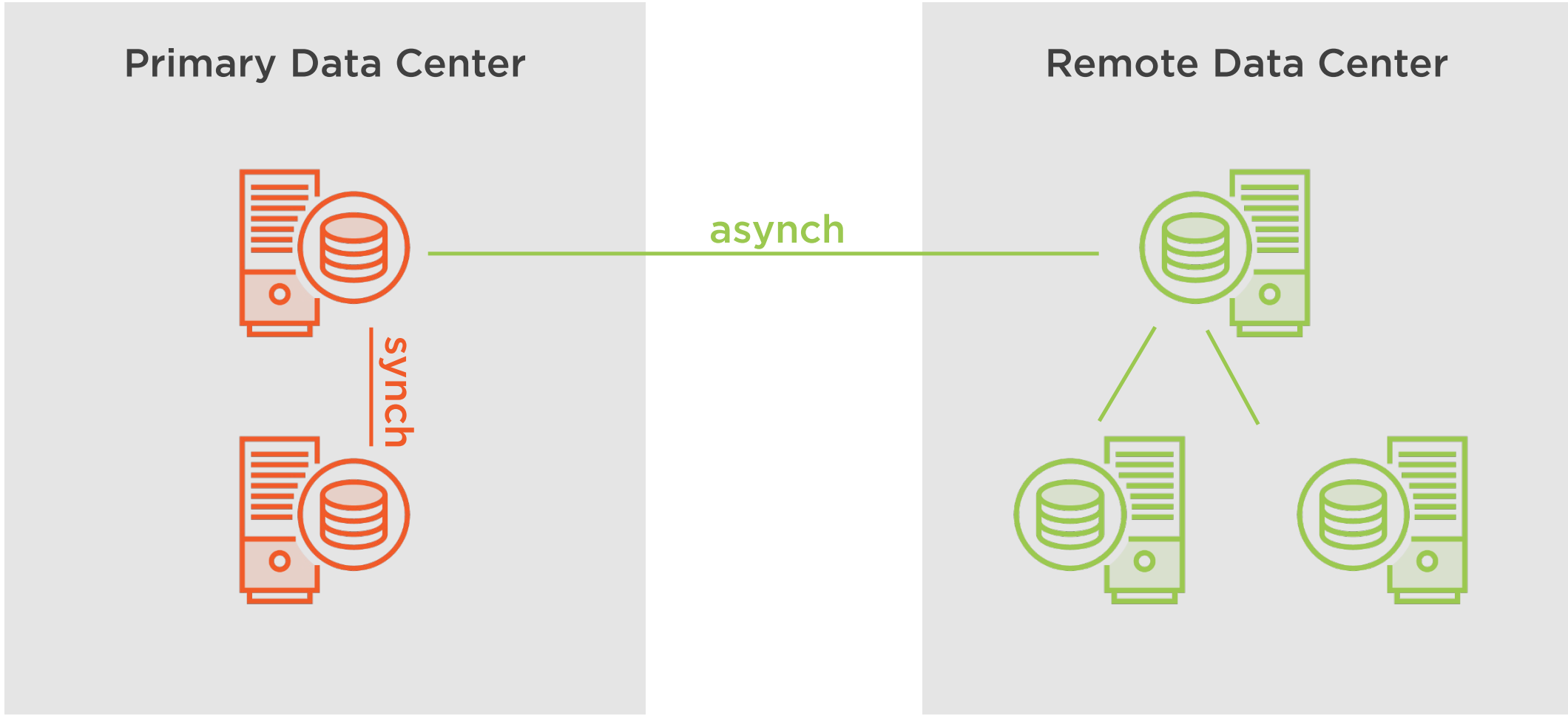
Advanced Failover Cluster Heartbeat and Health Monitoring

Administering Availability Groups with PowerShell

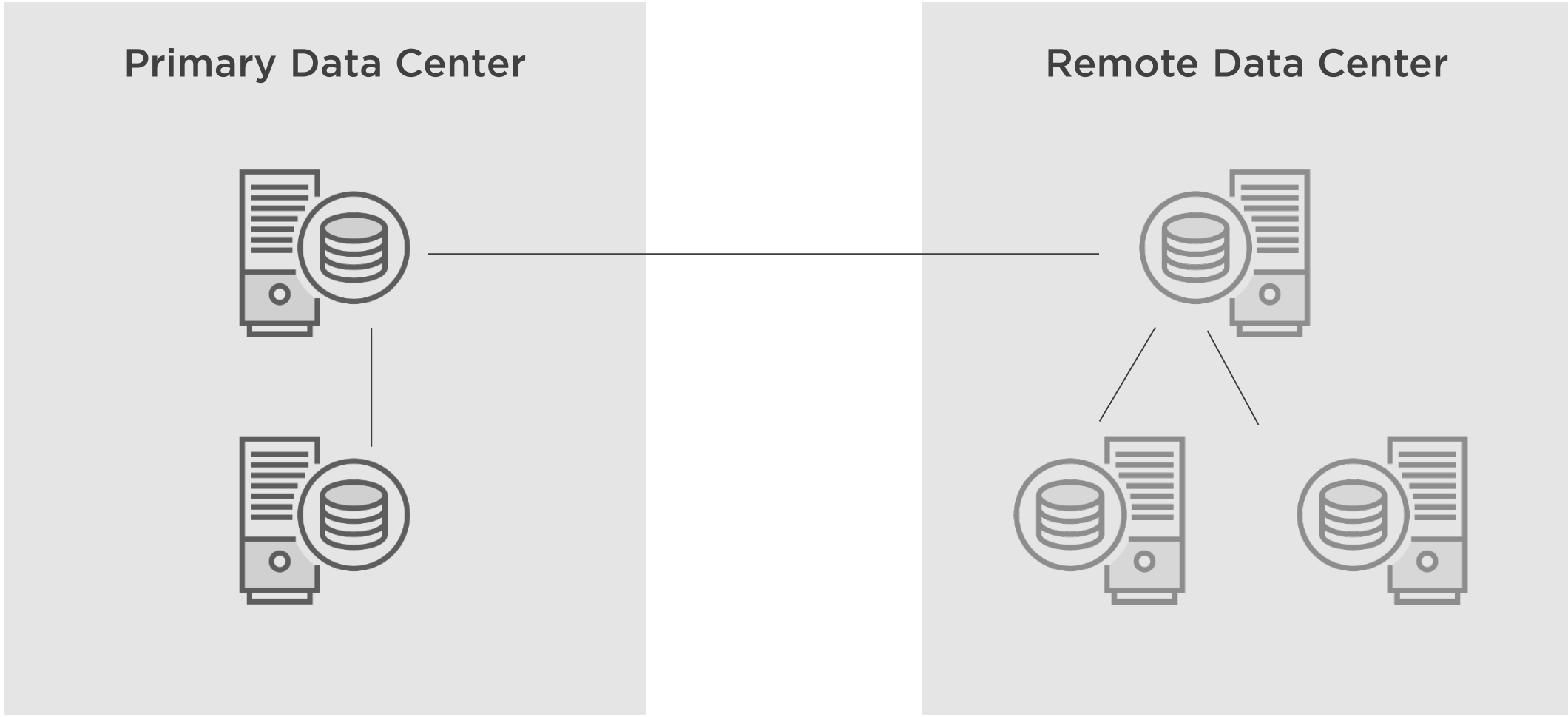
Sample Scenario



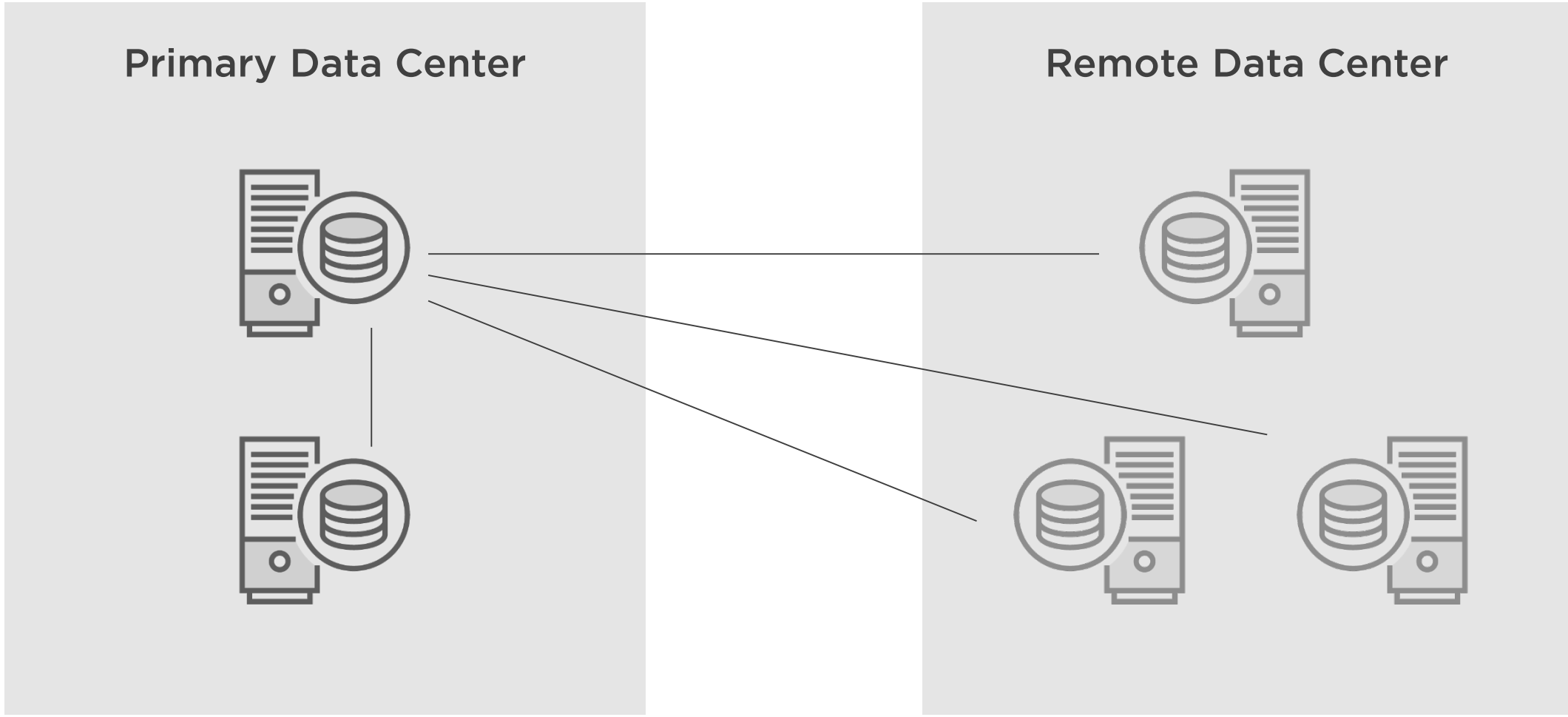
Sample Scenario



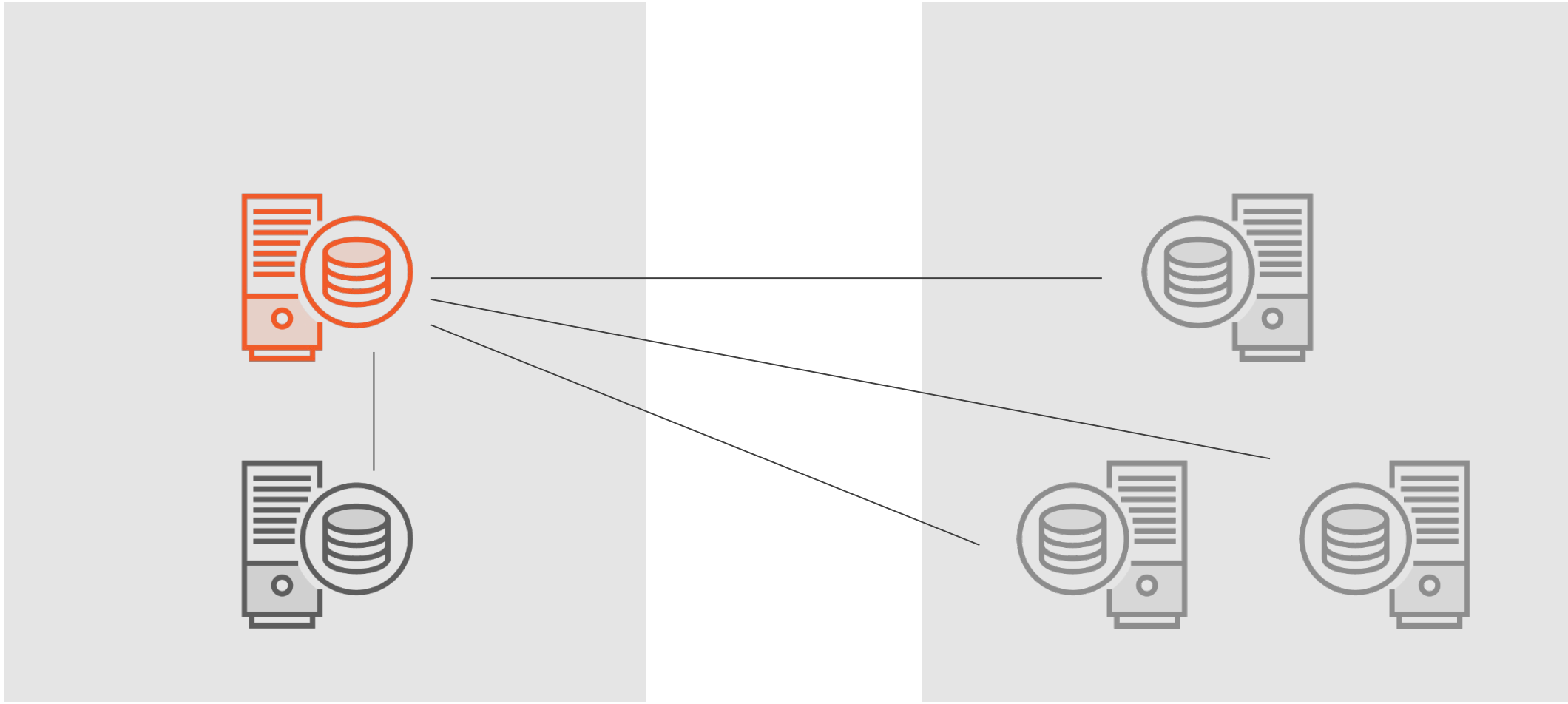
Distributed Availability Groups



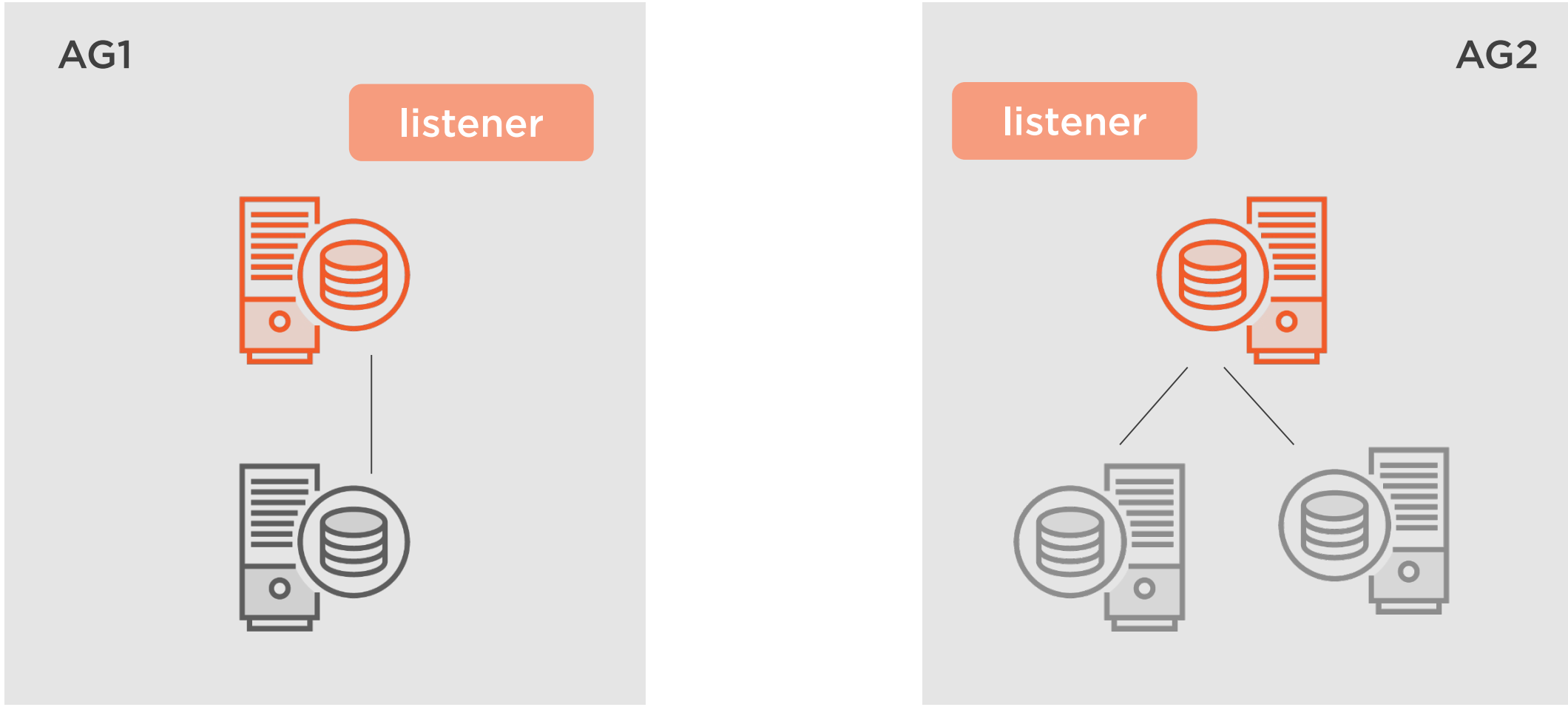
Single Availability Group



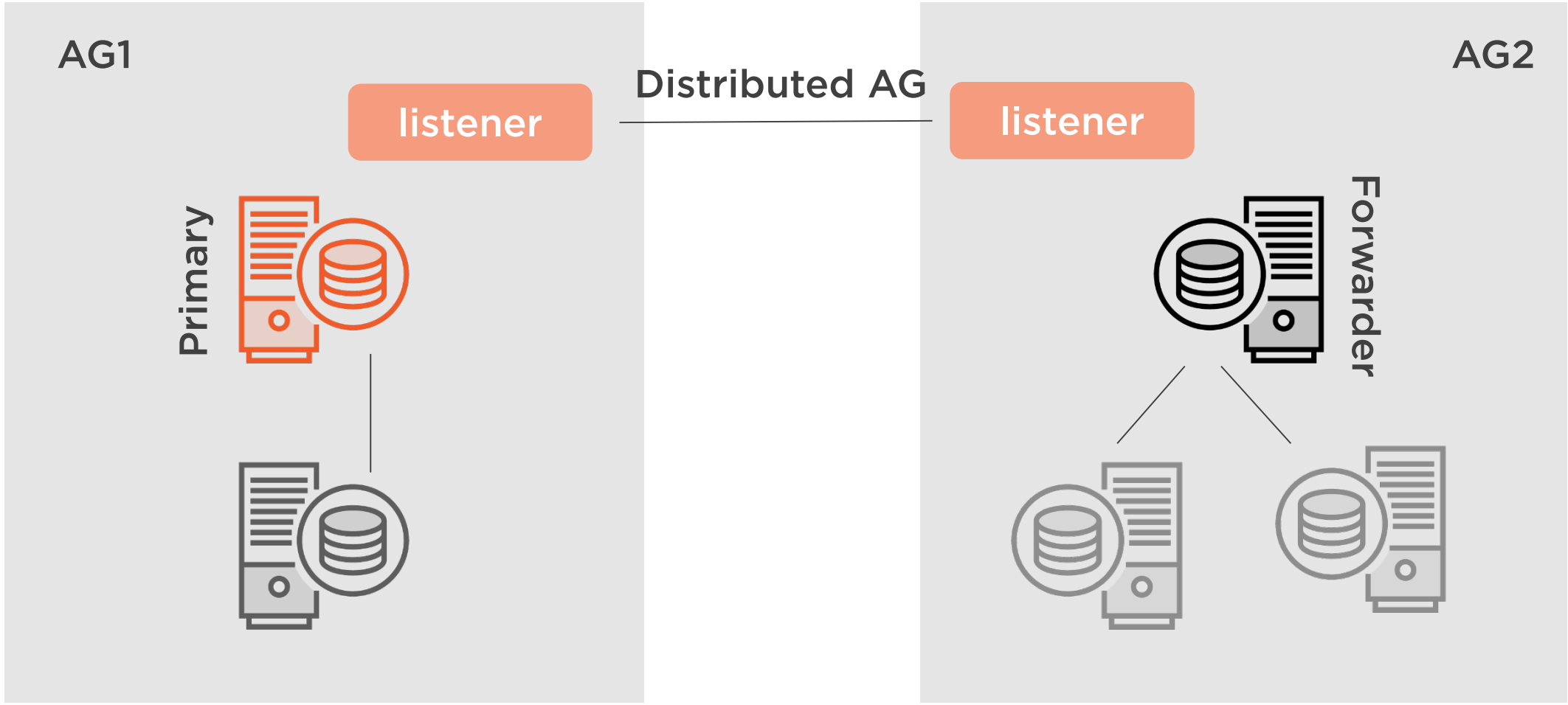
Single Availability Group



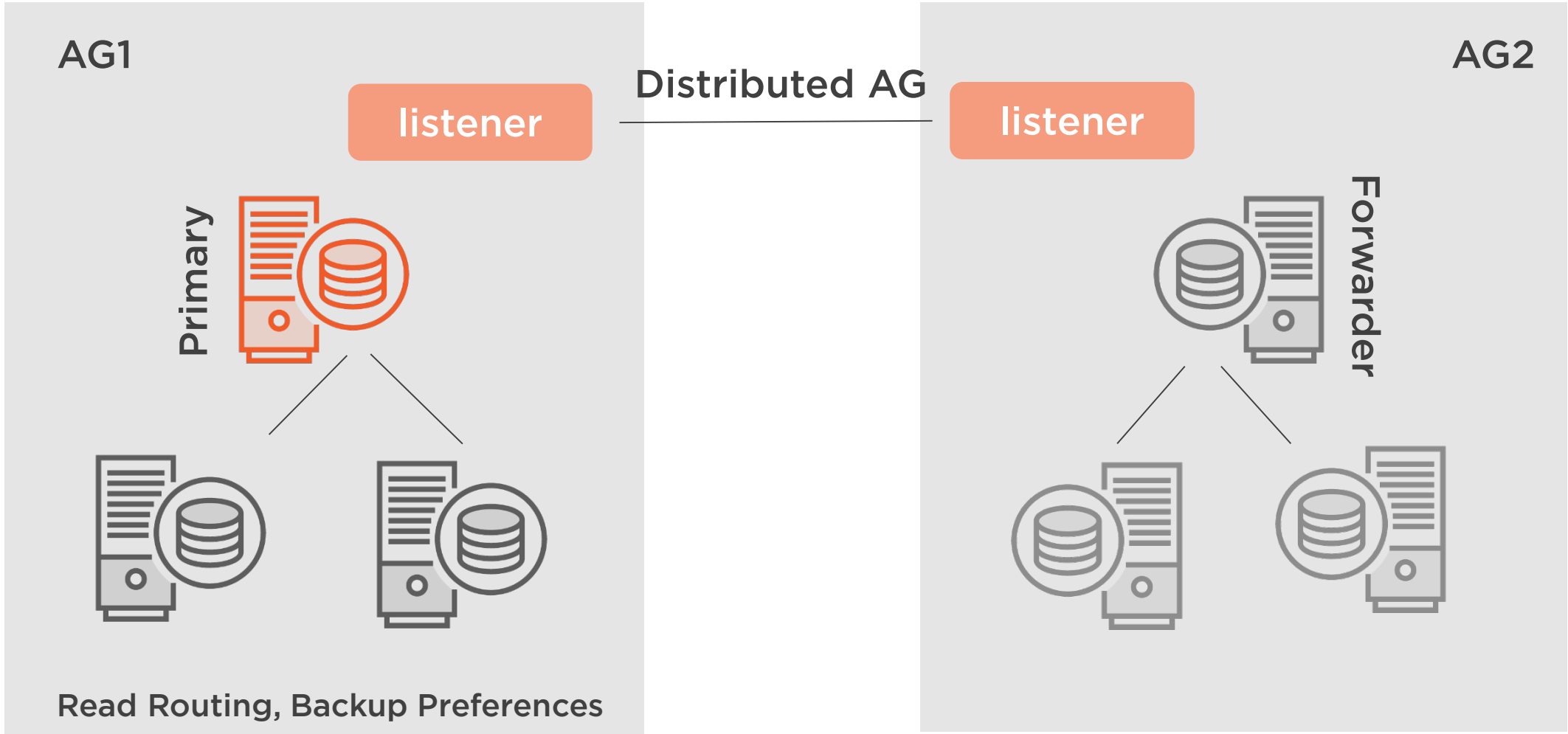
Two Availability Groups



Two Availability Groups



Distributed Availability Group



Distributed Availability Groups

Extend the maximum number of nodes in an AG solution

Supported on SQL Server 2016 and later

Administration and configuration limited to TSQL scripting

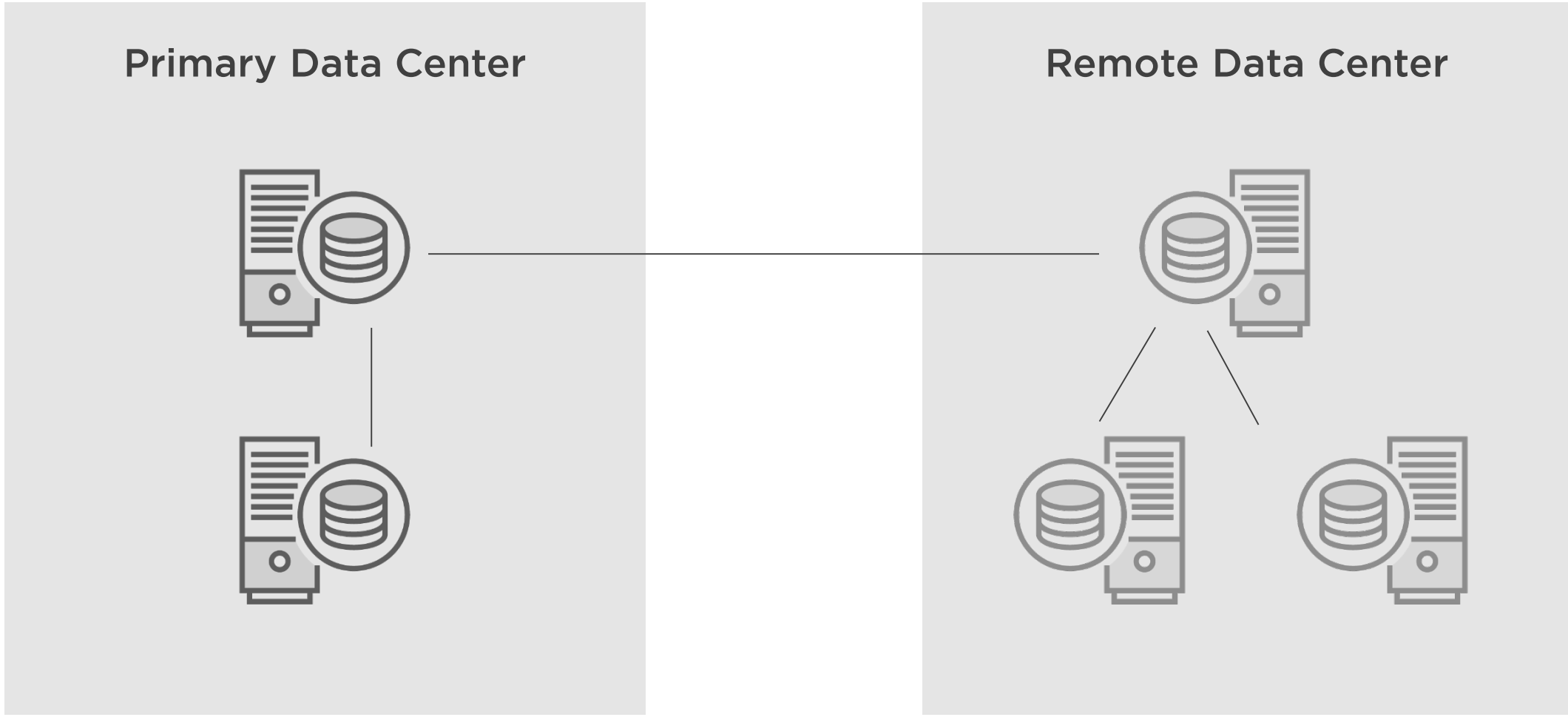
Distributed Availability Groups

Extend the maximum number of nodes in an AG solution

Supported on SQL Server 2016 and later

Administration and configuration limited to TSQL scripting

Distributed Availability Groups



Step 1 – Create Primary AG

```
CREATE AVAILABILITY GROUP AG1
WITH <options>
FOR DATABASE <db1>, <db2>, <db3>
REPLICA ON 'node1' WITH (<options>),
REPLICA ON 'node2' WITH (<options>);

ALTER AVAILBILITY GROUP AG1
ADD LISTENER 'AG1' WITH (<options>);
```

Step 2 - Create Secondary AG

```
CREATE AVAILABILITY GROUP AG2
FOR
REPLICA ON 'nodeA' WITH (<options>),
REPLICA ON 'nodeB' WITH (<options>);

ALTER AVAILBILITY GROUP AG2
ADD LISTENER 'AG2' WITH (<options>);
```

Step 3 – Listener IP

```
-- run on all participating nodes in both AGs

ALTER ENDPOINT [HADR]
AS TCP (LISTENER_IP = ALL)

-- select * from sys.tcp_endpoints to identify names
```

Step 4 – Grant Create Any Database

```
-- run on both participating AGs

ALTER AVAILABILITY GROUP [AG1] GRANT CREATE ANY DATABASE;
ALTER AVAILABILITY GROUP [AG2] GRANT CREATE ANY DATABASE;
```

Step 5 – Create Distributed AG on Primary

```
CREATE AVAILABILITY GROUP [AGDIST]
WITH (DISTRIBUTED)
AVAILABILITY GROUP ON
    'AG1' WITH
          (<options>),  -- specify desired seeding_mode
    'AG2' WITH
          (<options>);  -- specify desired seeding_mode
```

Step 6 – Join Distributed AG on Secondary

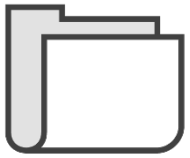
```
ALTER AVAILABILITY GROUP [AGDIST]
JOIN
AVAILABILITY GROUP ON
    'AG1' WITH
          (<options>),  -- same options
    'AG2' WITH
          (<options>);  -- same options
```

Step 7 – Join DBs on Secondary Secondaries

```
-- restore secondary databases with no recovery if necessary
-- wait until databases are caught up and show restoring

ALTER DATABASE [db1] SET HADR AVAILABILITY GROUP = [AG2];
ALTER DATABASE [db2] SET HADR AVAILABILITY GROUP = [AG2];
ALTER DATABASE [db3] SET HADR AVAILABILITY GROUP = [AG2];
```

Common Issues

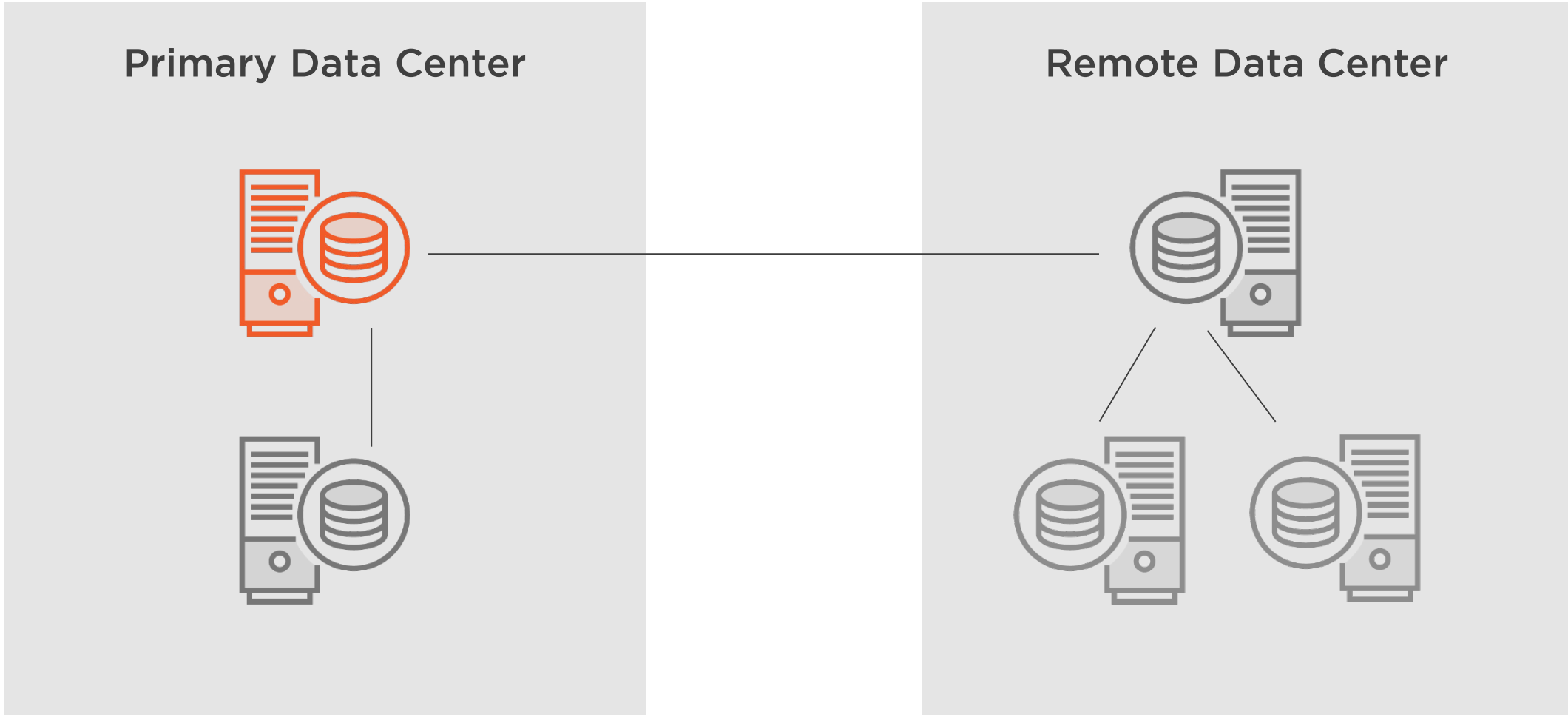


File paths and drive letters must match on all replicas

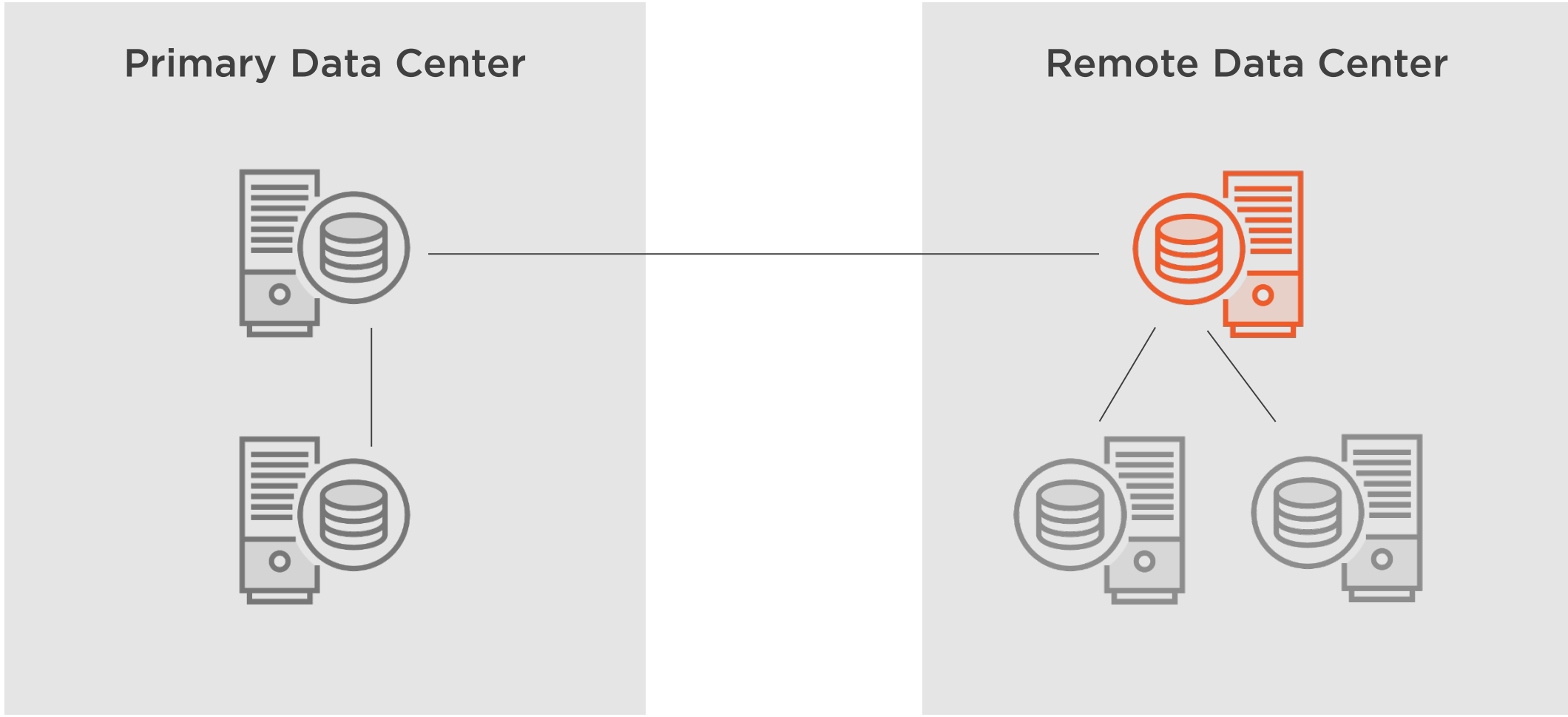
Make sure firewalls allow endpoint and listener ports both directions

Ensure databases seeded or restored properly prior to joining

Distributed Availability Groups



Distributed Availability Groups



Distributed Availability Group Failover

Set the distributed AG to synchronous commit on primary and forwarder(s).

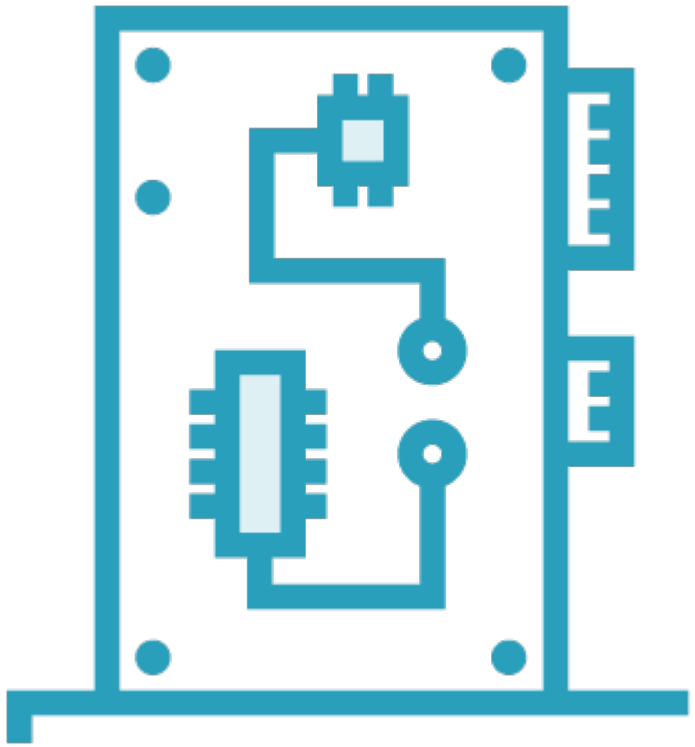
Wait until the distributed AG is synchronized.

On the global primary replica, set role to SECONDARY.

Verify failover readiness.

Fail over the primary availability group.

Failover Cluster Network Configuration



Windows 2003 and earlier failover clusters often required manual network configuration

Windows 2008 and later failover clusters are much more automated and hands off

Heartbeats

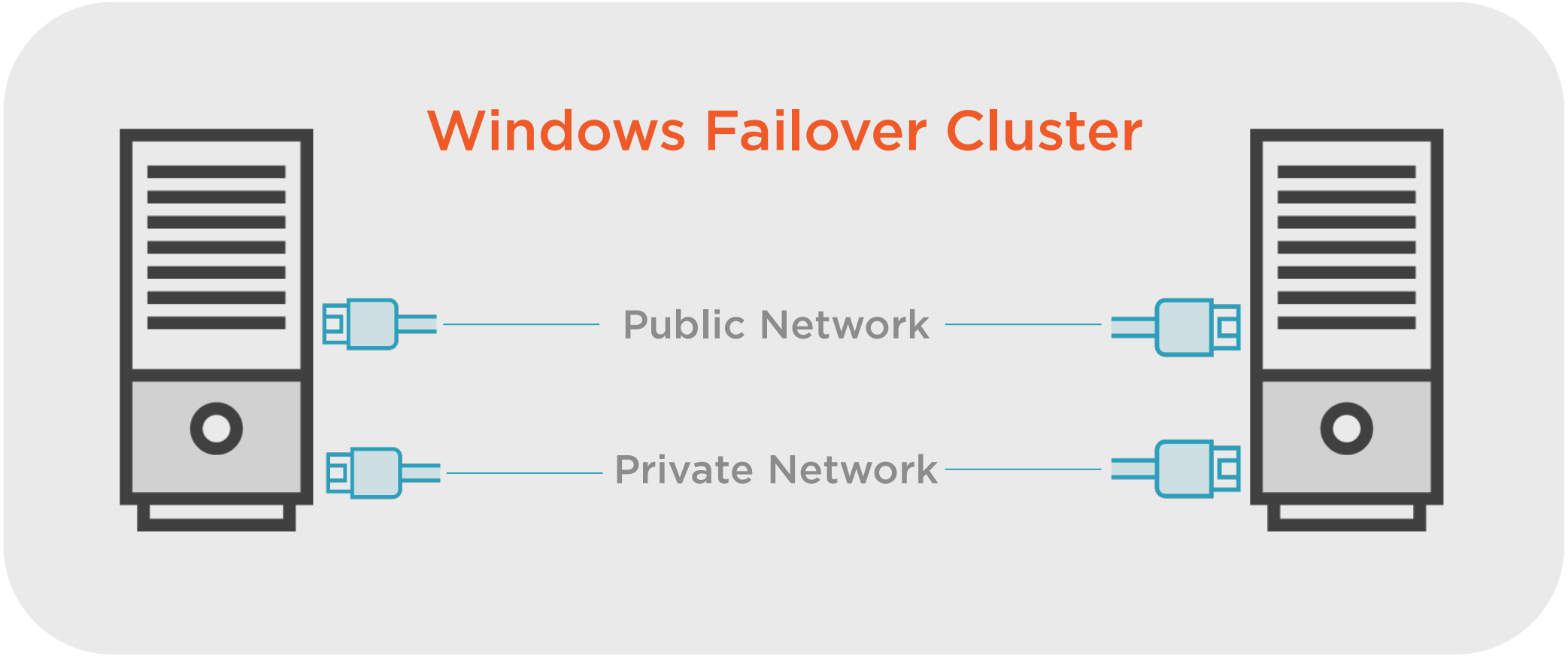
Heartbeat is a lightweight packet exchanged to verify cluster node health

More than simple ping response

Are you awake? Yes, I'm awake, are you awake?

“Heartbeat Network”

“Heartbeat Network”



Heartbeats

Windows 2008 and later automatically make use of all available networks

Uses internal logic to determine the best route available

Includes redundancy and fault tolerance through NetFT

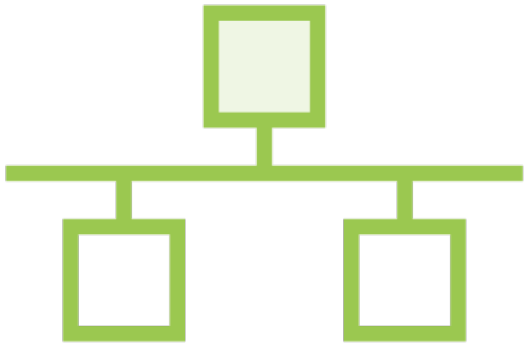
Failover Cluster Network Resources

Elden Christensen – Failover Cluster Network Essentials

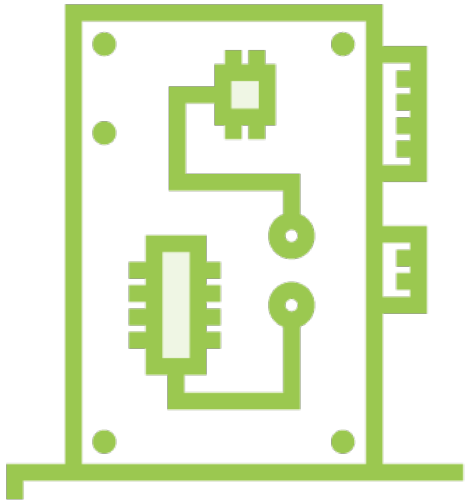
TechEd North America - https://bit.ly/2qajZDn

Core Team Best Practices - https://bit.ly/2R67pA5

Quality of Service Design



NIC Teaming



Multiple Physical Cards

Multiple VLANs

Failover Network Traffic Prioritization

Windows prioritizes best path available and types of cluster communication

Priority is based on calculated cost metrics

Least cost determined is prioritized for heartbeats

Items such as presence of default gateway affect network cost

Heartbeat Thresholds

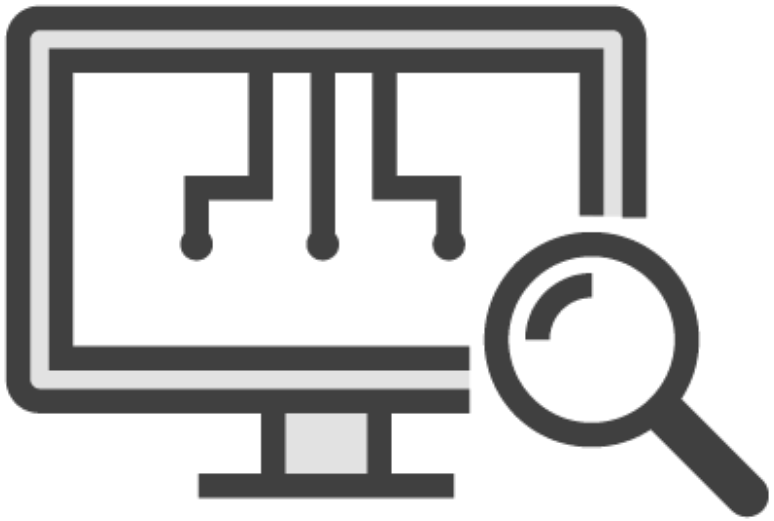
Network quality of service varies across implementations

By default the windows thresholds for heartbeat failures is aggressive

One heartbeat check a second

Failure declared after 5 missed heartbeats

Heartbeat Thresholds



TCP reconnect windows are longer than heartbeat thresholds

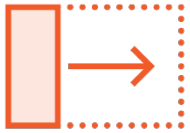
Network quality of service is borderline

Cluster failovers may appear as false positives

Always troubleshoot network quality of service first

Potential Side Effects to Modifying Thresholds

May defeat the purpose of your cluster



Can extend time to recovery

Impact of true failures extended beyond SLAs

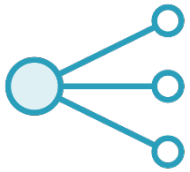
Heartbeat Thresholds

Property	Default	Maximum
SameSubnetDelay	1000 milliseconds	2000 milliseconds
SameSubnetThreshold	5	120
CrossSubnetDelay	1000 milliseconds	4000 milliseconds
CrossSubnetThreshold	5	120

Coordinate thresholds with environment and TCP reconnect windows

Modifying thresholds will usually be on the cross subnet properties

Cluster Network Relevant Topics



Quality of service best practices

Network prioritization and cost metrics

General cluster and network troubleshooting such as cluster.log

Different Environments Have Different Needs

SQL Server Availability Groups

SQL Server 2019 GA Release

No System DB Availability Group Support

This Course Current 2016 – 2019

PowerShell!

PowerShell

Capable of full scripting logic, structures, and development

Also extremely useful as a simple command line shell

Many tasks such as automation are much simpler in PowerShell

Good Luck

Russ Thomas

@SQLJudo