HA and DR

How the continuum works and where things fall

August 20th, 2021

Agenda

- Definitions
  - **Continuity**
  - **Redundancy**
  - **High Availability (HA)**
  - **Disaster Recovery (DR)**
  - **Recovery Point Objectives (RPO)**
  - **Recovery Time Objectives (RTO)**
- How HA, Backups and DR work together, and how they are separate
- How to identify single points of failure in each, and mitigate them
- Extra-IT concerns

Definitions

- Continuity
  - Being able to carry on after a failure has happened
  - Includes HA, DR, and Backups
  - Is a total plan for how life continues
- Redundancy
  - Having extra components to proceed un-interrupted after a failure
  - Should be essentially no outage with redundant components
  - EX: a RAID set, or synchronous SANs, or a NIC team, Vmware Fault Tolerance, etc.

Definitions

- High Availability
  - Being able to immediately recover after a failure, automatically, with a short outage
  - Usually DOES impact users, but normally for only a minute or two, ex: VMware HA, Spanning Tree reconvergence failover
- Disaster Recovery
  - Being able to recovery after a time, from a loss of production data.
  - Normally is going to be coming up on new hardware

The relationship between HA/DR/Backups

- Redundancy – Using duplicate components to **PREVENT** a failure
- High Availability – Using duplicate components to **SURVIVE** a failure
- Disaster Recovery – Using duplicate components to **RECOVER** from a failure
- Backups – Using duplicate components to **REBUILD** from a failure

Definitions

- Recovery Point Objective (RPO)
  - The point in time that the recovery is happening to
  - Often the time is measured in maximum lost data
  - EX: 2-hour RPO means up to 2 hours of data can be lost
- Recovery Time Objective (RTO)
  - How long it takes to recover data to that recovery point
  - Often is measured in the number of hours from disaster declaration until the service is back up and running
  - EX: 4-hour RPO means it takes up to 4 hours to bring the resource back online

Definitions

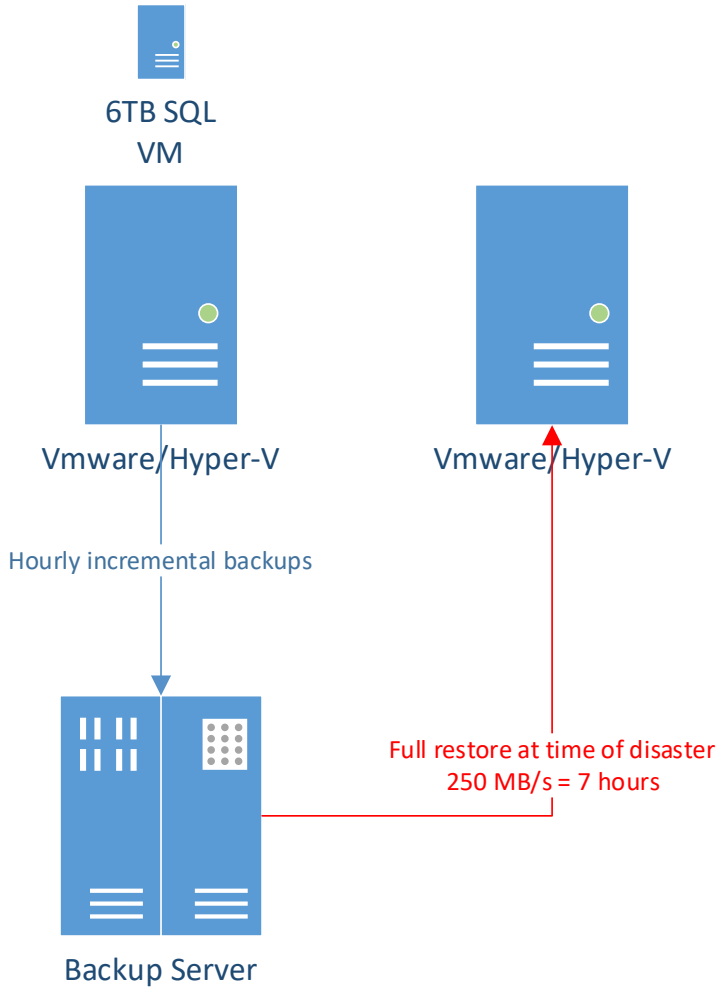
- Backup – Making a copy of the data
- Backup Copy – Copying the data to another site
- VM replication – Copying the VM to another host where it's able to be turned on
- Synchronization – Keeping both copies in lockstep of each other

The relationship between RPO and RTO

- A low RPO doesn't necessarily dictate a low RTO.

Example:

- Backup a 6TB SQL server every hour
- RPO is 1 hour
- A FULL restore at 250MB/s will take 7 hours (RTO)

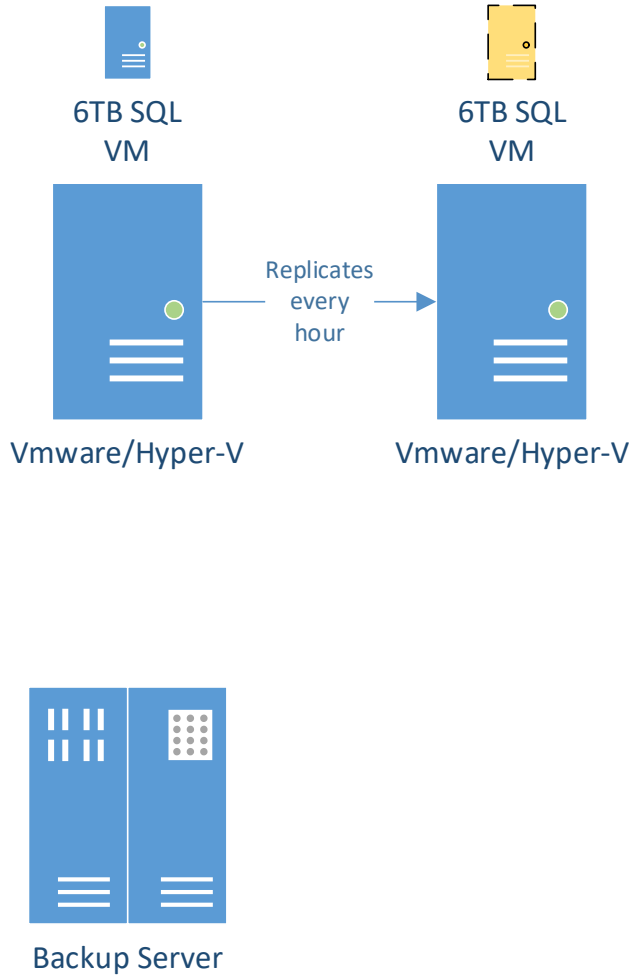


The relationship between RPO and RTO

- A low RPO doesn't necessarily dictate a low RTO.

Example 2:

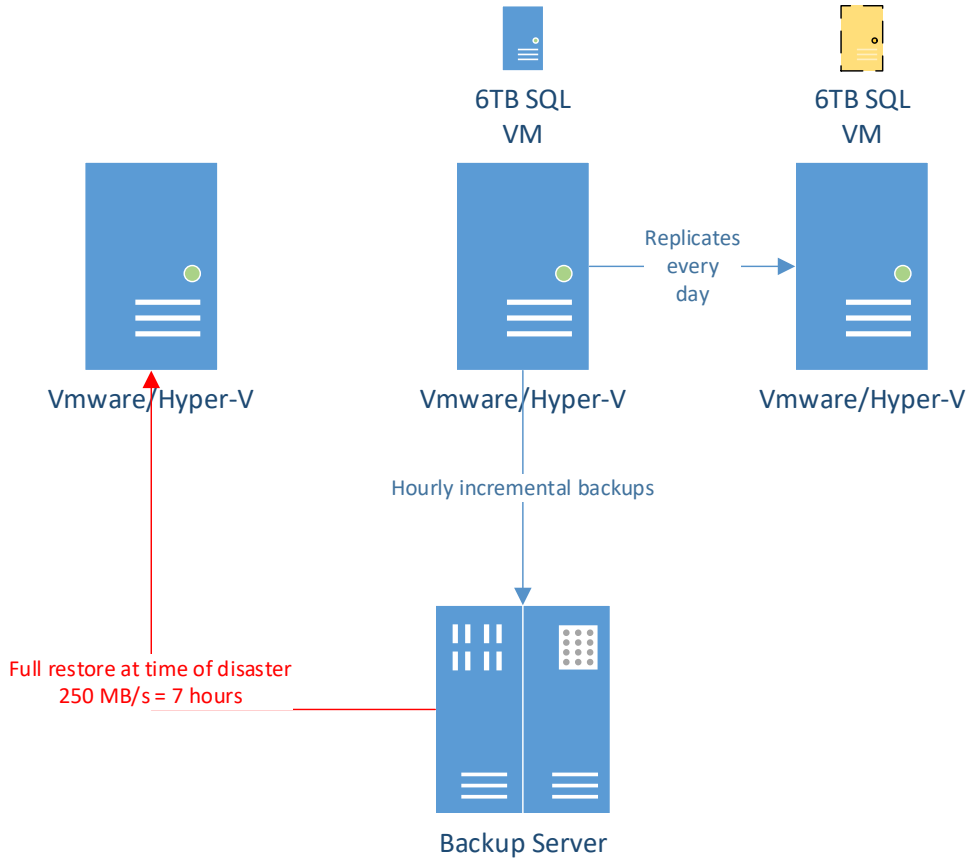
- Replicate a 6 TB SQL VM every hour to another server
- RPO is 1 hour
- Powering on the replica may only take 3 minutes (RTO)



The relationship between RPO and RTO

Example 3:

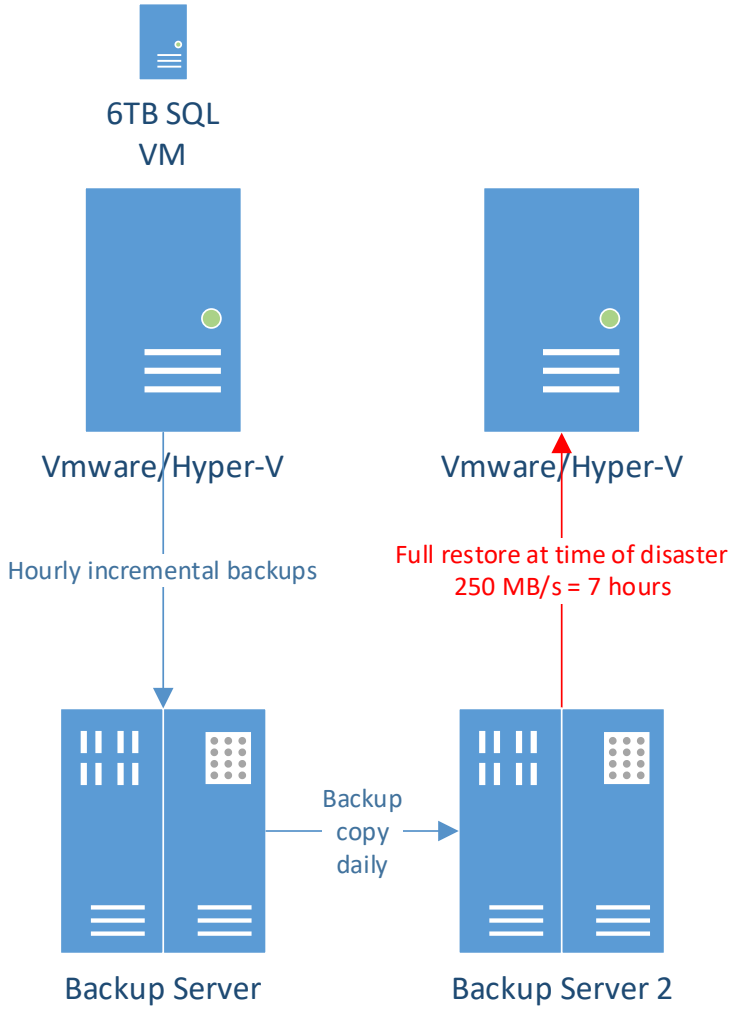
- Backup a 6 TB SQL server every hour locally
- Replicate a 6 TB SQL VM to another server daily
- RPO for local recovery is 1 hour
- RTO for local full restore might be 7 hours
- RPO for secondary site is up to 24 hours
- RTO for secondary site might be 3 minutes



The relationship between RPO and RTO

Example 4:

- Backup a 6 TB SQL server every hour locally
- Copy the backup to another server daily
- RPO for local recovery is 1 hour
- RTO for local full restore might be 7 hours
- RPO for secondary site is up to 24 hours
- RTO for secondary site might 7 hours



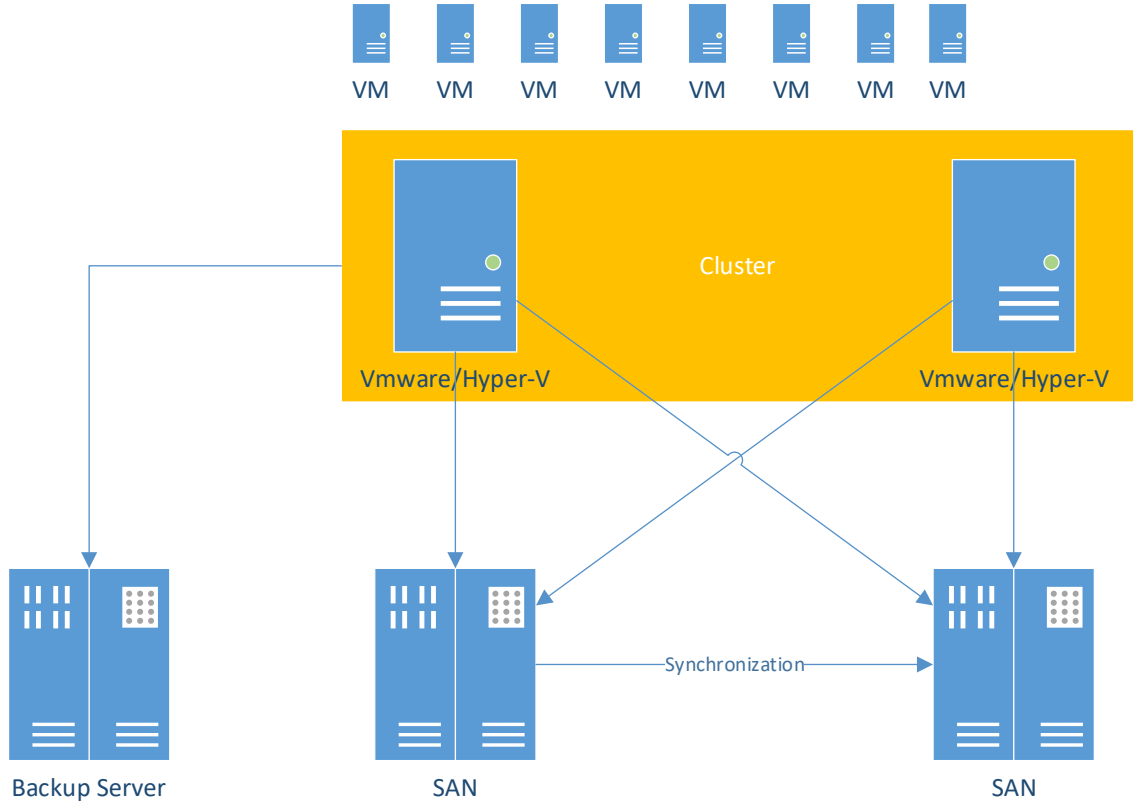
HA vs DR

- A minor change in configuration can change which you're doing
- Example
  - Two SANs at two different sites
  - High speed connectivity
  - ESXi hosts at two sites
  - Is it HA or DR?

HA vs DR

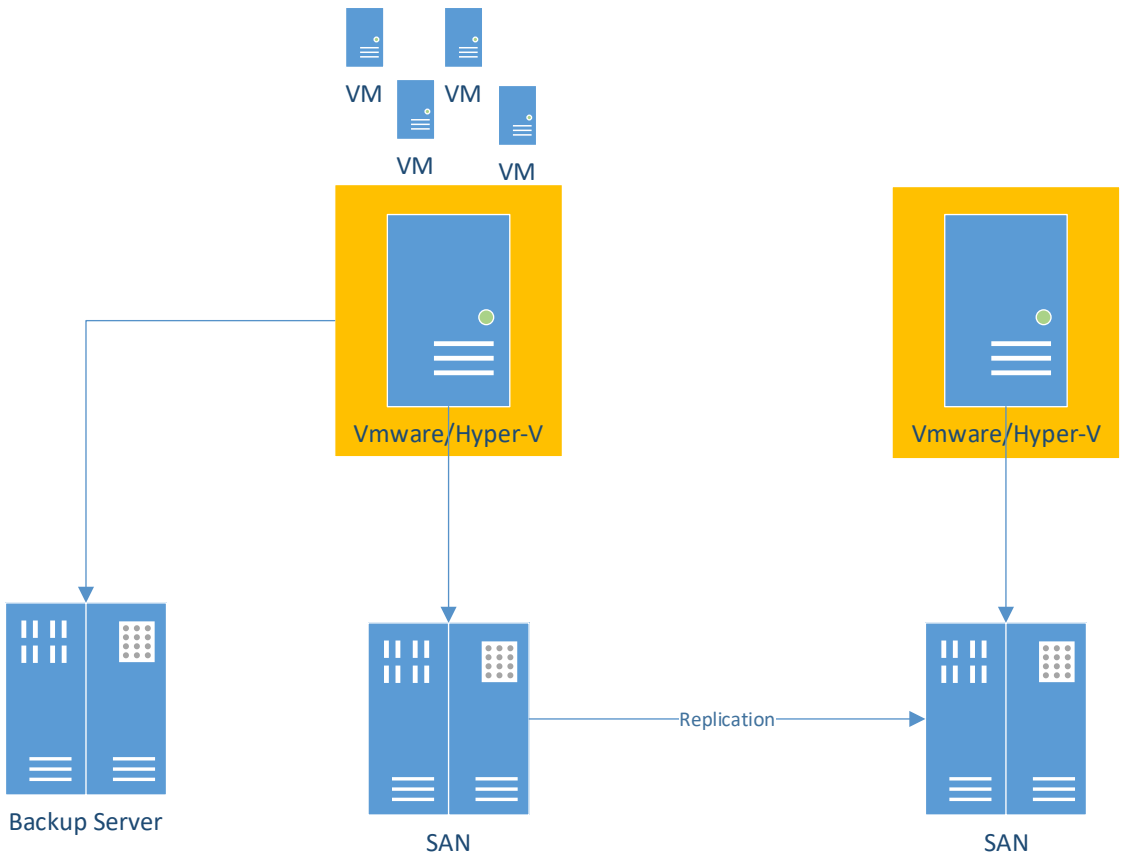
High availability

- If either SAN fails, there are paths to the other ones
- Workloads can traverse between hosts
- RPO – 0
- RTO - minutes



HA vs DR

- Disaster Recovery
- If left SAN fails, that cluster dies
- Workloads are siloed
- RPO – Could be 0 depending on configuration
- RTO – Could be hours depending on process



HA vs DR

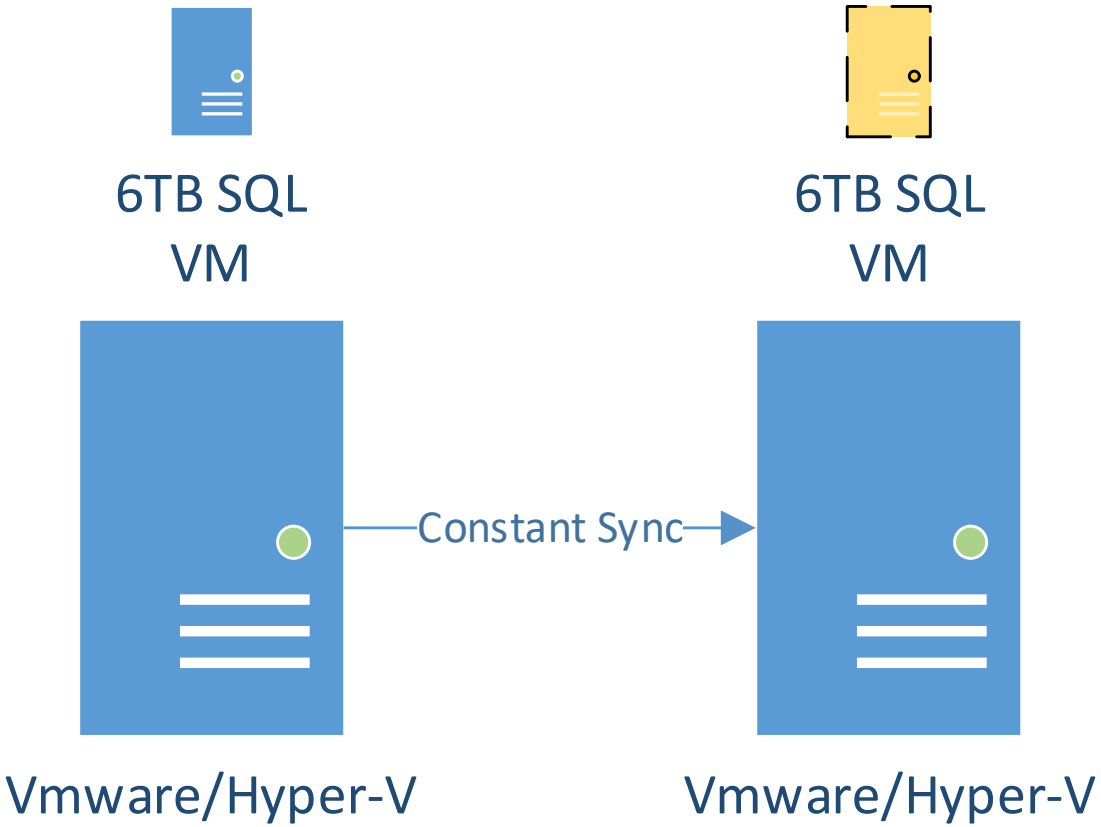
Which do you do?

- Benefits of HA
  - Instant failover
  - Whole site recovery automatically
  - No data loss
- Benefits of DR
  - Decision has to be made to move over
  - Could be a time delay on replica (or using snapshots) for older points
  - Could be better for certain recoveries
- Problems with HA
  - Could failover automatically when you don't want
  - If corrupt data at one side, corrupt at another
- Problems with DR
  - Slower failover – have to manually do it
  - Data could be lost depending on replication period

HA vs DR

What about smaller scale failures

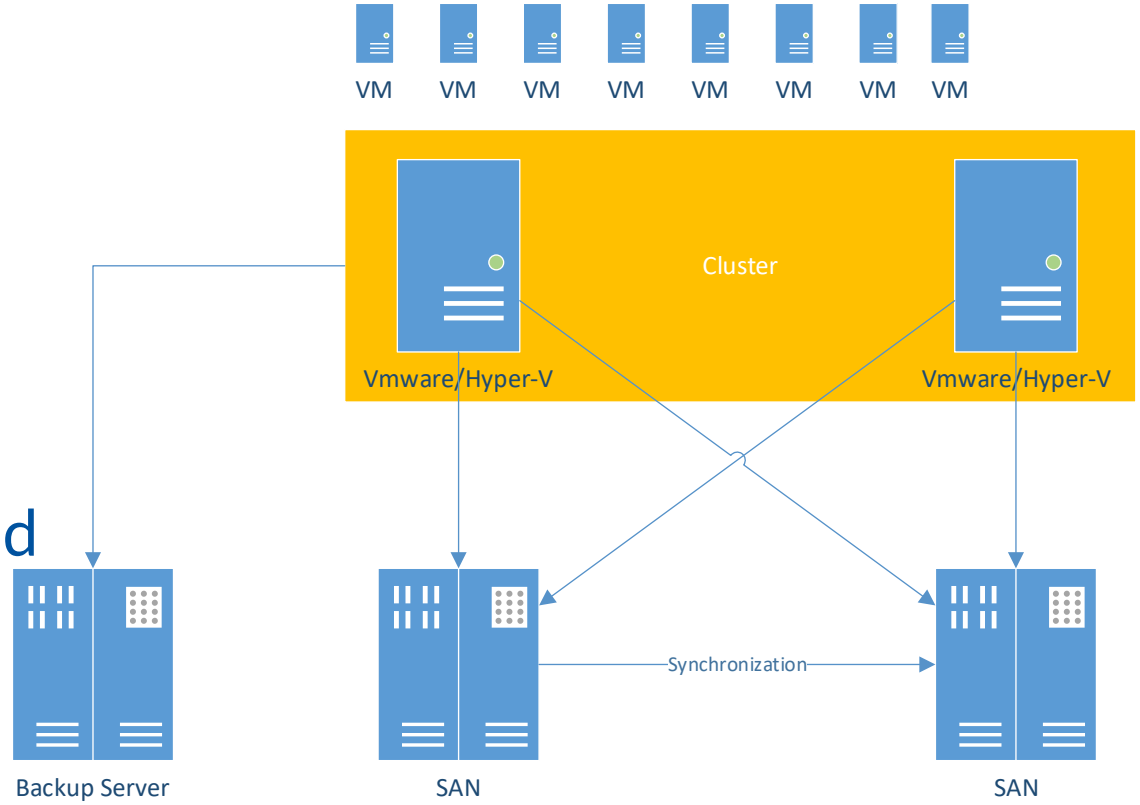
- What if a single VM dies?
- Redundancy (VMware FT)
  - No impact as long as the storage is happy.



HA vs DR

What about smaller scale failures

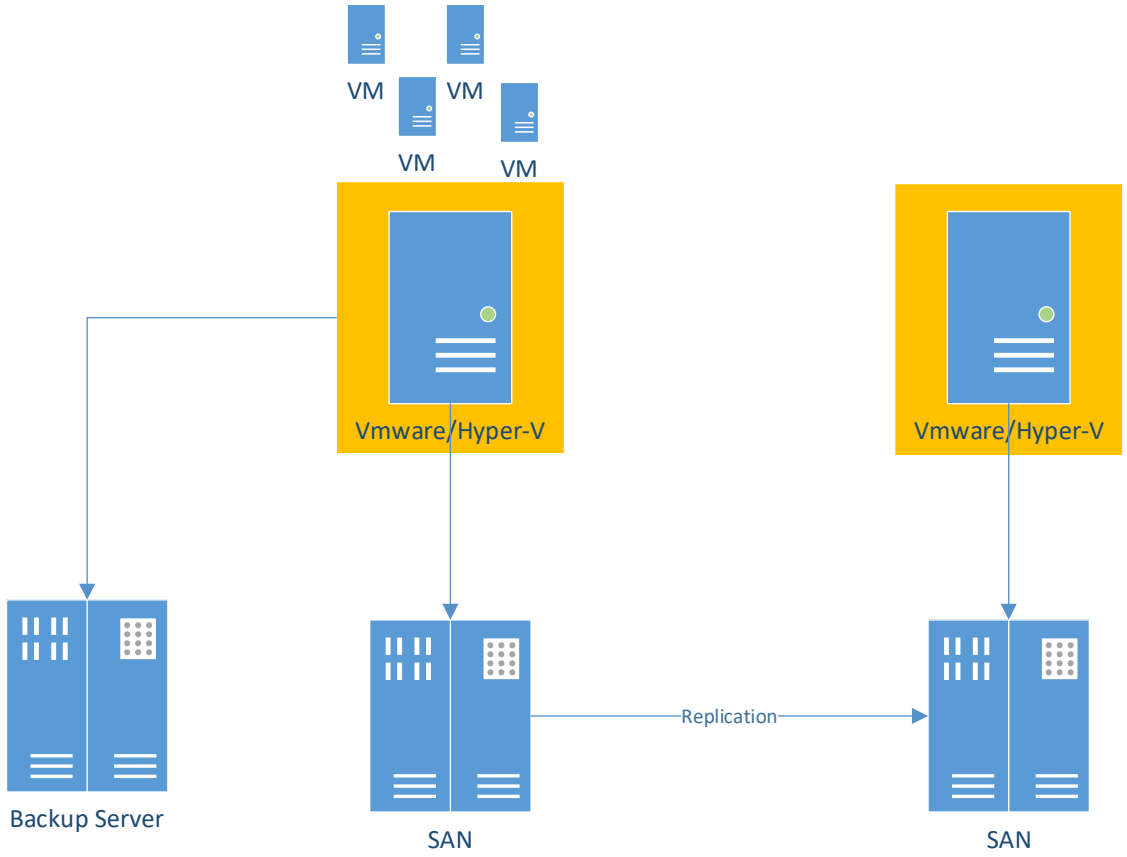
- What if a single VM dies?
- HA
  - Was it a hardware problem?
    - Probably fixed when it boots
  - Was it a software problem?
    - Probably still broken unless you did OS level HA [failover clustering]
  - Restore from backup?
  - Back to 7 hours



HA vs DR

What about smaller scale failures

- What if a single VM dies?
- DR
  - New hardware
  - Older point based on replication
  - Probably fixed
  - But can you get to it in the DR site?
  - Can you fail it back over?
  - Can you back it up for now?



HA vs DR

- What if a single VM dies?
- There's no panacea.
- A combination of redundancy, HA, DR, and Backups
- May be a combination of multiple HA situations

HA vs DR

Where do backups fall?

- It depends ☺
- At small businesses backups may be the only continuity
  - Essentially, everything is a DR event
  - Pull the backup hard drive, start the restore
- At larger businesses, things can get blurry
  - EX: Veeam with instant-on can bring a server back online with an RTO of only a few minutes
  - Various products doing CDP can get an RPO of only a few seconds
  - That can fall somewhere on the continuum.

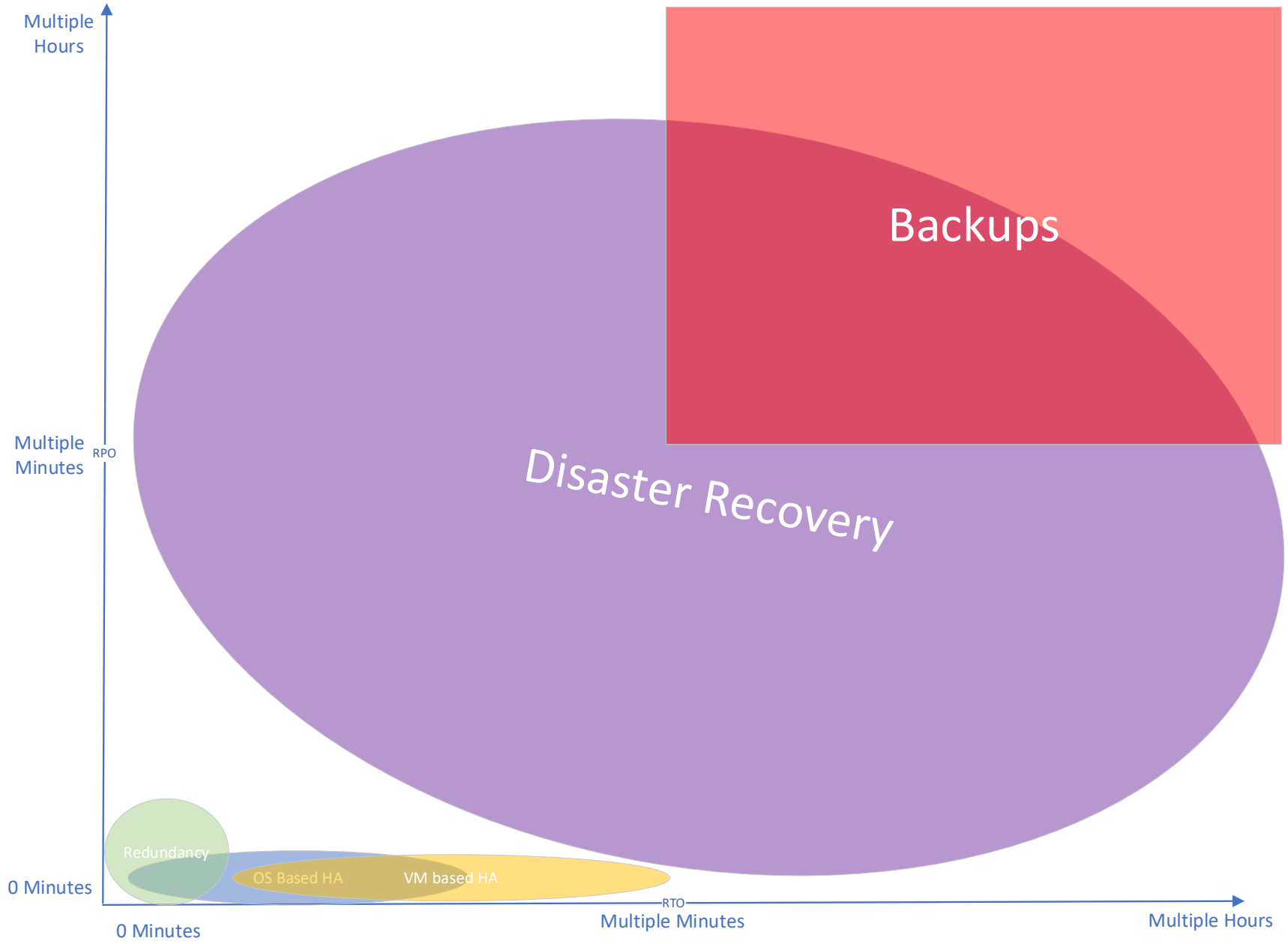
Redundancy vs HA vs DR vs Backups

All complimentary, all different

- Topics are normally completely different
- Work in separate manners
- Work together though, to form a full solution
- Each separate piece of the environment should have at least ONE of these
- They all give a different level of survivability with different caveats

HA vs DR

Ugly diagram



How to identify single points of failure

Real talk.

- There will ALWAYS be a single point of failure
- No really, there will.
- No?
- Do you have data copies off planet?
- How about out of the solar system?
- That's what I thought.

How to identify single points of failure

Where in the world do I start?

- Start with the thing you're protecting, the application
- Step out to the OS and repeat
- Step out to the hypervisor (if applicable)
- Step out to the physical servers

- For each step think, for that of our continuum:
  - Redundancy
  - High Availability
  - Disaster Recovery
  - Backups

How to identify single points of failure

Where in the world do I start?

- Step out to the rack delivery
  - Network they plug into
  - Power everything plugs into
  - Rack its physically
- Step out to the building it's in
- Step out to the city it's in
- Step out to the state it's in
- Step out to the country it's in
- Step out to the planet it's on

- For each step think, for that of our continuum:
  - Redundancy
  - High Availability
  - Disaster Recovery
  - Backups

How to identify single points of failure

Redundancy Examples

- At some point for each of those, you probably went “this is ridiculous”
  - That’s probably where that tier stops
- For example:
  - Application redundancy is often unfeasible unless it scales out
  - OS redundancy - probably not viable
  - Hypervisor redundancy – VMware FT
  - Physical device redundancy – RAID, Power Supplies, NIC teaming, Fans, etc. but probably not ‘redundant’ processors

How to identify single points of failure

Redundancy Examples

  - Redundant rack delivery – Power, Cooling, cabling
- After that, it starts to get tricky:
  - Redundant buildings? Only works if you're stretching redundancy at a higher level
  - Redundant cities, states, countries, planets? Nearly impossible

How to identify single points of failure

High Availability Examples

- Then think about High Availability at each level:
  - Application – Exchange DAGs for example
  - OS – Microsoft Failover Clustering
  - Hypervisor – Cluster
  - Physical device – the upper two layers protect against this
  - Rack delivery – Hopefully redundancy protects against this
  - Building – stretch HA/Metro Cluster between buildings
  - City – Can possibly do stretch HA, but it's hard
  - State – unless you're really close to a border, probably not feasible

How to identify single points of failure

High Availability Examples

- Then think about High Availability at each level:
  - Application – Exchange DAGs for example
  - OS – Microsoft Failover Clustering
  - Hypervisor – Cluster
  - Physical device – the upper two layers protect against this
  - Rack delivery – Hopefully redundancy protects against this
  - Building – stretch HA/Metro Cluster between buildings
  - City – Can possibly do stretch HA, but it's hard
  - State – unless you're really close to a border, probably not feasible
  - Country? Planet? - Probably not

How to identify single points of failure

Disaster Recovery Examples

- Then think about Disaster Recovery at each level:
  - Application – Async copy? {DAGs again}
  - OS – backups?
  - Hypervisor – Replication?
  - Physical device – other servers at another place
  - Rack delivery – other resources at another place
  - Building, city, state, country, planet – How close is it to production?

How to identify single points of failure

Backups Examples

- Then think about Backups at each level:
  - Application – how do you take them?
  - OS – how do you take them?
  - Hypervisor – do you take them?
  - Physical device – where do you store them?
  - Rack delivery – where do you store them?
  - Building – where do you store them?
  - City – where do you store them?
  - State – where do you store them?
  - Country - where do you store them?
  - Planet – where do you store them?

How to identify single points of failure

Struggling to think through single points of failure?

- Draw it out
- Include all systems
- Cut it in half
- Did any piece get bifurcated?
- Did any piece get left on an island all alone (your internet? The SAN?)
- That's potentially a problem!

How to identify single points of failure

Think about redundancy, HA, DR, backups, but for the business

- People – cross training? Or finding an email from 2 years ago?
- Business devices – Are there business devices connected that are single points of failure? (computer hanging off a press brake)
- Buildings – If the main warehouse burns down, where do employees work? CAN employees work?
- These things dictate how much redundancy/DR you really need

So how do they all come together

A common example

Level	Redundancy?	High Availability?	Disaster Recovery?	Backups?
Application	No	No	No	No
Operating System	No	No	Yes, replicated elsewhere	Yes, Yes, a thousand times, yes!
Hypervisor	No	Yes, Clustered	Yes, installed at another site	Yes, OS backups stored off hypervisor
Physical Device	Yes, RAID, Fans, PSU,	Yes, 2 network switches	Yes, provisioned at another site	Yes, OS backups stored on separate device
Rack Delivery	Yes, 2 PDU, dual AC	No	Yes, provisioned at another site	Yes, OS backups stored in different rack
Building	No	Yes, metro cluster	Yes, DR in a 3rd building	Yes, OS backups stored in different building
City	No	No	Yes, DR in another city	No
State	No	No	Yes, DR in another state	No
Country	No	No	No	No
Planet	No	No	No	No

So how do they all come together?

You can preclude some with others

- Do you need RAID if you have two copies?
- Do you need hypervisor HA if the application is giving it?
- Do you need server DR if you have backups?
- Do you need server backups if you have DR [YES! is always the answer]
- Think through each piece and each step of the way.

Putting a bow on it

Use your tiers together for success!

- Redundancy – Using duplicate components to **PREVENT** a failure
- High Availability – Using duplicate components to **SURVIVE** a failure
- Disaster Recovery – Using duplicate components to **RECOVER** from a failure
- Backups – Using duplicate components to **REBUILD** from a failure

Questions?

Thanks for attending!

Brent Earls
brent.earls@mirazon.com