ECE566
Enterprise Storage Architecture

Spring 2025

Business Continuity: Disaster Recovery

Tyler Bletsch
Duke University

Includes material adapted from the course "Information Storage and Management v2" (modules 9-12), published by EMC corporation.

Meta-notes

Notes I've added to the EMC stuff will appear in boxes like this one.

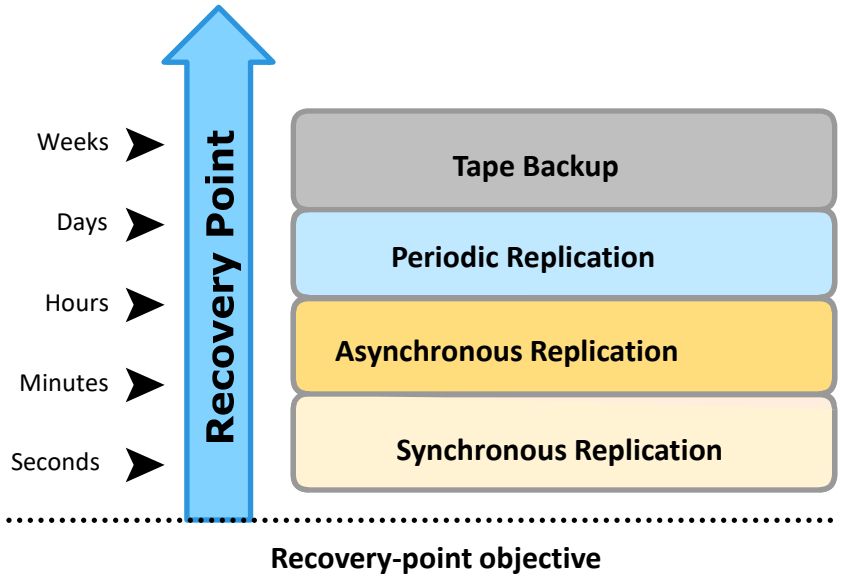
BC Terminologies – 1

- Disaster recovery
  - Coordinated process of restoring systems, data, and infrastructure required to support business operations after a disaster occurs
  - Restoring previous copy of data and applying logs to that copy to bring it to a known point of consistency
  - Generally implies use of backup technology
- Disaster restart
  - Process of restarting business operations with mirrored consistent copies of data and applications
  - Generally implies use of replication technologies

BC Terminologies – 2

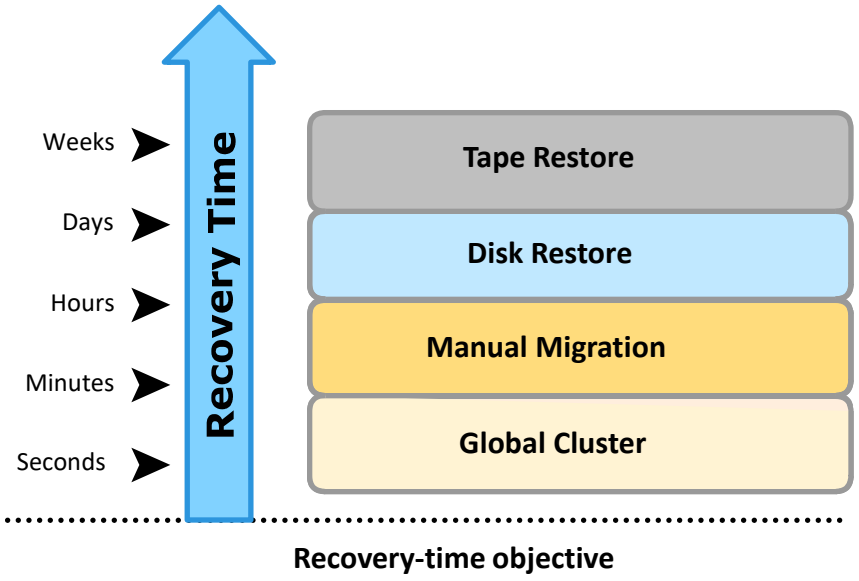
Recovery-Point Objective (RPO)

- Point-in-time to which systems and data must be recovered after an outage
- Amount of data loss that a business can endure



Recovery-Time Objective (RTO)

- Time within which systems and applications must be recovered after an outage
- Amount of downtime that a business can endure and survive



RPO vs RTO

Recovery Point Objective (RPO)

- How much did I lose?

Recovery Time Objective (RTO)

- How long until it's back?

Business Impact Analysis

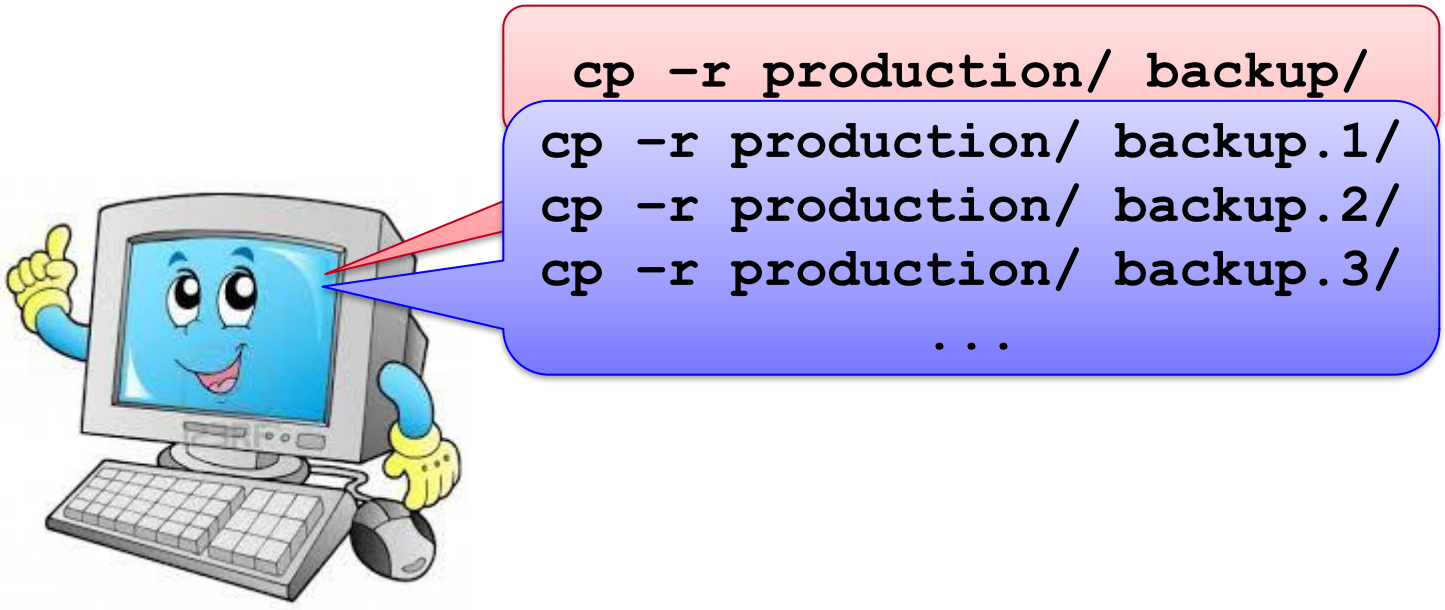
- Identifies which business units and processes are essential to the survival of the business
- Estimates the cost of failure for each business process
- Calculates the maximum tolerable outage and defines RTO for each business process
- Businesses can prioritize and implement countermeasures to mitigate the likelihood of such disruptions

Translation

Identify what will hurt the most to lose, spend your money there.

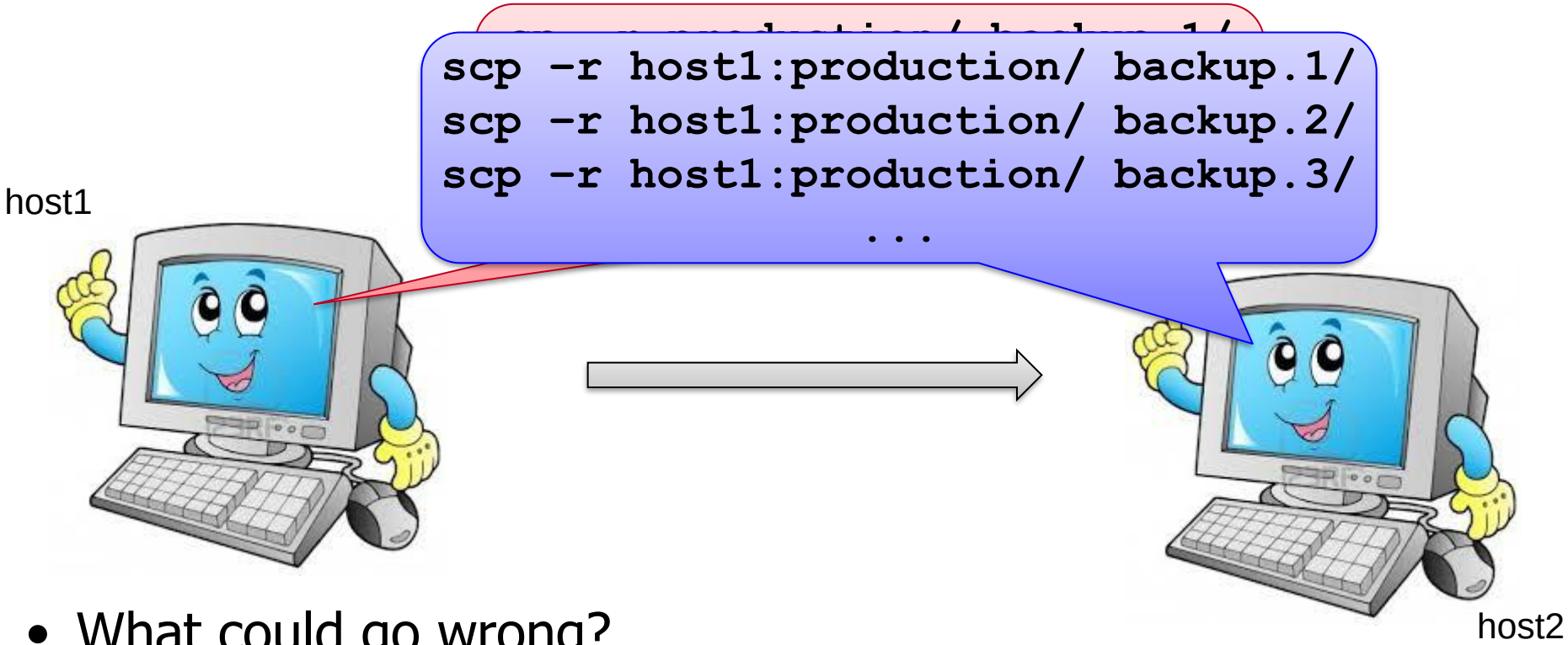
Backup and Archive

Building a backup solution



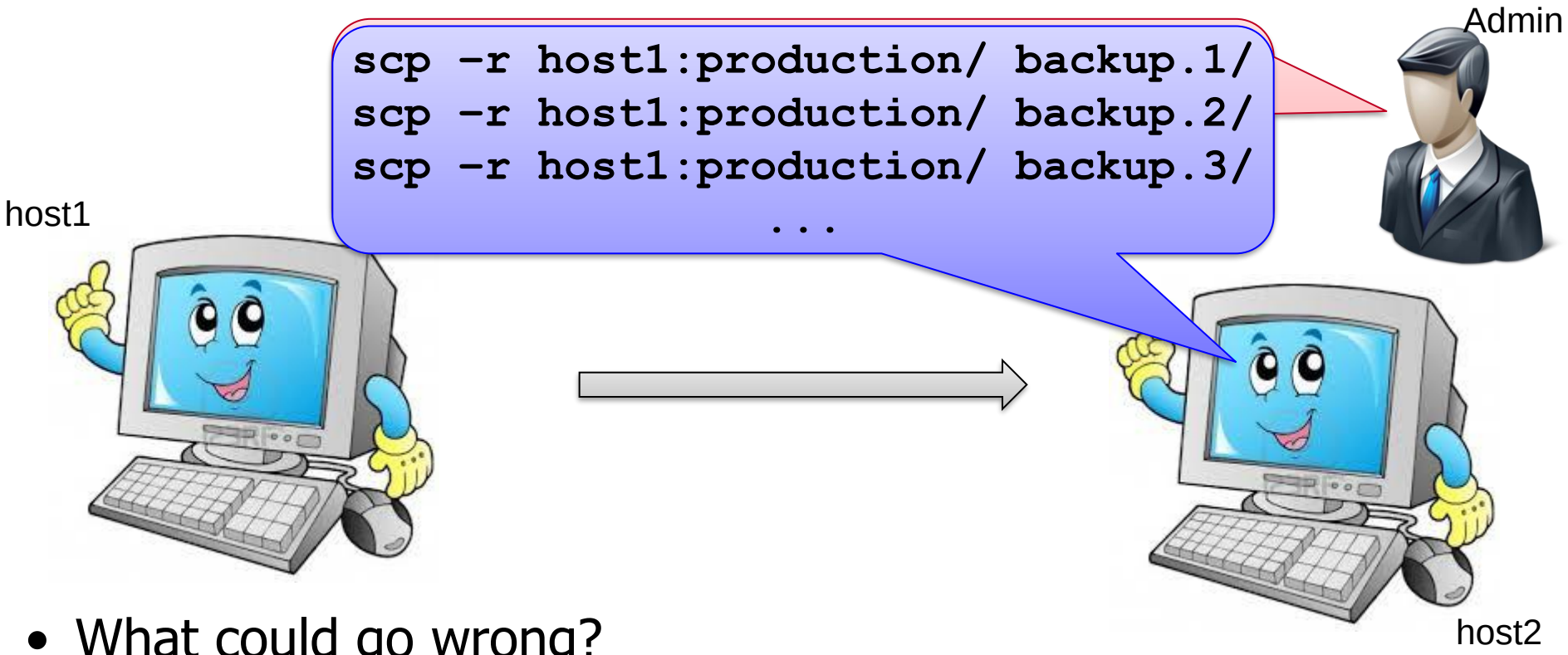
- What could go wrong?
  - Data corruption → Corrupt data overwrites backup → data loss
  - Solution: **multiple snapshots**

Building a backup solution



- What could go wrong?
  - System stolen/fails/destroyed → data loss
  - Solution: **separate systems**

Building a backup solution



- What could go wrong?
  - Admin forgets → data loss
  - Solution: **automation**

Building a backup solution

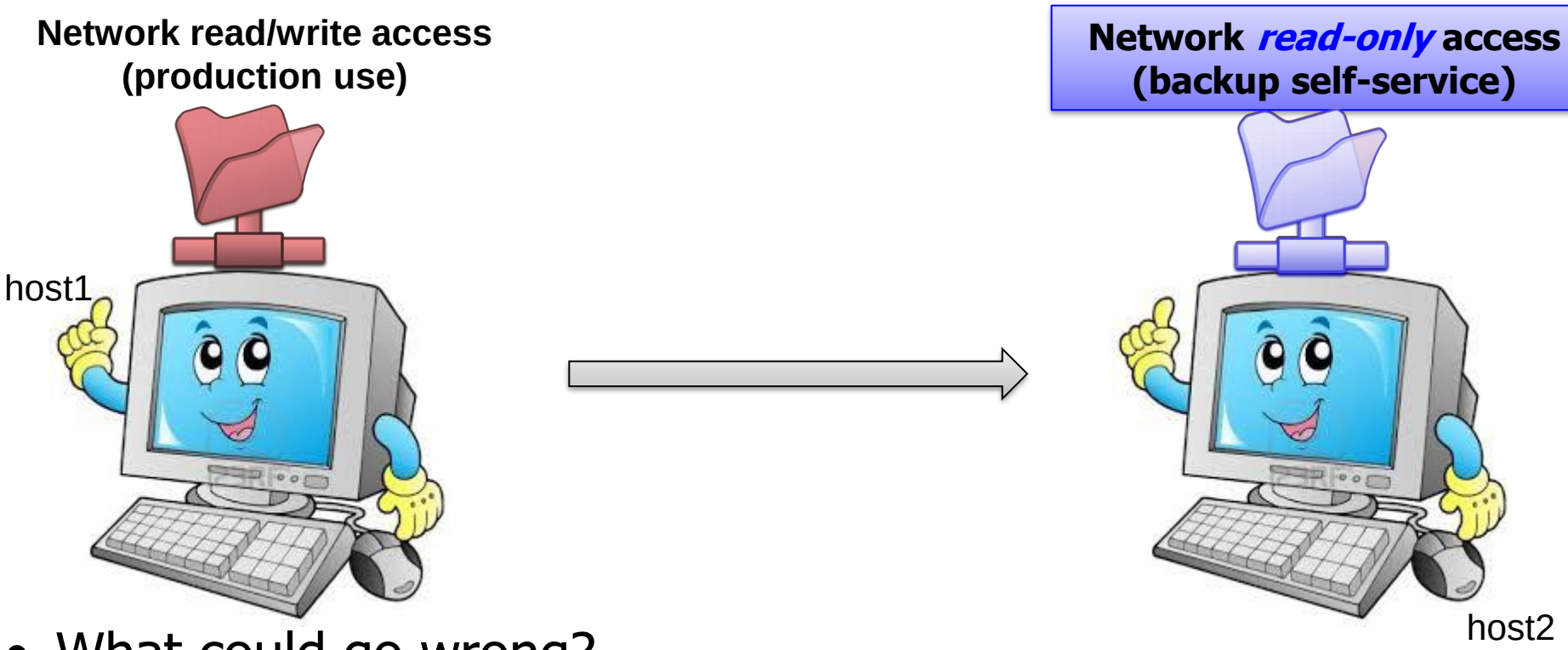
host1

Same admin access credentials

host2

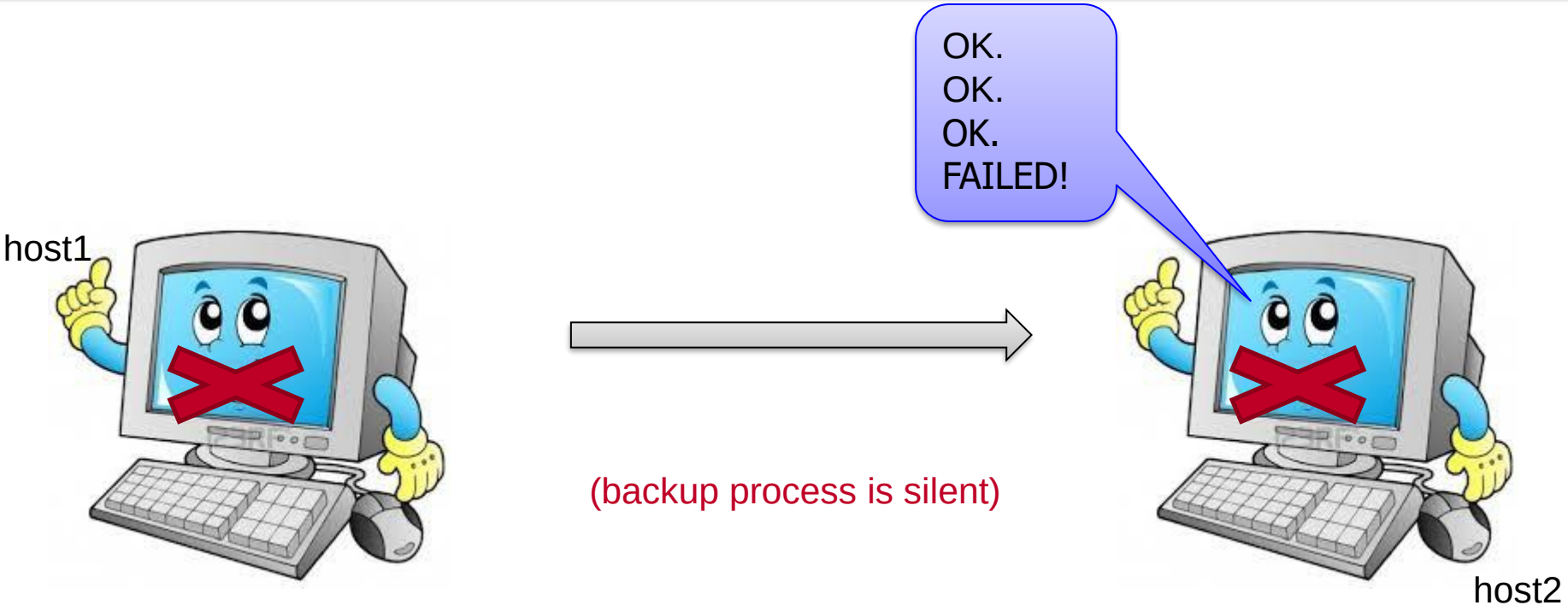
- What could go wrong?
  - Attacker gains one credential → Attacker can kill/corrupt all copies → data loss
  - Solution: **separate credentials for backup**

Building a backup solution



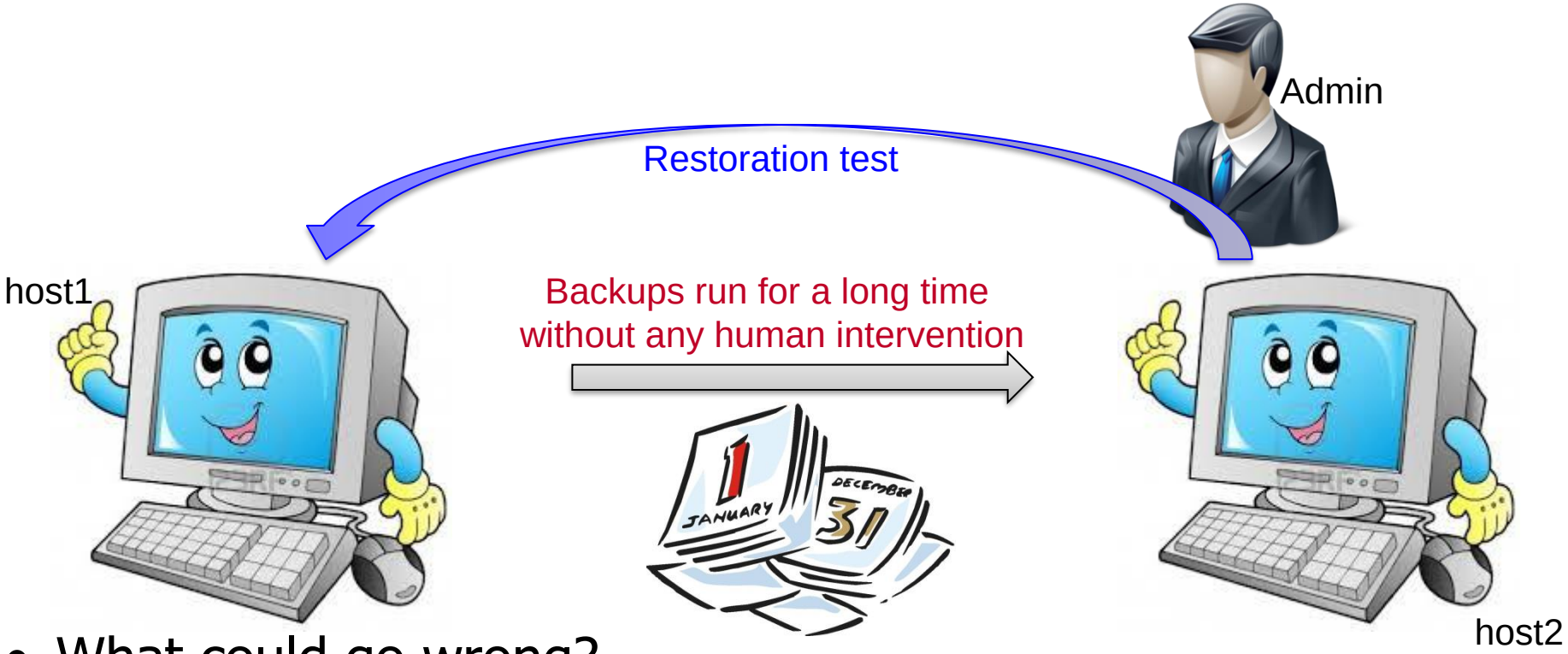
- What could go wrong?
  - User modifies primary and backup data → data loss
  - Solution: **backups must be unwritable**

Building a backup solution



- What could go wrong?
  - Backup server quietly dies → No backups kept for a while → data loss
  - Solution: **backups must report on success and alert on failure**

Building a backup solution



- What could go wrong?
  - Backups were done wrong all along → data not there when needed → data loss
  - Solution: **periodic restoration tests**

Tyler's Immutable Rules Of Backup
A BACKUP SOLUTION MUST:

1. Record changes to data **over time**
   - If I just have the most recent copy, then I just have the most recently corrupted copy. *RESULT: MIRRORING ISN'T BACKUP!!!!*
2. Have a copy at a **separate physical location**
   - If all copies are in one place, then a simple fire or lightning event can destroy all copies
3. Must be **automatic**
   - When you get busy, you'll forget, and busy people make the most important data
4. Require **separate credentials** to access
   - If one compromised account can wipe primary and secondary, then that account is a single point of failure
5. Be **unwritable** by anyone except the backup software (which ideally should live in the restricted backup environment)
   - If I can cd to a directory and change backups, then the same mistake/attack that killed the primary can kill the backup
6. Reliably **report** on progress and **alert** on failure
   - I need to know if it stopped working or is about to stop working
7. Have periodic **recovery tests** to ensure the right data is being captured
   - Prevent "well it apparently hasn't been backing up properly all along, so we're screwed"

If you encounter backups that don't meet these rules, explain the potential dangers until they do!

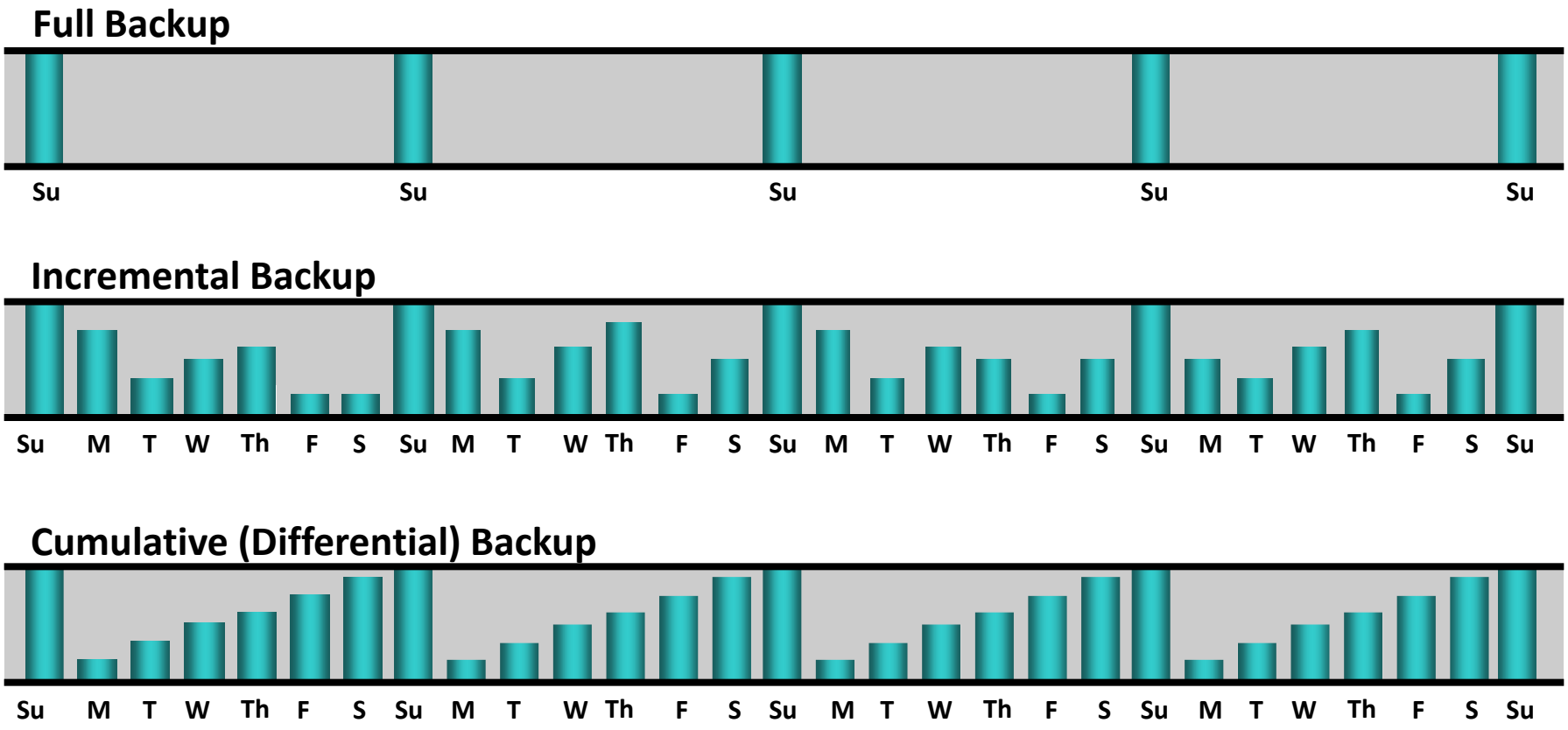
What is Backup?

Backup

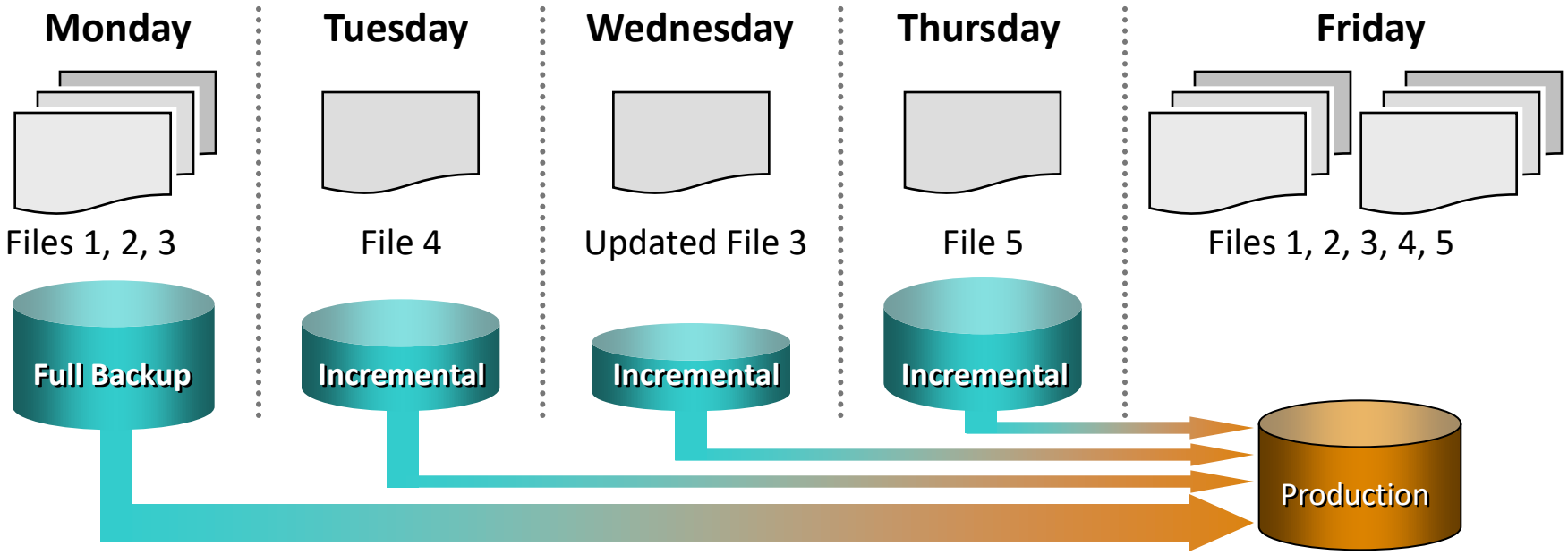
It is an additional copy of production data that is created and retained for the sole purpose of recovering lost or corrupted data.

- Organization also takes backup to comply with regulatory requirements
- Backups are performed to serve three purposes:
  - Disaster recovery
  - Operational recovery
  - Archive

Backup Granularity

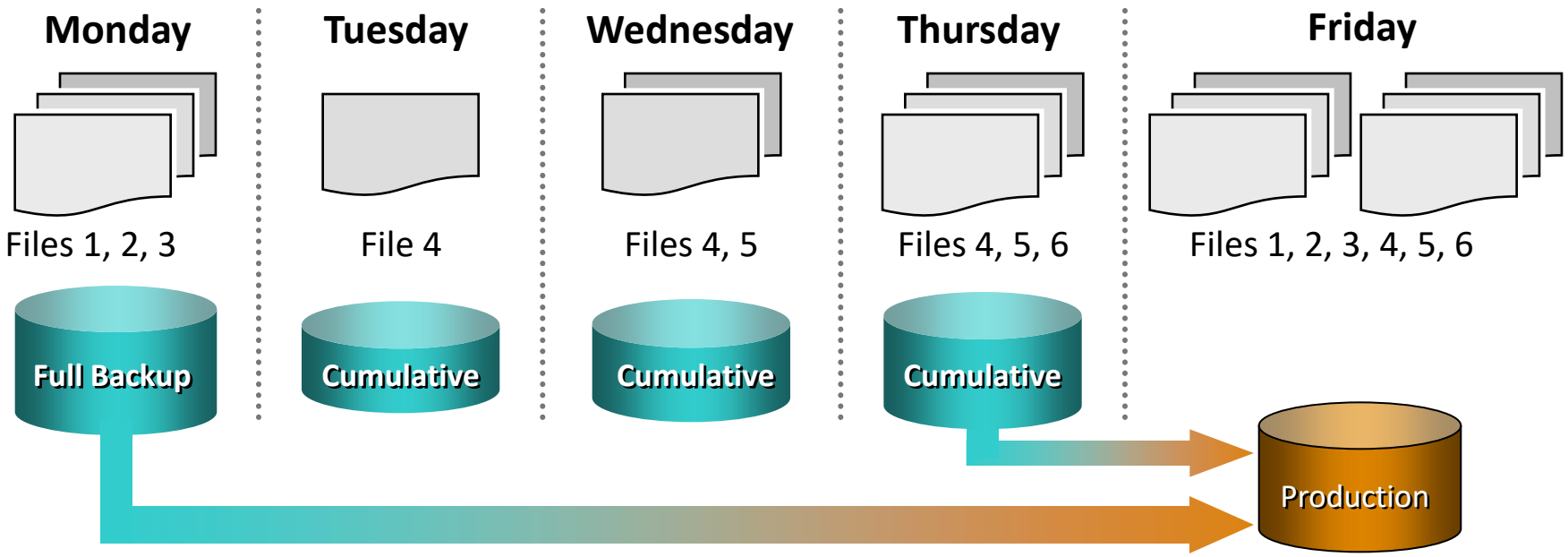


Restoring from Incremental Backup



- Less number of files to be backed up, therefore, it takes less time to backup and requires less storage space
- Longer restore because last full and all subsequent incremental backups must be applied

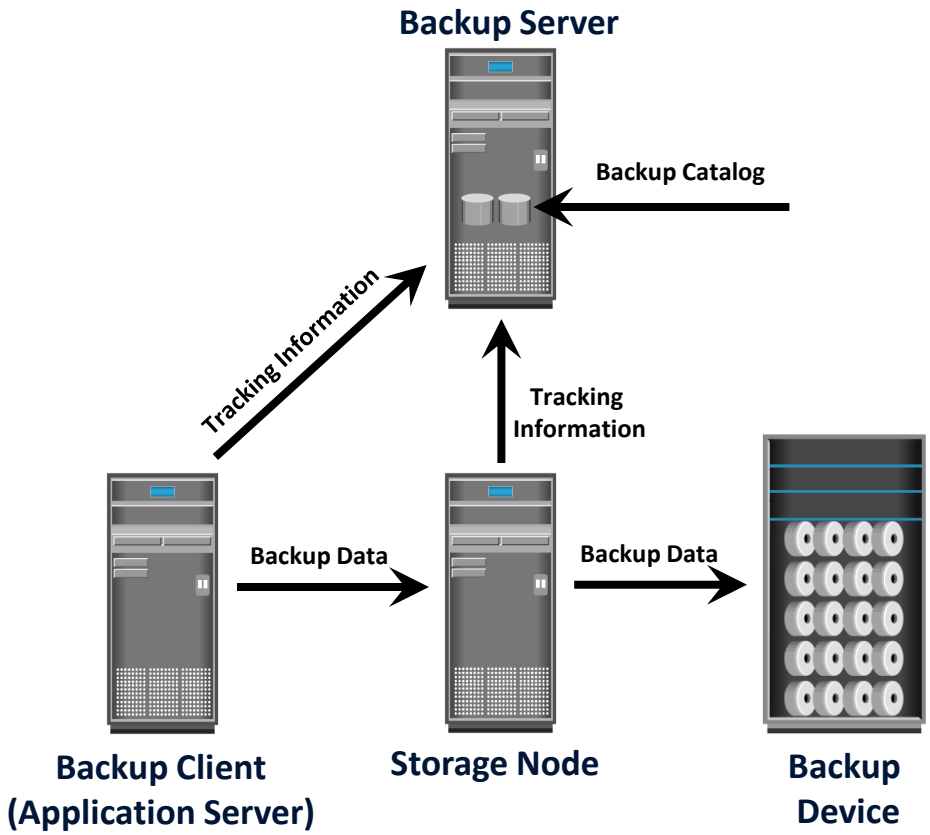
Restoring from Cumulative Backup



- More files to be backed up, therefore, it takes more time to backup and requires more storage space
- Faster restore because only the last full and the last cumulative backup must be applied

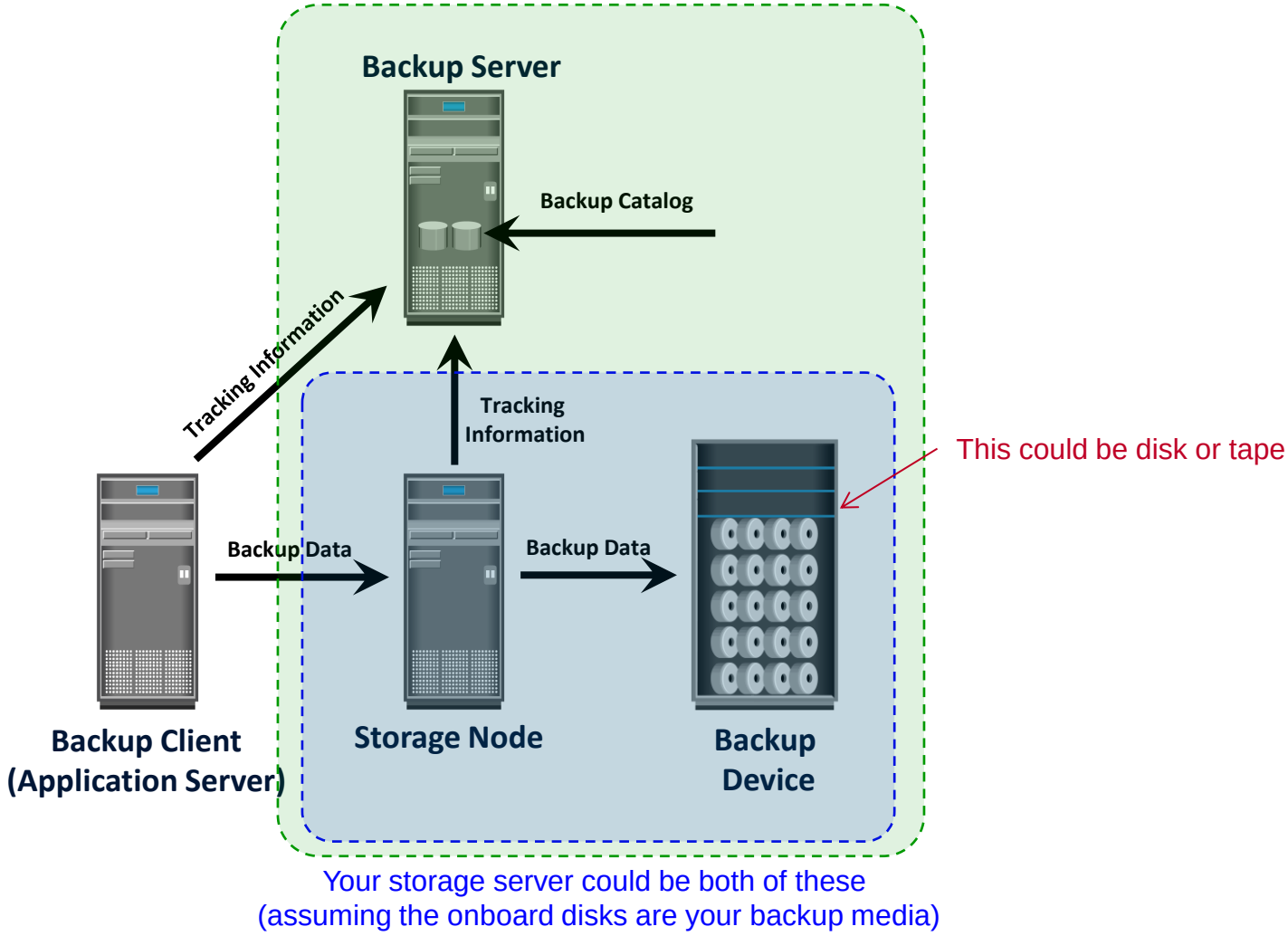
Backup Architecture

- Backup client
  - Gathers the data that is to be backed up and send it to storage node
- Backup server
  - Manages backup operations and maintains backup catalog
- Storage node
  - Responsible for writing data to backup device
  - Manages the backup device



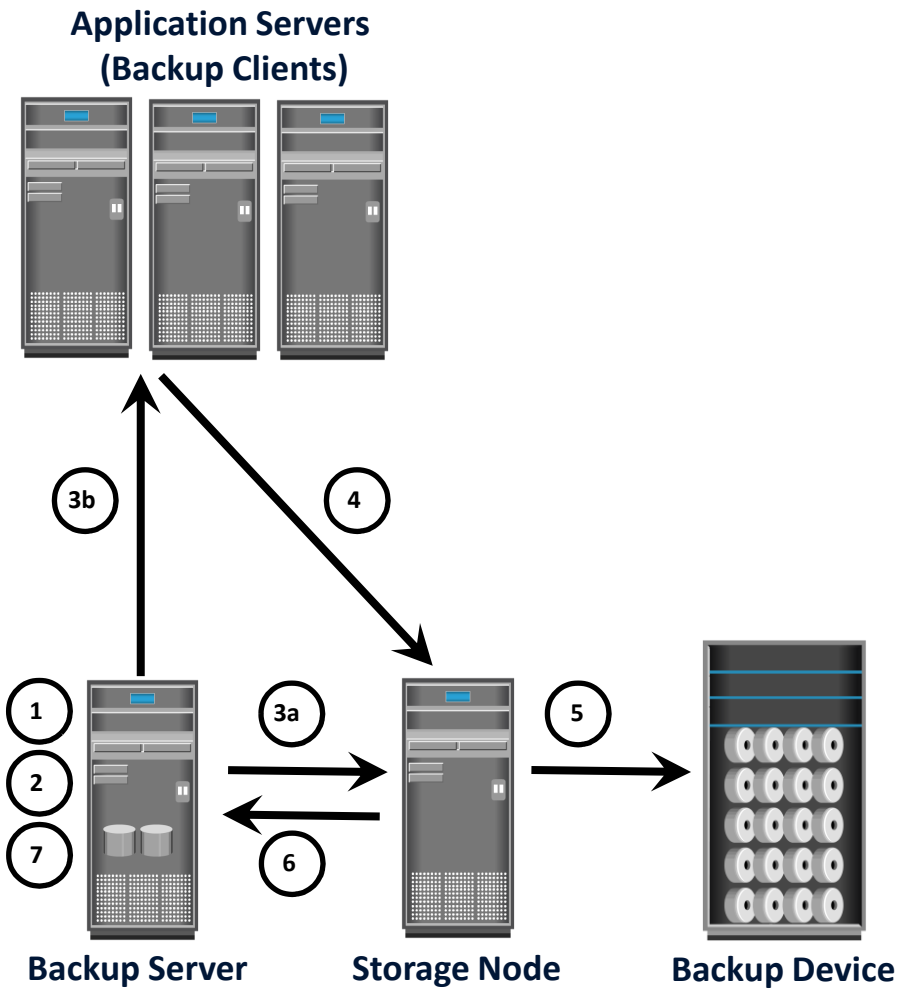
Understanding this traditional model

A storage server could even be all three if index data is just kept with backup data



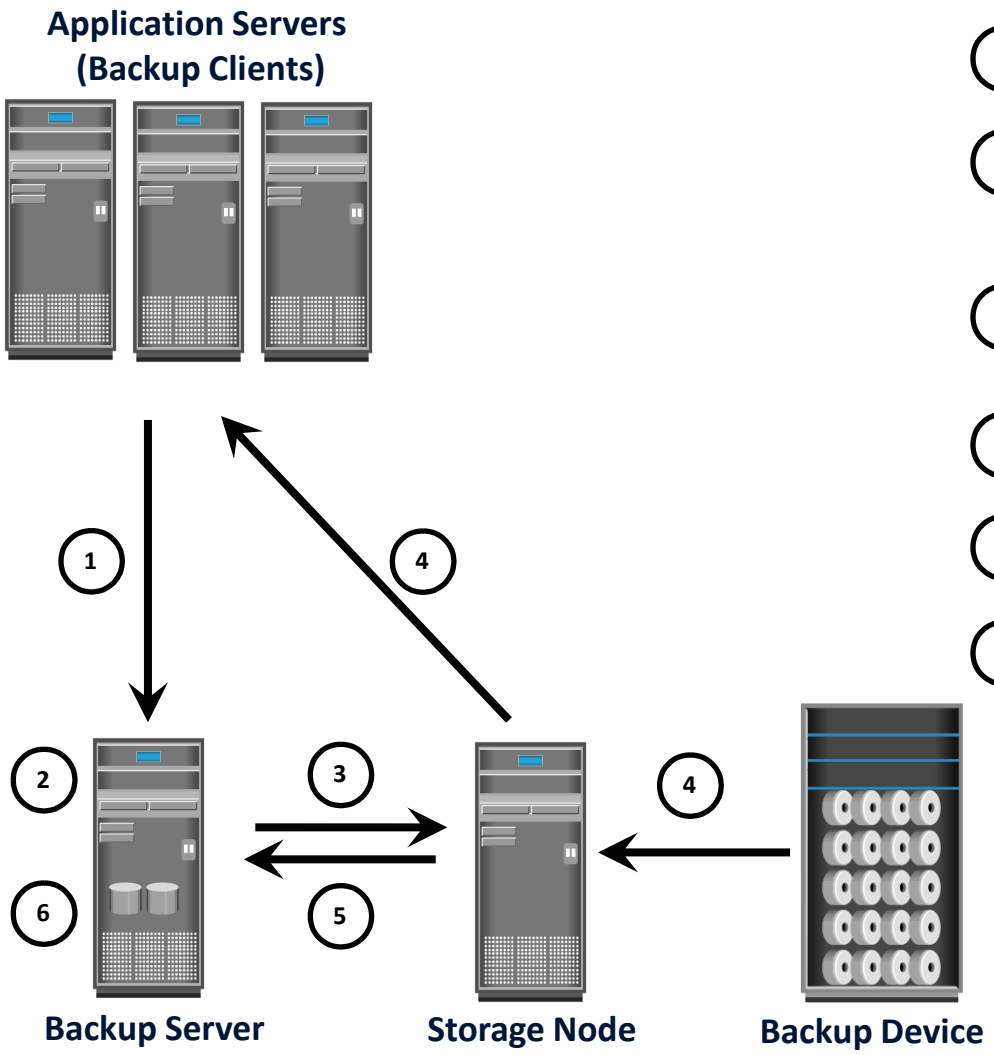
Your storage server could be both of these
(assuming the onboard disks are your backup media)

Backup Operation



1. Backup server initiates scheduled backup process.
2. Backup server retrieves backup-related information from the backup catalog.
3a. Backup server instructs storage node to load backup media in backup device.
3b. Backup server instructs backup clients to send data to be backed up to storage node.
4. Backup clients send data to storage node and update the backup catalog on the backup server.
5. Storage node sends data to backup device.
6. Storage node sends metadata and media information to backup server.
7. Backup server updates the backup catalog.

Recovery Operation



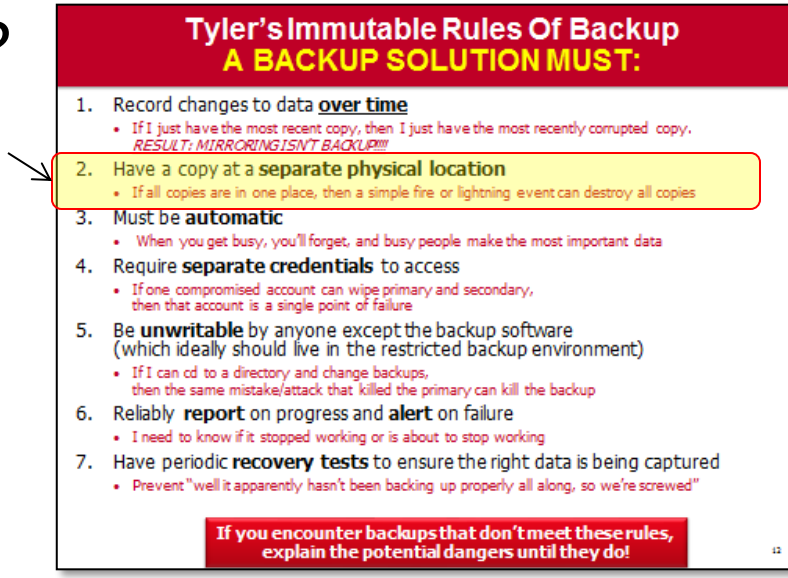
1. Backup client requests backup server for data restore.
2. Backup server scans backup catalog to identify data to be restored and the client that will receive data.
3. Backup server instructs storage node to load backup media in backup device.
4. Data is then read and send to backup client.
5. Storage node sends restore metadata to backup server.
6. Backup server updates the backup catalog.

Backup Methods

- Two methods of backup, based on the state of the application when the backup is performed
  - Hot or Online
    - Application is up and running, with users accessing their data during backup
    - Open file agent can be used to backup open files
  - Cold or Offline
    - Requires application to be shutdown during the backup process
- Bare-metal recovery
  - OS, hardware, and application configurations are appropriately backed up for a full system recovery

Snapshots as backups

- Can we use **snapshots** as backups?
- Well, remember my rules?
  - Not a separate physical location!
  - But, we can do snapshots AND proper remote backups



- Why?
  - Lower RPO: Snapshots are near-instant, so we can take a lot
  - Lower RTO: Restoring snapshots is also near-instant
  - Cheaper: Can take fewer remote backups, since many most mundane recovery scenarios can use snapshots
  - May be sufficient for non-critical workloads where data loss is survivable
- Local snapshots are *useful* but not *sufficient* for business-critical workloads!

Backup consistency

- Assume live ("hot") backup
- Is data crash-consistent, or can we do better?
- **Quiesce**: To make consistent at this time (**quiescent**).
  - Tell the OS that you're about to take a snapshot, request quiescence
  - OS flushes all buffers and commits the journal, pauses all IO, says OK
  - Take snapshot
  - Allow OS to resume
  - Base the backup (which takes longer) off this snapshot
  - Resulting backup is **OS consistent**
- Can also be application-aware
  - Same as above, but you tell the *application* to quiesce
  - Requires backup-aware applications (e.g. Microsoft SQL Server, Oracle database, etc.)
  - Resulting backups are **application consistent**

Lesson 3: Backup Targets

During this lesson the following topics are covered:

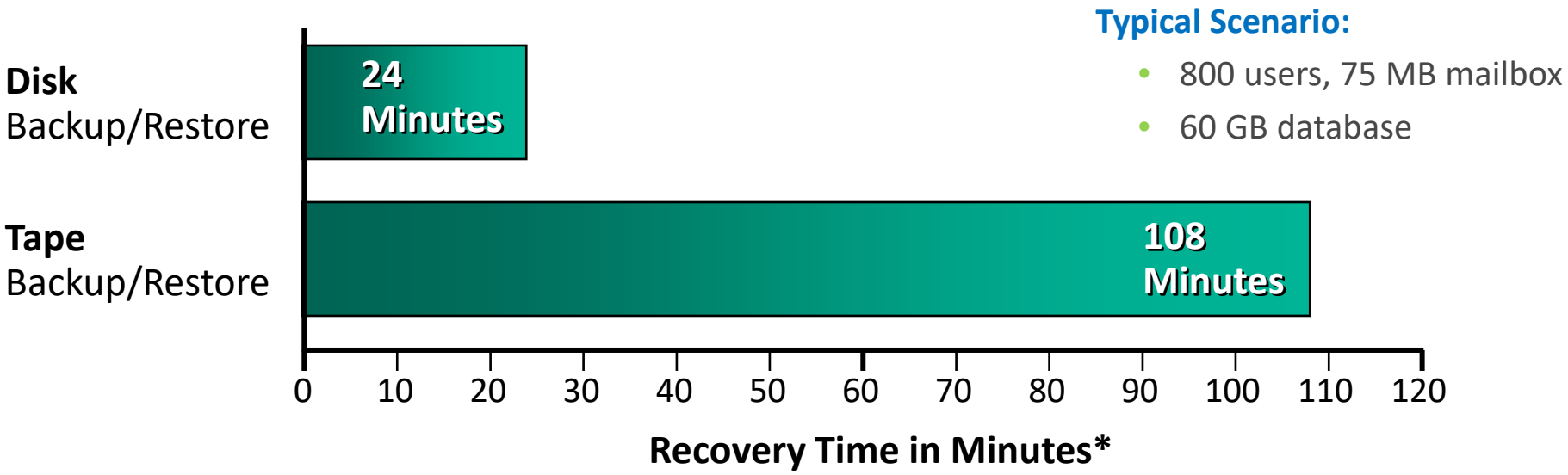
- Backup to Tape
- Backup to Disk
- Backup to Virtual Tape

Backup to Tape

- Traditionally low cost solution
- Tape drives are used to read/write data from/to a tape
- Sequential/linear access
- Multiple streaming to improve media performance
  - Writes data from multiple streams on a single tape
- Limitation of tape
  - Backup and recovery operations are slow due to sequential access
  - Wear and tear of tape
  - Shipping/handling challenges
  - Controlled environment is required for tape storage
  - Causes “shoe shining effect” or “backhitching”

Backup to Disk

- Enhanced overall backup and recovery performance
  - Random access
- More reliable
- Can be accessed by multiple hosts simultaneously



Source: EMC Engineering and EMC IT

Biased

Backup Target Comparison

	Tape	Disk
Offsite Replication Capabilities	No	Yes
Reliability	No inherent protection methods Actually, Yes there are	RAID, spare
Performance	Low	High
Use	Backup only	Multiple (backup and production)

In defense of tape

- These slides omit a key features of tape that's the reason it's still not dead:
  - **Longevity:** You can stick a tape in a vault for 20 years and probably still read it. A tape can't have a head crash, bad bearing, or flaky controller board.
  - **Cost:** Tape *drives* are expensive, but tape *media* is way cheaper per GB than even disk.
    - Current-gen "LTO-8" tapes:
      - Tape capacity: 12TB raw (30TB compressed)
      - Cost of a tape: $50 (so $4.16/TB)
      - Cost of a drive: ~$3k
    - Comparison between LTO-8 and 12TB $200 HDDs*:

Capacity	Cost of drive + tapes	Cost of HDDs
60 TB	3,250	1,000
120 TB	3,500	2,000
180 TB	3,750	3,000
240 TB	4,000	4,000
300 TB	4,250	5,000
360 TB	4,500	6,000
420 TB	4,750	7,000

This article goes into more depth if you're interested

* Seagate 12TB, $200 in Feb 2024

This isn't changing… Here's the same table with 2019 prices – the trend is identical!

Capacity	Cost of drive + tapes	Cost of HDDs
60 TB	3,750	1,500
120 TB	4,500	3,000
180 TB	5,250	4,500
240 TB	6,000	6,000
300 TB	6,750	7,500
360 TB	7,500	9,000
420 TB	8,250	10,500

Special-case backups: Virtual environments

Modern virtual environment note

- In a modern cluster of hypervisors, you don't worry so much about server configuration
- All servers are similar: they're just dumb hosts for the hypervisor
- Virtual machines are the true unit of backup in this case

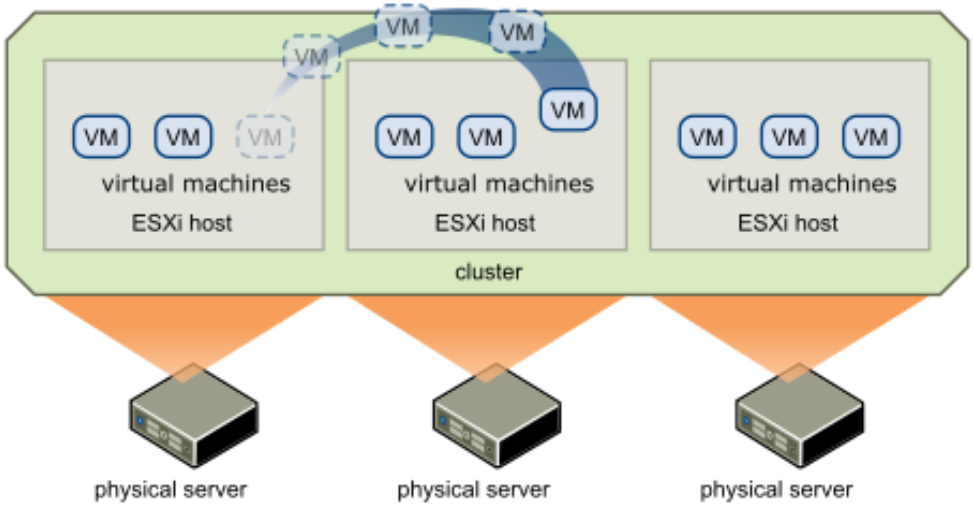
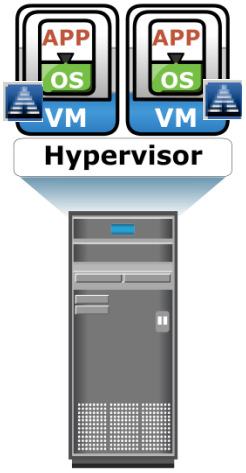


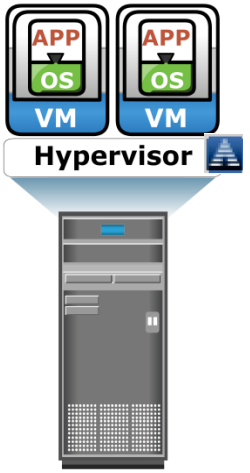
Figure from VMware docs here.

Traditional Backup Approaches

- Backup agent on VM
  - Requires installing a backup agent on each VM running on a hypervisor
  - Can only backup virtual disk data
  - Does not capture VM files such as VM swap file, configuration file
  - Challenge in VM restore
- Backup agent on Hypervisor
  - Requires installing backup agent only on hypervisor
  - Backs up all the VM files



Backup agent runs on each VM

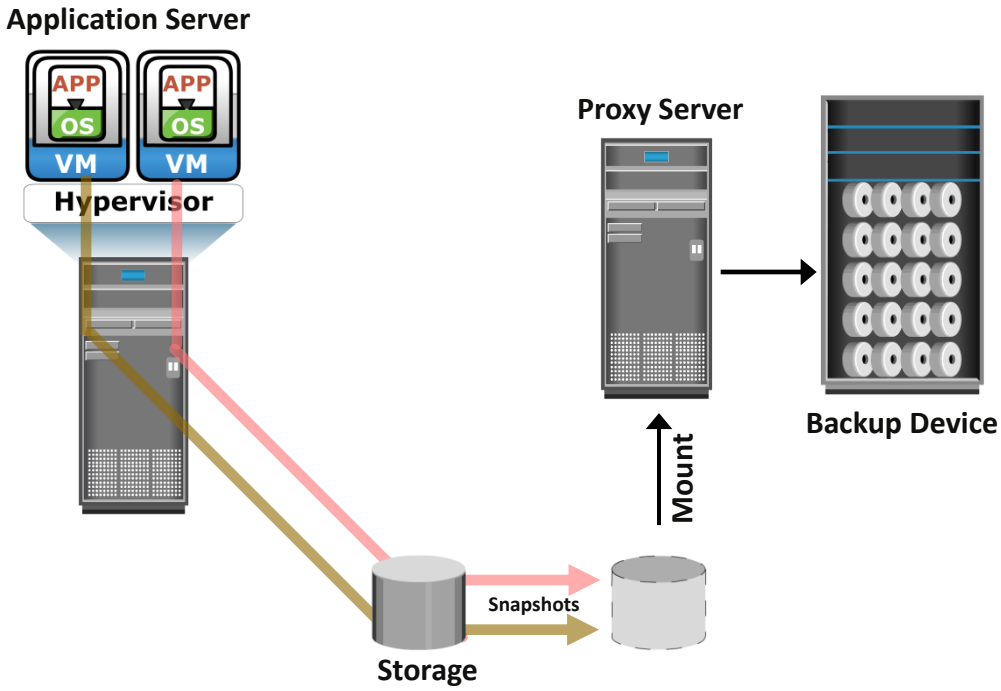


Backup agent runs on Hypervisor

= *Backup Agent*

Image-based Backup

- Creates a copy of the guest OS, its data, VM state, and configurations
  - The backup is saved as a single file – "image"
  - Mounts image on a proxy server
  - Offloads backup processing from the hypervisor
- Enables quick restoration of VM



Special-case backups: Archival

Fixed Content

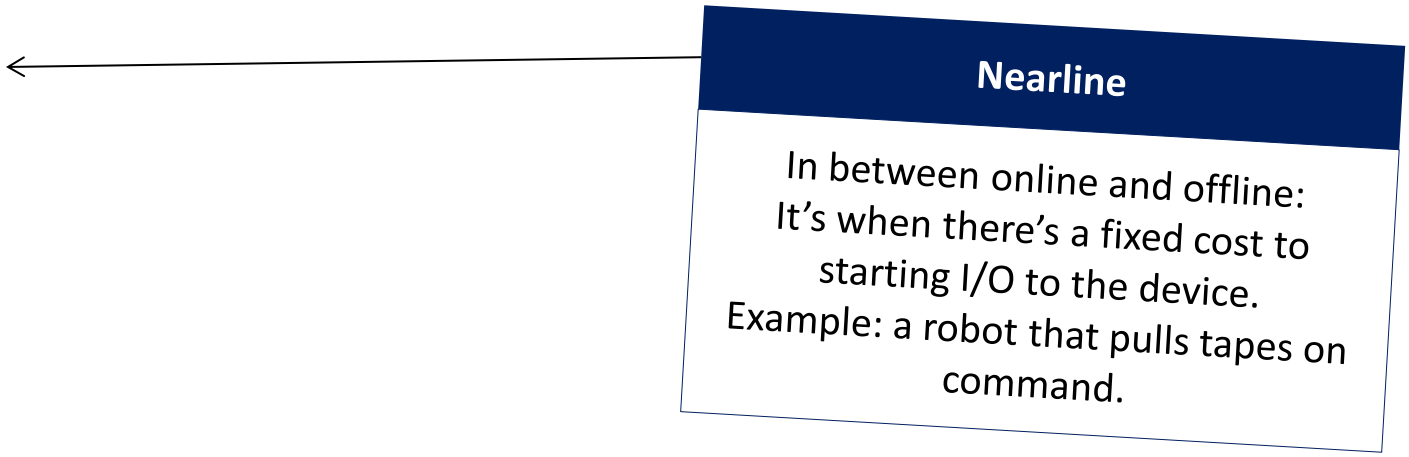
- Fixed content is growing at more than 90% annually
  - Significant amount of newly created information falls into this category
  - New regulations require retention and data protection

Examples of Fixed Content		
Electronic Documents	***Digital Records***	***Rich Media***

• Contracts and claims
• Email attachments
• Financial spread sheets
• CAD/CAM designs
• Presentations
	• Documents
	  • Checks, securities trades
	  • Historical preservation
	• Photographs
	  • Personal/professional
	• Surveys
	  • Seismic, astronomic, geographic
		• Medical
		  • X-rays, MRIs, CT Scan
		• Video
		  • News/media, movies
		  • Security surveillance
		• Audio
		  • Voicemail
		  • Radio

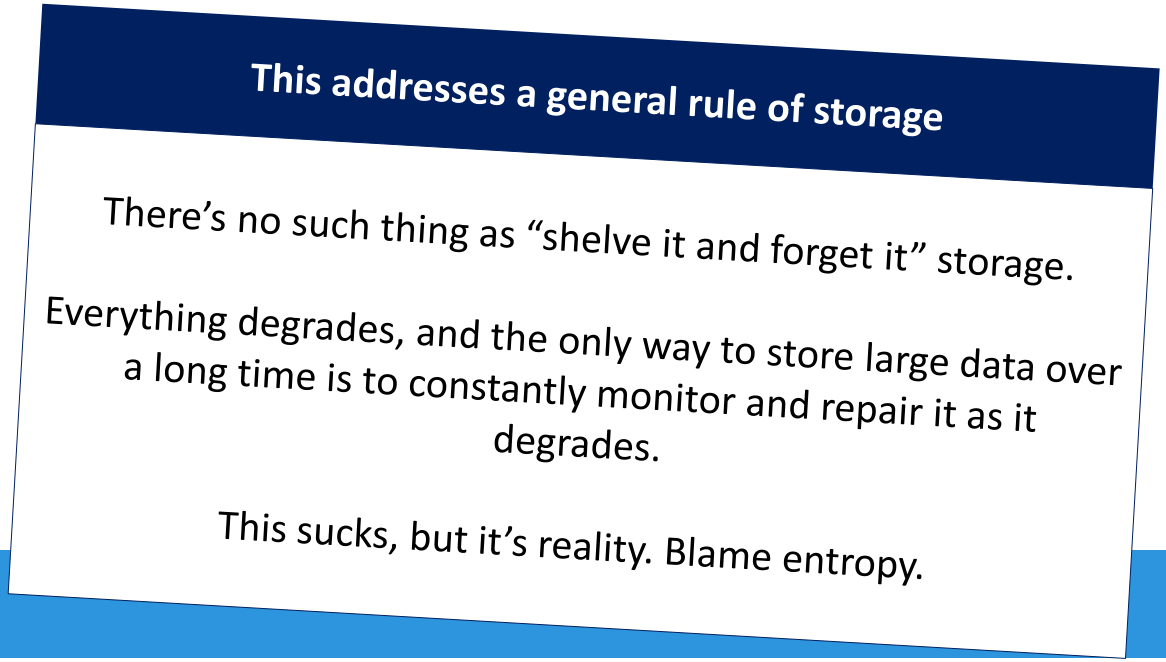
Data Archive

- A repository where fixed content is stored
- Enables organizations retaining their data for an extended period of time in order to
  - Meet regulatory compliance
  - Plan new revenue strategies
- Archive can be implemented as
  - Online
  - Nearline
  - Offline



Challenges of Traditional Archiving Solutions

- Both tape and optical are susceptible to wear and tear
  - Involve operational, management, and maintenance overhead
- Have no intelligence to identify duplicate data
  - Same content could be archived many times
- Inadequate for long-term preservation (years-decades)
- Unable to provide online and fast access to fixed content



Use Case: Email Archiving

- Moves the emails from primary to archive storage, based on policy
- Saves space on primary storage
- Enables to retain emails in the archive for longer period to meet regulatory requirements
- Gives end users virtually unlimited mailbox space
- File archiving is another use case that benefits from an archival solution

Email archiving in practice

Systems like this are deployed in large companies for regulatory compliance and to save on primary storage. In your email app, you'll see something like "this email has been archived, click here to view", and it goes to some janky web app that has the content.

Replication for lower RPO/RTO

Backup vs Replication

Backups

Periodically, make a copy of the data

Replication

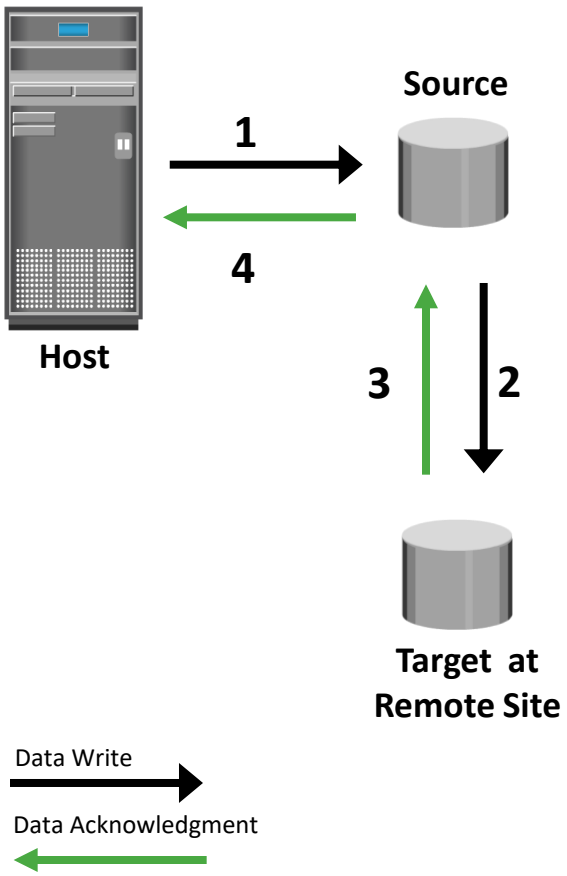
Continuously, keep a copy of the data up-to-date

What is Remote Replication?

- Process of creating replicas at remote sites
  - Addresses risk associated with regionally driven outages
- Modes of remote replication
  - Synchronous
  - Asynchronous

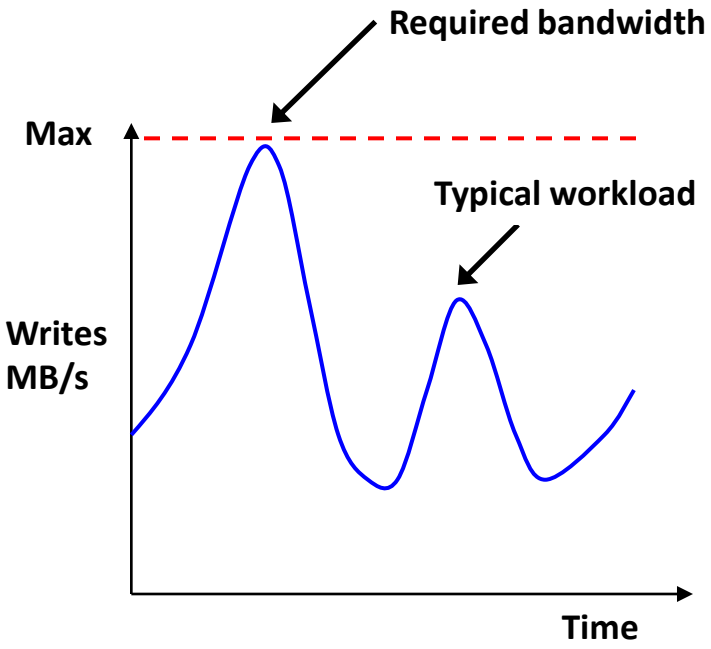
Synchronous Replication – 1

- A write is committed to both source and remote replica before it is acknowledged to the host
- Ensures source and replica have identical data at all times
  - Maintains write ordering
- Provides near-zero RPO



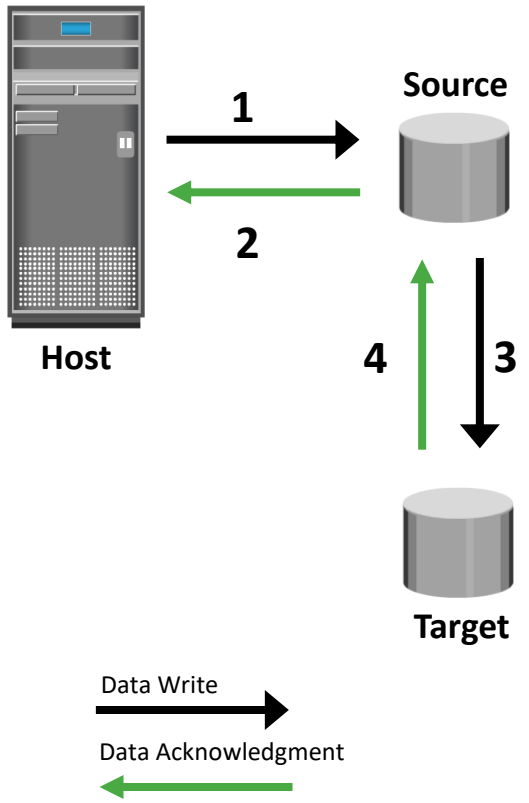
Synchronous Replication – 2

- Response time depends on bandwidth and distance
- Requires bandwidth more than the maximum write workload
- Typically deployed for distance less than 200 km (125 miles) between two sites



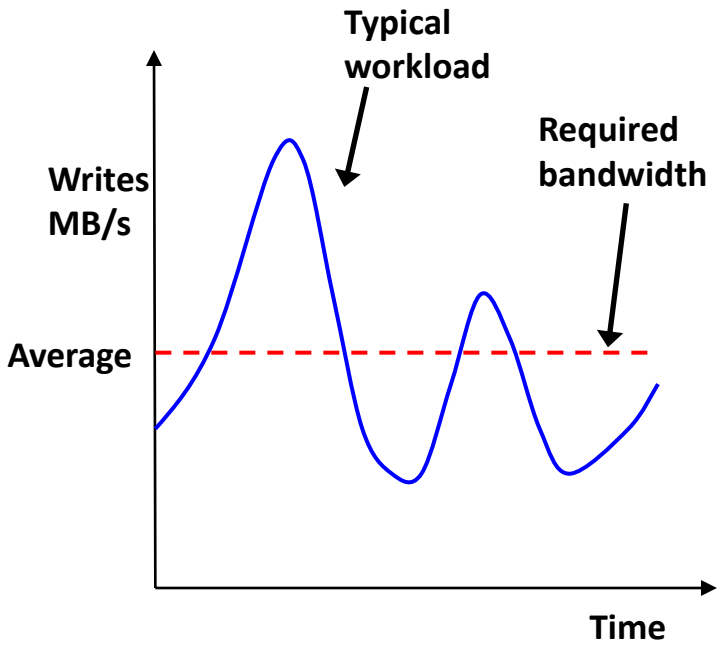
Asynchronous Replication – 1

- A write is committed to the source and immediately acknowledged to the host
- Data is buffered at the source and transmitted to the remote site later
- Finite RPO
  - Replica will be behind the source by a finite amount



Asynchronous Replication – 2

- RPO depends on size of buffer and available network bandwidth
- Requires bandwidth equal to or greater than average write workload
- Sufficient buffer capacity should be provisioned
- Can be deployed over long distances



Host-based Remote Replication

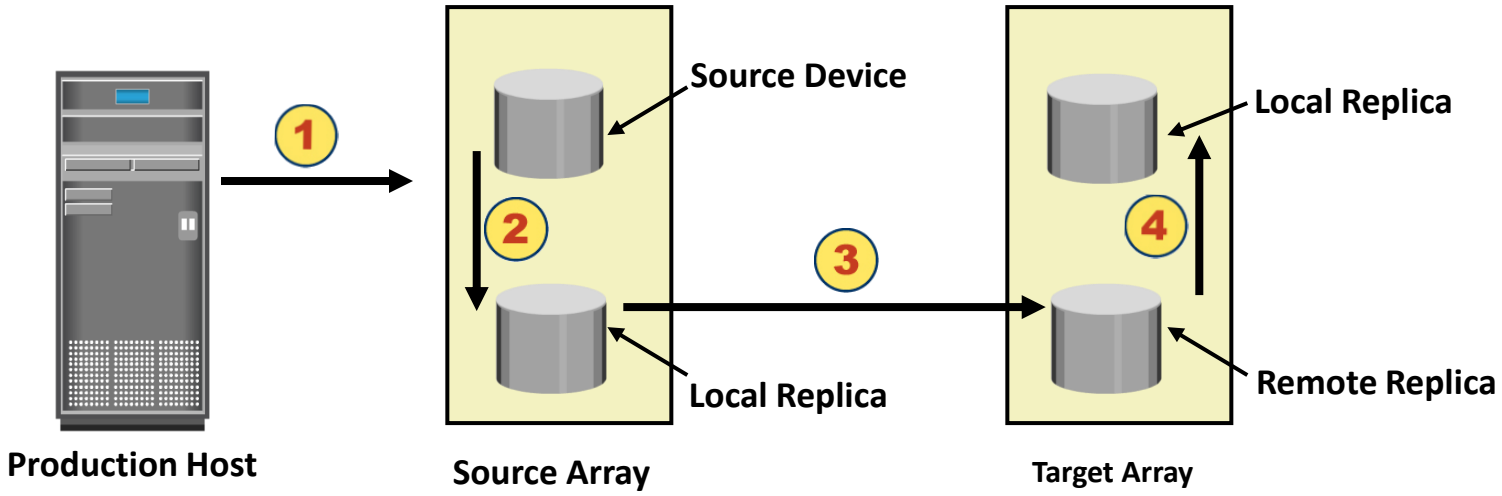
- Replication is performed by host-based software
- LVM-based replication
  - All writes to the source volume group are replicated to the target volume group by the LVM
  - Can be synchronous or asynchronous
- Log shipping
  - Commonly used in a database environment
  - All relevant components of source and target databases are synchronized prior to the start of replication
  - Transactions to source database are captured in logs and periodically transferred to remote host

Storage Array-based Remote Replication – 1

- Replication is performed by array-operating environment
- Three replication methods: synchronous, asynchronous, and disk buffered
- Synchronous
  - Writes are committed to both source and replica before it is acknowledged to host
- Asynchronous
  - Writes are committed to source and immediately acknowledged to host
  - Data is buffered at source and transmitted to remote site later

Storage Array-based Remote Replication – 2

- Disk-buffered



1. Production host writes data to source device.
2. A consistent PIT local replica of the source device is created.
3. Data from local replica is transmitted to the remote replica at target.
4. Optionally a PIT local replica of the remote replica on the target is created.

Summary

- Disaster Recovery (DR) exists to handle cases where High Availability (HA) redundancy is overwhelmed
- For data, the key is backups;
  for compute, it's secondary compute servers
- Backup isn't just mirroring! Rules:
  1. Record changes to data **over time**
  2. Have a copy at a **separate physical location**
  3. Must be **automatic**
  4. Require **separate credentials** to access
  5. Be **unwritable** by anyone except the backup software (which ideally should live in the restricted backup environment)
  6. Reliably **report** on progress and **alert** on failure
  7. Have periodic **recovery tests** to ensure the right data is being captured
- Can do backup locally (for low cost, low RTO/RPO) and/or remotely (true DR, RTO/RPO proportional to cost)
  - Replication can do so continuously