

# ICA0002: IT Infrastructure Services

## Backups

Roman Kuchin  
Juri Hudolejev  
2024

What is a backup?

# Backup

1. A copy of data that can be used to restore the original
2. A process of copying the data for restoration purposes

^^^ *these are fine if you want to explain your non-IT friend what a backups is*

# Backup system as an infrastructure service

Systematic established **process** of

- **Making** a copy of data that can be used to restore the original
- **Storing** that data for required period of time
- **Verifying** the usability of that data for restoration purposes
- **Documenting** the regular data restoration and disaster recovery plans
- **Verifying** the compliance of the documentation and the actual processes

# Backup system as an infrastructure service

In simple words, good backup

- Is created in time
- Is stored as long as needed
- Is available when needed
- Can be used to restore the service to the needed state

And every step of this process is documented and verified

# Purpose of the backup

Ability to restore the **service** back to some certain state

- More than just restoring the data!

Important part of security policies

- Guarantees system availability and usability

# Backup

1. Coverage
2. Frequency
3. Retention
4. Usability
5. Storage
6. Security
7. Documentation

# Coverage: what to backup?

Ideally, everything

In real world, as much as your backup system can handle

Better to have multiple copies of the same data than no copies at all



# Coverage: what to backup?

## Examples:

- Customer created data: Always backup
- Code repositories: Always backup
- Documentation: Backup unless it is in the code repositories
- Secrets: Case by case
- Service configuration: Case by case
- Logs and audit records: Case by case
- Server configuration: Rather do not backup


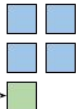
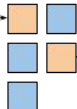

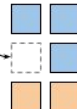


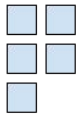
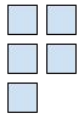

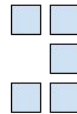







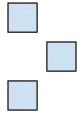


# Data level backups

Export data from the running service

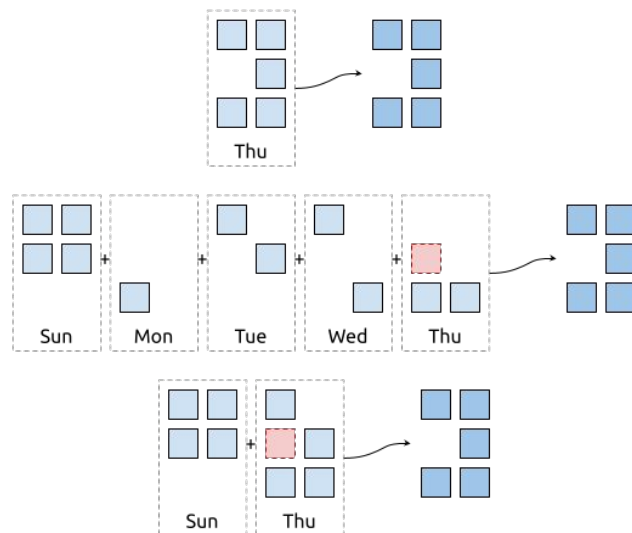
- Usually universal format, can be restored to a different system
- Storage can be optimized (incremental / differential / deduplication)

Examples: SQL dump, JSON export, XML export, ...

# Incremental vs. differential backups

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday
Actual data		Added 	Changed 		Deleted 	
Full backups daily						
Full backups on Sundays <b>Incremental</b> backups on other days (difference since last backup)						
Full backup on Sundays <b>Differential</b> backups on other days (difference since last <b>full</b> backup)						

Backup restore



# Data level backups

Export data from the running service

- Usually universal format, can be restored to a different system
- Storage can be optimized (incremental / differential / deduplication)
- Creation and restore can be time consuming

Portability over efficiency

Examples: SQL dump, JSON export, XML export, ...

# File level backups

Why export data if we can backup the original file?

- Storage can still be optimized
- Creation and restore might require less time
- System being restored must support the same file format

Efficiency over portability -- although win in efficiency is often minimal...

Example: cannot restore MySQL 5.6 files backup to MySQL 5.7 server

# Virtual machine and/or filesystem snapshots

Data, files... Why not just backup entire disk?

- For virtual machines -- creation and restore might require even less time
- May require machine or service downtime to create a backup
- Higher storage and bandwidth requirements

Example: your own machine (laptop, VirtualBox) snapshots

# OS image backups

OS image backups are often overlooked:

- Why would you need them?
- The image can always be downloaded from the internet, right?

What if

- Public image is not available?
- You use custom modified images?

Questions?



Frequency: how often to backup?

# Frequency: how often to backup?

Examples: every night, every Sunday

More often:

- More backup storage is required
- More resources (CPU, memory) are needed to create and verify the backups
- Less service availability (downtimes or reduced HA during backup creation)

Less often:

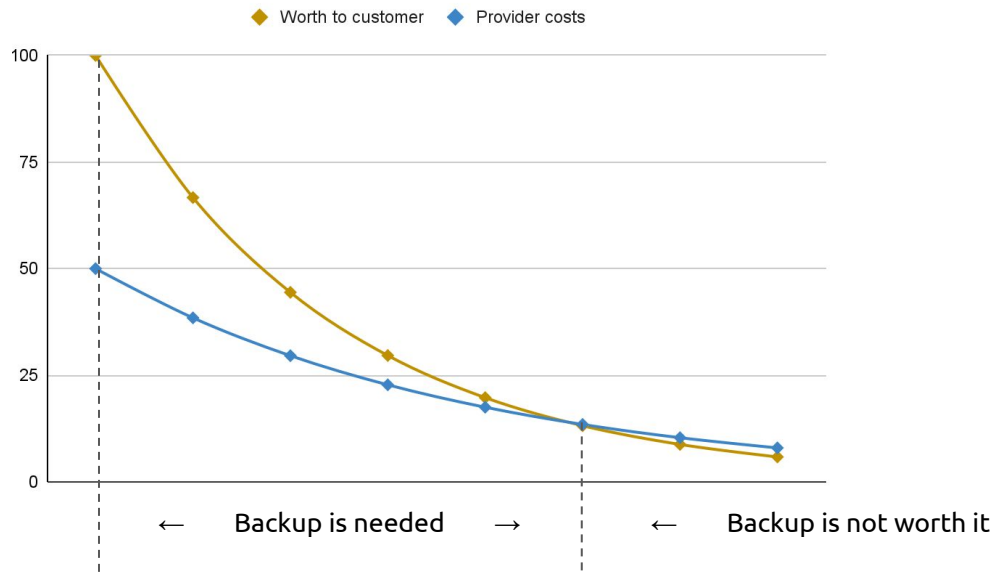
- RPO (recovery point objective): interval between two consecutive backups
- RPO == "acceptable data loss"

Retention: how long to store the backup?

# Retention: how long to store the backup?

Backup storage has its limits

Data value reduces over time



# Retention: how long to store the backup?

Backup storage has its limits

Data value reduces over time

*Backup retention must be longer than RPO!*

*-- Captain Obvious*



# Retention: how long to store the backup?

Use clearly and precisely defined periods

Examples:

- 30 days
- 13 weeks

Number of days and weeks varies in months and years:

- 3 months -- 89, 90, 91 or 92 days?
- 2 years -- 104, 105 or 106 weeks?

# How many versions of the backup are stored?

Ideally, as many versions as possible

In real world:

$$\text{version\_count} = \text{frequency} * \text{retention}$$

Note: "N versions" means "N recovery points", **not** "N copies"!

# Usability: is the backup usable?

Ensure that you can restore the service from the backup

- Was the backup created **in time**?
- Is the stored backup **readable**?
- Is the stored backup **enough** to restore the service?
- Could the service be restored **to the needed state**?

Solution: test backup restores regularly

- Use backup data (not live systems) for reporting and all sorts of data mining
- Do **regular** recovery exercises, automate them as much as possible



Usability is the most important  
feature of the backup

Questions?

# Storage: where to store backups?

3-2-1 rule:

At least 3 copies of the data -- 2 storage types onsite -- 1 copy offsite

**Separate** cloud or physical location -- preferably both!

Monitoring physical conditions: temperature, humidity etc.

# Storage: where **not** to store backups?

*"We have redundant hardware (RAID), so the backups are automated!"*

*"We are keeping backup on the same server to save costs!"*

*"We are keeping backup in the same datacenter, so the restore is really fast!"*

*"We are using cloud platforms, they do the backups themselves!"*

What is wrong with these approaches? What are the risks?

# Security: how to protect backups?

Good backup contains all the data needed to restore the system.

Backup should be protected as severely as the live system itself.

Store the backup in the safe place:

- Physical copies -- vault
- Online copies -- encrypt the transport, storage -- whatever possible

# Documentation: how to describe backups?

Three main documents:

1. Backup SLA (service level agreement)
2. Backup restore plan
3. Technical documentation for backup operators

Related documents:

- DR (disaster recovery) plan

# Documentation: how to describe backups?

## Backup SLA:

1. Coverage -- what is being backed up
2. Schedule -- how often backups are made; what is RPO  
(recovery **point** objective = acceptable data loss)
3. Storage -- how and where are backups stored
4. Retention -- how many versions of the backups (recovery points) are stored and for how long
5. Usability checks -- how and how often is the backup usability verified
6. Restore process -- how to decide when to restore from the backup; what is the RTO (recovery **time** objective)

# RPO vs. RTO

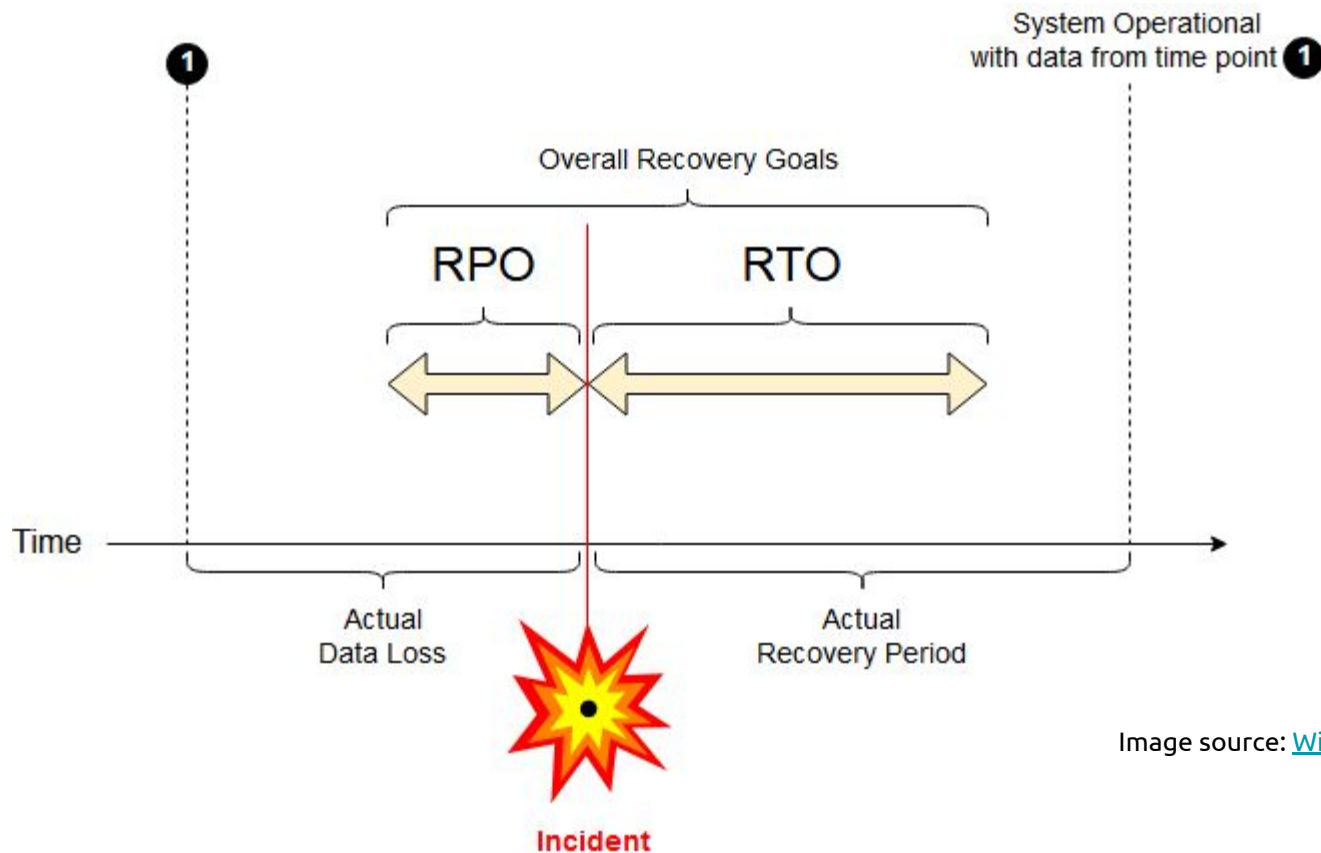


Image source: [Wikipedia](#) | [CC BY 4.0](#)



# RPO

Example: backup is created every night at 01:00.

What is the RPO?

# RPO

Example: backup is created every night at 01:00.

**Note: backup creation and upload also take some time!**

- At 01:00 the next backup starts
- It takes  $T_D$  to extract the data
- It takes  $T_U$  to upload the data to the backup server
- It takes  $T_S$  to sync the backup server with offsite storage

Honest RPO is **24 hours +  $T_D$  +  $T_U$  +  $T_S$**  (worst case scenario)

# RPO

Example: backup is created every night at 01:00.

**Note: some time zones have daylight saving adjustments!**

- Consider daylight saving time in your RPO, or
- (recommended) use UTC zone for backup schedules

Example: backup is created every night at 01:00 **UTC**.

# Documentation: how to describe backups?

Backup restore plan:

- Documented process of the backup restoration
- Explains in details who does what, how and when

Targets engineers who may be not familiar with your backup system

- Ideally any employee with needed permissions could to restore the service

# Documentation: how to describe backups?

Technical documentation for operators:

- How to check if the backup was created
- How to verify the backup
- How to install, configure and administer the backup system

# Backup is your last resort!

Build your systems so that you would not need backups:

- High availability and redundancy
- Multiple copies of the data

**But still do backups properly!**

Never underestimate the importance of the backups

Questions?

What is **not** a backup?



# Backup vs. archive

## Backup:

- Main focus: fast service restore to a certain state
- Retention period: weeks, sometimes months

## Archive:

- Main focus: data consistency + low cost of ownership
- Retention period: years, sometimes decades
- Ideally retyped to a new media from time to time

# Backup vs. configuration management

## Configuration management

- Helps to restore the service faster
- Does **not** replace the backup!

Goal: restore the service **configuration**, not the data

Some components are easier to recreate than to recover

Better to have a backup unused than not have it all

*Only wimps use tape backup; real men just upload their  
important stuff on ftp, and let the rest of the world mirror it ;)*

*-- Linus Torvalds*

