200 XP

Design for availability and recoverability

10 minutes

Clinicians at Lamna Healthcare have little tolerance for downtime. They need to have access to clinical IT systems around the clock to ensure they're providing the highest quality care at all times. To meet the around-the-clock demands of clinicians, Lamna Healthcare's applications must be able to handle failures with minimal impact to their users. How do they ensure their applications remain operational, both for localized incidents and large-scale disasters?

Here, we'll talk through how to include availability and recoverability in your architecture design.

Availability and recoverability

In a complex application, any number of things can go wrong at any scale. Individual servers and hard drives can fail. A deployment issue could unintentionally drop all tables in a database. Whole datacenters may become unreachable. A ransomware incident could maliciously encrypt all your data.

Designing for *availability* focuses on maintaining uptime through small-scale incidents and temporary conditions like partial network outages. You can ensure your application can handle localized failures by integrating high availability into each component of an application and eliminating single points of failure. Such a design also minimizes the impact of infrastructure maintenance. High-availability designs typically aim to eliminate the impact of incidents quickly and automatically and ensure that the system can continue to process requests with little to no impact.

Designing for *recoverability* focuses on recovery from data loss and from larger scale disasters. Recovery from these types of incidents often involves active intervention, though automated recovery steps can reduce the time needed to recover. These types of incidents may result in some amount of downtime or permanently lost data. Disaster recovery is as much about careful planning as it is about execution.

Including high availability and recoverability in the design of your architecture protects your business from financial losses resulting from downtime and lost data. They ensure your

business from financial losses resulting from downtime and lost data. They ensure your reputation isn't negatively impacted by a loss of trust from your customers.

Architecting for availability and recoverability

Architecting for availability and recoverability ensures that your application can meet the commitments you make to your customers.

For availability, identify the service-level agreement (SLA) you're committing to. Examine the potential high-availability capabilities of your application relative to your SLA, and identify where you have proper coverage and where you'll need to make improvements. The goal is to add redundancy to components of the architecture so that you are less likely to experience an outage. Examples of high-availability design components include clustering and load balancing. Clustering replaces a single VM with a set of coordinated VMs. When one VM fails or becomes unreachable, services can fail over to another one that can service the requests. Load balancing spreads requests across many instances of a service, detecting failed instances and preventing requests from being routed to them.

For recoverability, perform an analysis that examines possible data loss and major downtime scenarios. The analysis should include an exploration of recovery strategies and the cost-benefit tradeoff for each. This exercise will give you important insight into your organization's priorities and help clarify the role of your application. The results should include the application's recovery point objective (RPO) and recovery time objective (RTO).

- **Recovery Point Objective**: The maximum duration of acceptable data loss. RPO is measured in units of time, not volume: "30 minutes of data", "four hours of data", and so on. RPO is about limiting and recovering from data *loss*, not data *theft*.
- **Recovery Time Objective**: The maximum duration of acceptable downtime, where "downtime" needs to be defined by your specification. For example, if the acceptable downtime duration is eight hours in the event of a disaster, then your RTO is eight hours.

With RPO and RTO defined, you can design backup, restore, replication, and recovery capabilities into your architecture to meet these objectives.

Every cloud provider offers a suite of services and features you can use to improve your application's availability and recoverability. When possible, use existing services and best practices, and try to resist creating your own.

Hard drives can fail, datacenters can become unreachable, and hackers can attack. It's important that you maintain a good reputation with your customers using availability and recoverability. Availability focuses on maintaining uptime through conditions like network outages, and recoverability focuses on retrieving data after a disaster.

Check your knowledge

Check your knowledge

1. Suppose you would like to increase the availability of your system to provide a better service-level agreement (SLA) to your customers. Which of the following is a guiding principle you can use?

- ○ Reduce your target for maximum duration of acceptable data loss.
- ○ Encrypt all data at rest
- ○ Eliminate single point of failure

2. Which of the following would be impacted by your defined Recovery Point Objective (RPO)?

- ○ The frequency of database backups
- ○ The number of regions that data is replicated to
- ○ The number of instances in a database cluster
- ○ The type of load-balancing technology used in your application

Check your answers