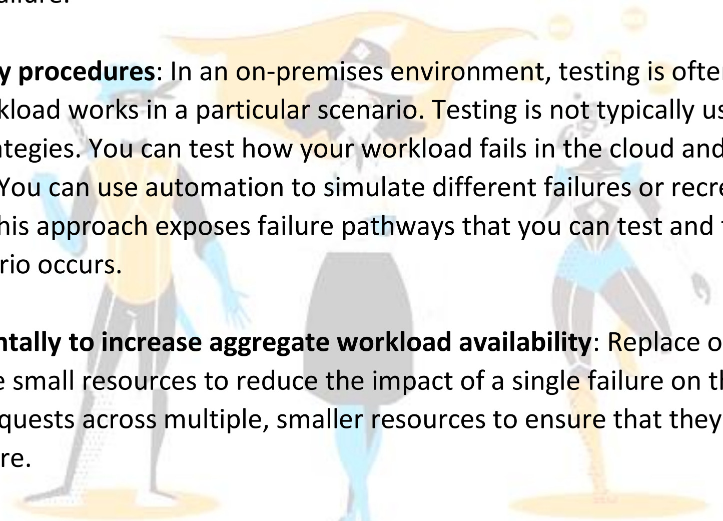
- ## Phase 2 - Reliability

Reliability Design Principles

There are five design principles for reliability in the cloud:

- **Automatically recover from failure**: You can trigger automation when a threshold is breached by monitoring a workload for key performance indicators (KPIs). These KPIs should be a measure of business value, not the technical aspects of the operation of the service. This allows automatic notification and tracking of failures and automated recovery processes that work around the failure.



- **Test recovery procedures**: In an on-premises environment, testing is often conducted to prove that the workload works in a particular scenario. Testing is not typically used to validate recovery strategies. You can test how your workload fails in the cloud and validate your recovery procedures. You can use automation to simulate different failures or recreate scenarios that led to failures. This approach exposes failure pathways that you can test and fix before an actual failure scenario occurs.

- **Scale horizontally to increase aggregate workload availability**: Replace one significant resource with multiple small resources to reduce the impact of a single failure on the overall workload. Distribute requests across multiple, smaller resources to ensure that they don't share a common point of failure.

- **Stop guessing capacity**: A common cause of failure in on-premises workloads is resource saturation when the demands placed on a workload exceed the capacity of that workload (this is often the objective of denial of service attacks). In the cloud, you can monitor demand and workload utilization, and automate the addition or removal of resources to maintain the optimal level to satisfy demand without over-or under-provisioning. There are still limits, but some quotas can be controlled, and others can be managed (see Manage Service Quotas and Constraints).

- **Manage change in automation**: Changes to your infrastructure should be made using automation. The changes that need to be managed include changes to the automation, which then can be tracked and reviewed.

Reliability Evaluation Questions

1. How do you manage service quotas or constraints?
2. How do you plan your network topology?

3. How do you design your workload service architecture?
4. How do you design interactions in a distributed system to prevent failure?
5. How do you design interactions in a distributed system to mitigate or withstand failures?
6. How do you monitor workload progress?
7. How do you design your workload to adapt to changes in demand?
8. How do you implement change?
9. How do you backup data?
10. How do you use fault isolation to protect your workload?
11. How do you test reliability?
12. How do you plan for disaster recovery?

Apply these to your design. Update the design. Commit the design to the repository.