General Design Principles

The Well Architected Framework identifies a set of general design principles to facilitate good design in the cloud:

Stop guessing your capacity needs: Eliminate guessing about your infrastructure capacity needs. When you make a capacity decision before you deploy a system, you might end up sitting on expensive idle resources or dealing with the performance implications of limited capacity. With cloud computing, these problems can go away. You can use as much or as little capacity as you need, and scale up and down automatically.

Test systems at production scale: In the cloud, you can create a production-scale test environment on demand, complete your testing, and then decommission the resources. Because you only pay for the test environment when it's running, you can simulate your live environment for a fraction of the cost of testing on premises.

Automate to make architectural experimentation easier: Automation allows you to create and replicate your systems at low cost and avoid the expense of manual effort. You can track changes to your automation, audit the impact, and revert to previous parameters when necessary

Allow for evolutionary architectures: In a traditional environment, architectural decisions are often implemented as static, one-time events, with a few major versions of a system during its lifetime.

As a business and its context continue to change, these initial decisions might hinder the system's ability to deliver changing business requirements. In the cloud, the capability to automate and test on demand lowers the risk of impact from design changes. This allows systems to evolve over time so that businesses can take advantage of innovations as a standard practice.

Drive architectures using data: In the cloud, you can collect data on how your architectural choices affect the behavior of your workload. This lets you make fact-based decisions on how to improve your workload. Your cloud infrastructure is code, so you can use that data to inform your architecture choices and improvements over time.

Improve through game days: Test how your architecture and processes perform by regularly scheduling game days to simulate events in production. This will help you understand where improvements can be made.

The Five Pillars of the Well Architected Framework. 1. Creating a software system is a lot like constructing a building.

1. If the foundation is not solid structural problems can undermine the integrity and function of the building.

2. When architecting technology solutions, if you neglect the five pillars of **operational excellence, security, reliability, performance efficiency, and cost optimization** it can become challenging to build a system that delivers on your expectations and requirements.

3. Incorporating these pillars into your architecture will help you produce stable and efficient systems.

4. This will allow you to focus on the other aspects of design, such as functional requirements.