

Performance Efficiency on the Cloud

Use computing resources efficiently to meet system requirements and to maintain that efficiency as demand changes and technologies evolve.

Performance Efficiency on the Cloud

The performance efficiency pillar includes the ability to use computing resources efficiently to meet system requirements and to maintain that efficiency as demand changes and technologies evolve. Design Principles

There are five design principles for performance efficiency in the cloud:

Democratize advanced technologies:

Technologies that are difficult to implement can become easier to consume by pushing that knowledge and complexity into the cloud vendor's domain.

Rather than having your IT team learn how to host and run a new technology, they can simply consume it as a service. For example, NoSQL databases, media transcoding, and machine learning are all technologies that require expertise that is not evenly dispersed across the technical community. In the cloud, these technologies become services that your team can consume while focusing on product development rather than resource provisioning and management.

Go global in minutes:

Easily deploy your system in multiple Regions around the world with just a few clicks. This allows you to provide lower latency and a better experience for your customers at minimal cost.

Use serverless architectures:

In the cloud, serverless architectures remove the need for you to run and maintain servers to carry out traditional compute activities. For example, storage services can act as static websites, removing the need for web servers, and event services can host your code for you. This not only removes the operational burden of managing these servers, but also can lower

transactional costs because these managed services operate at cloud scale.

Experiment more often:

With virtual and automatable resources, you can quickly carry out comparative testing using different types of instances, storage, or configurations.

Mechanical sympathy:

Use the technology approach that aligns best to what you are trying to achieve. For example, consider data access patterns when selecting database or storage approaches. Take a data-driven approach to selecting a high-performance architecture. Gather data on all aspects of the architecture, from the high-level design to the selection and configuration of resource types. By reviewing your choices on a cyclical basis, you will ensure that you are taking advantage of the continually evolving Cloud providers. Monitoring will ensure that you are aware of any deviance from expected performance and can take action on it.

Finally, your architecture can make tradeoffs to improve performance, such as using compression or caching, or relaxing consistency requirements.