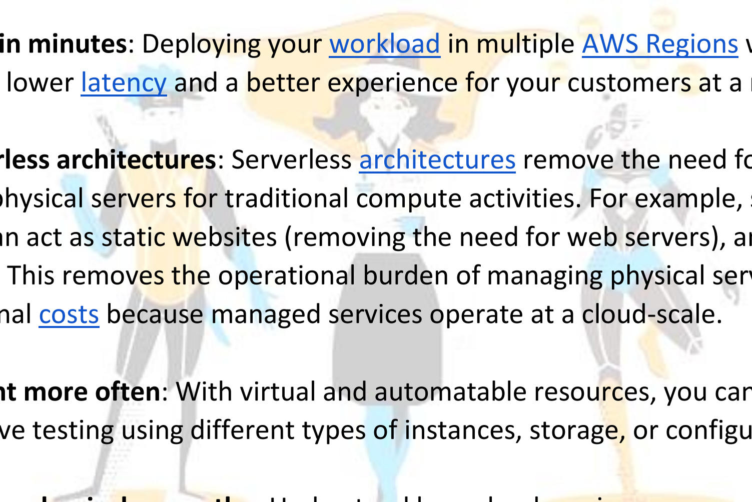
- ## Phase 3 - Performance Efficiency

Performance Efficiency Design Principles

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

- **Democratize advanced technologies**: Make advanced technology implementation easier for your team by delegating complex tasks to your cloud vendor. Consider consuming the technology as a service rather than asking your IT team to learn about hosting and running a new technology. For example, NoSQL databases, media transcoding, and machine learning are all technologies that require specialized expertise. These technologies become services that your team can consume in the cloud, allowing your team to focus on product development rather than resource provisioning and management.



- **Go global in minutes**: Deploying your workload in multiple AWS Regions worldwide allows you to provide lower latency and a better experience for your customers at a minimal cost.

- **Use serverless architectures**: Serverless architectures remove the need for you to run and maintain physical servers for traditional compute activities. For example, serverless storage services can act as static websites (removing the need for web servers), and event services can host code. This removes the operational burden of managing physical servers and can lower transactional costs because managed services operate at a 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.

- **Consider mechanical sympathy**: Understand how cloud services are consumed and always use the technology approach that aligns best with your workload goals. For example, consider data access patterns when selecting database or storage approaches.

Performance Efficiency Evaluation Questions

1. How do you select the best-performing architecture?
2. How do you select the best compute solution?
3. How do you select your storage solution?
4. How do you select your database solution?
5. How do you configure your networking solution?
6. How do you evolve your workload to take care of new releases?
7. How do you monitor your resources to ensure they are performing?
8. How do you use tradeoffs to improve performance?

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