**Pillar three: Performance Efficiency**

**What is it?**

* The Performance Efficiency Pillar Focuses on how to use computing resources efficiently to meet your requirements and how to maintain that efficiency as demand changes and technology evolves.

**Design Principles:**

* Democratize advanced technologies
* Go global in minutes
* Use serverless architectures
* Experiment more often

**Definition:**

* Performance efficiency in the cloud consists of 4 areas;
  + Compute:
    - When architecting your system, it is important to choose the right kind of server. Some applications require heavy CPU utilization, some require heavy memory utilization, etc.
    - With AWS, servers are virtualized and at the click of a button (or API call) you can change the type of server in which your environment is running on. You can even switch to running with no servers at all and use AWS Lambda.
    - At A Cloud Guru, this is how they have designed their environment. If you are watching the video for this lecture on their platform, you are interacting with Lambda in a serverless fashion.
    - Compute Questions:
      * How do you select the appropriate instance type for your system?
      * How do you ensure that you continue to have the most appropriate instance type as new instance types and features are introduced?
      * How do you monitor your instances post launch to ensure they are performing as expected?
      * How do you ensure that the quantity of your instances matches demand?
  + Storage:
    - The optimal storage solutions for your environment depends on a number of factors. For example;
      * Access method – block, File or Object.
      * Patterns of access – random or sequential.
      * Throughput required.
      * Frequency of access – Online, Offline or Archival.
      * Frequency of update – Worm, Dynamic.
      * Availability constraints.
      * Durability constraints.
    - At AWS, the storage is virtualized. With S3 you can have 11 x 9’s durability, cross region replication, etc. With EBS you can choose between different storage mediums (such as SSD, Magnetic, PIOPS, etc.). You can also easily move volumes between the different types of storage mediums.
    - Storage Questions:
      * How do you select the appropriate storage solution for your system?
      * How do you ensure that you continue to have the most appropriate storage solution as new storage solutions and features are launched?
      * How do you monitor your storage solution to ensure it is performing as expected?
      * How do you ensure that the capacity and throughput of your storage solutions matches demand?
  + Database:
    - The optimal database solution depends on a number of factors. Do you need database consistency, do you need high availability, do you need noSQL, do you need DR, etc?
    - With AWS, you get a lot of options; RDS, DynamoDB, Redshift, etc.
    - Database Questions:
      * How do you select the most appropriate database solution for your system?
      * How do you ensure that you continue to have the most appropriate database solution and features as new database solution and features are launched?
      * How do you monitor your databases to ensure performance is as expected?
      * How do you ensure the capacity and throughput of your databases matches demand?
  + Space-time trade-off:
    - When you are building out your infrastructure there is a trade-off where space (think memory or storage) is used to reduce processing time (think compute) or time is used to reduce space. You can also cache data to reduce time (like with CloudFront or ElastiCache).
    - Using AWS, you can use services such as RDS to add read replicas, reducing the load on your database and creating multiple copies of the database. This helps to lower latency.
    - You can use Direct Connect to provide predictable latency between your HQ and AWS.
    - You can use the global infrastructure to have mupltiple copies of your environment, in regions that is closest to your customer base.
    - You can also use caching services such as ElastiCache or CloudFront to reduce latency.
    - Space-time Trade-off
      * How do you select the appropriate proximity and caching solutions for your system?
      * How do you ensure that you continue to have the most appropriate proximity and caching solutions as new solutions are launched?
      * How do you monitor your proximity and caching solutions to ensure performance is as expected?
      * How do you ensure that the proximity and caching solutions you have matches demand?

**Key AWS Services:**

* Compute:
  + Autoscaling.
* Storage:
  + EBS, S3, Glacier.
* Databases:
  + RDs, DynamoDB, Redshift.
* Space-time Trade-off:
  + CloudFront, ElastiCache, Direct Connect, RDS Read Replicas, etc.

**Exam Tips – Performance Efficiency:**

* Compute
* Storage
* Database
* Space-time Trade-off

**Exam Tips – Performance Efficiency – Questions:**

* Compute:
  + How do you select the appropriate instance type for your system?
  + How do you ensure that you continue to have the most appropriate instance type as new instance types and features are introduced?
  + How do you monitor your instances post launch to ensure they are performing as expected?
  + How do you ensure that the quantity of your instances matches demand?
* Storage:
  + How do you select the most appropriate database solution for your system?
  + How do you ensure that you continue to have the most appropriate database solution and features as new database solution and features are launched?
  + How do you monitor your databases to ensure performance is as expected?
  + How do you ensure the capacity and throughput of your databases matches demand?
* Database:
  + How do you select the most appropriate database solution for your system?
  + How do you ensure that you continue to have the most appropriate database solution and features as new database solution and features are launched?
  + How do you monitor your databases to ensure performance is as expected?
  + How do you ensure the capacity and throughput of your databases matches demand?
* Space-time Trade-off:
  + How do you select the appropriate proximity and caching solutions for your system?
  + How do you ensure that you continue to have the most appropriate proximity and caching solutions as new solutions are launched?
  + How do you monitor your proximity and caching solutions to ensure performance is as expected?
  + How do you ensure that the proximity and caching solutions you have matches demand?