Time left 0:07:51

Question 1

Not yet answered

Marked out of 1.00

A deadline is approaching, and your build keeps failing tests.

- a. Roll back to old version
- b. Communicate issue early and seek help
- c. Stay silent and debug overnight
- d. Push partial code

Clear my choice

Question 2

Not yet answered

Marked out of 1.00

A system doubles its storage every 12 months. If the initial capacity is 4 TB, what is the capacity after 3 years?

- a. 32 TB
- b. 64 TB
- c. 16 TB
- d. 128 TB

Clear my choice

Question 3

Not yet answered

Marked out of 1.00

Find the next term: 2, 4, 8, 16, 32, ?

- a. 64
- b. 80
- c. 48
- d. 60

Clear my choice

Question 4

Not yet answered

Marked out of 1.00

If 4 engineers replicate a dataset in 8 hours, how many engineers are needed to finish in 4 hours (same efficiency)?

- a. 10
- b. 8
- c. 6
- d. 12

Clear my choice

Question 5

Not yet answered

Marked out of 1.00

If data replication takes 5 minutes for 10 GB, how long for 50 GB under the same bandwidth?

- a. 50 min
- b. 25 min
- c. 30 min
- d. 20 min

Question 6

Not yet answered

Marked out of 1.00

If latency between nodes is halved, throughput will likely:

- a. Double
- b. Drop to half
- c. Decrease
- d. Stay constant

Question 7

Not yet answered

Marked out of 1.00

Nasuni's architecture most resembles which design principle?

- a. Distributed file system
- b. Monolithic system
- c. Centralized database model
- d. Single-threaded core service

Question 8

Not yet answered

Marked out of 1.00

The team wants to use a new CI/CD tool you've never heard of.

- a. Wait for others to decide
- b. Research and prototype it
- c. Resist change
- d. Reject due to risk

Question 9

Not yet answered

Marked out of 1.00

What does CI/CD stand for?

- a. Continuous Iteration / Code Definition
- b. Continuous Intelligence / Continuous Debugging
- c. Cloud Integration / Code Delivery
- d. Continuous Integration / Continuous Deployment

Question 10

Not yet answered

Marked out of 1.00

What does Jenkins primarily automate?

- a. Build, test, and deployment pipelines
- b. Database backups
- c. File encryption
- d. UI rendering

Question 11

Not yet answered

Marked out of 1.00

What is common between Nasuni and Git?

- a. Encryption algorithms
- b. Object versioning and replication
- c. Blockchain verification
- d. Centralized control

Question 12

Not yet answered

Marked out of 1.00

Which cloud concept matches Nasuni's file architecture?

- a. Pure block storage
- b. Object storage with metadata intelligence
- c. In-memory caching only
- d. Local RAID replication

Question 13

Not yet answered

Marked out of 1.00

Which Docker command lists all active containers?

- a. docker active
- b. docker ps
- c. docker list
- d. docker show

Question 14

Not yet answered

Marked out of 1.00

Which Git command merges feature branches?

- a. git connect
- b. git merge
- c. git join
- d. git link

Question 15

Not yet answered

Marked out of 1.00

Which is the odd one out conceptually?

- a. NFS
- b. FTP
- c. NAS
- d. SAN

Question 16

Not yet answered

Marked out of 1.00

Which of the following best explains ‘eventual consistency’?

- a. Data syncs asynchronously but reaches same state eventually
- b. No replication occurs
- c. Data always synchronized instantly
- d. Only master copy is updated

Question 17

Not yet answered

Marked out of 1.00

Which option best represents scalability?

- a. Adding more CPU cores to increase performance
- b. Limiting users per region
- c. Caching data locally
- d. Deleting data to save space

Question 18

Not yet answered

Marked out of 1.00

You discover data loss during replication tests.

- a. Raise immediate alert and begin rollback
- b. Reboot system and hope it resolves
- c. Hide the issue to fix later
- d. Delete logs to reduce noise

Question 19

Not yet answered

Marked out of 1.00

You find a small bug that rarely affects users. What’s your approach?

- a. Report and schedule for next sprint
- b. Patch and push fix immediately
- c. Delete the module
- d. Log it and ignore

Question 20

Not yet answered

Marked out of 1.00

You're working on a live deployment and a critical bug appears. Your teammate blames your module. What do you do?

- a. Investigate first and share logs openly
- b. Escalate to manager
- c. Ignore and continue my work
- d. Defend my code immediately