Datadog

(Datadog Fundamentals)

Datadog Fundamentals

Total: **90 Questions**

Link:

MYEXAM.FR

Intel® Architecture Software Developer's Manual – "Memory Hierarchy and RAM":
https://software.intel.com/content/www/us/en/develop/articles/intel-sdm.html

Question: 4

A high number of page faults usually indicates insufficient

A. storage
B. RAM
C. CPU

Answer: B

Explanation:

Justification

Page faults occur when the operating system cannot find the required pages of a process in physical memory and must retrieve them from secondary storage.
A **high frequency of page faults** therefore signals that the amount of available **RAM** is insufficient to hold the working set of active processes.
When RAM is limited, the OS constantly swaps pages in and out, increasing fault rates; this is independent of storage capacity, CPU speed, or other resources.
Consequently, the metric is used primarily to diagnose memory scarcity, making **B. RAM** the correct choice.
Options A (storage) and C (CPU) relate to different performance domains: storage latency affects I/O-bound workloads, while CPU utilization pertains to instruction execution, neither of which directly cause an increase in page-fault count.

References

Datadog Monitoring: Memory-related metrics –https://docs.datadoghq.com/monitors/memory-metrics/
Understanding page faults in OS design –https://www.ibm.com/docs/en/zos/2.5.0?topic=performance-page-faults

Question: 5

Which two IP addresses are located in the same subnet?

A. 192.168.1.1/16 and 192.168.2.1/16
B. 192.168.1.1/32 and 192.168.1.65/32
C. 192.168.1.1/24 and 192.168.2.1/24

Answer: A

Explanation:

A

The pair that share the same subnet is the one where the network portion derived from the prefix length matches for both addresses.

Cisco – Classless Inter-Domain Routing (CIDR) Overview:
https://www.cisco.com/c/en/us/support/docs/ip/classless-inter-domain-routing-cidr/71923-classless-inter-domain-routing.html
RFC 1918 – Address Allocation for Private Internets (shows 255.255.255.0 mask for /24 networks):
https://datatracker.ietf.org/doc/html/rfc1918#section-3

Question: 7

Compared to TCP-based communication. UDP-based communication is

A. more secure
B. less reliable
C. more compressed

Answer: B

Explanation:

Key Explanation

UDP is a connectionless, best-effort protocol that does not provide sequencing, acknowledgments, flow control, or error recovery. Consequently, it is **less reliable** than TCP, which guarantees ordered, loss-free delivery through handshakes, retransmissions, and congestion control.
Security is not an inherent property of the transport layer; UDP does not encrypt or authenticate traffic any more than TCP. “More secure” would only apply if an application adds security features on top of UDP, which is not guaranteed.
Compression is unrelated to the transport protocol. UDP offers no built-in compression; any compression must be performed by the application or higher-level protocol.

Why the other choices are unsuitable

A – More secure – Security is not provided by UDP; it requires additional mechanisms.
C – More compressed – Compression is independent of the transport layer and is not a characteristic of UDP.
D – None of the above – One of the options (B) accurately describes the fundamental difference.

Answer: B. less reliable

References

IETF RFC768 – User Datagram Protocol: https://datatracker.ietf.org/doc/html/rfc768
Datadog documentation on UDP monitoring – https://docs.datadoghq.com/networking/udp/quickstart/

Question: 8

Which of the following can be graphed with metrics exposed by Cloud Networking Monitoring?

A. throughput
B. UDP retransmits
C. packet size

Question: 11

Which of the following serve as a key foundation for Datadog observability?

A. errors
B. tags
C. traces

Answer: B

Explanation:

Why "tags" are a foundational element of Datadog observability

Tags are immutable key-value pairs attached to every metric, log, and trace, enabling flexible classification without altering the underlying data source.
They allow you to slice-and-dice data across dimensions (e.g., service:auth, env:prod) for rapid root-cause analysis, custom dashboards, and automated alerting.
By providing a consistent, query-friendly way to group and filter data, tags turn raw telemetry into an organized, searchable observability fabric.

Why the other options are not central foundations

Errors – While Datadog can aggregate error counts, "errors" are just one type of event; they are not a structural building block for observability.
Traces – Traces are essential for distributed tracing, but they rely on the tagging infrastructure to be meaningful; by themselves they do not provide the broader dimensional filtering that tags enable.

Therefore, tags serve as the key foundation for Datadog observability.

References

Datadog Observability Fundamentals – Tagging: https://docs.datadoghq.com/tracing/setup/tagging/
Datadog Metrics & Metrics Explorer – Using tags to filter metrics:
https://docs.datadoghq.com.metrics/explore/tags/

Question: 12

Which data types are most fundamental to observability?

A. events, traces, logs
B. metrics, logs, tags
C. metrics, traces, logs

Answer: C

Explanation:

Correct option (C):Metrics, traces, and logs form the core "three pillars" of modern observability.

Metrics provide quantitative measurements of system health.
Traces capture the flow of requests across service boundaries, enabling end-to-end latency analysis.

Question: 15

In the following YAML block, what is the data type of excluded_filesystems? instances:
-use_mount: false
excluded_filesystems:
-tmpfs
-/dev/sda2
-/dev/sda1

A. a list
B. a mapping
C. a tuple

Answer: A

Explanation:

Justification

The YAML block defines excluded_filesystems under a top-level key, and its value is expressed as a series of items prefixed with -.
In YAML syntax, a hyphen followed by a space introduces an element of a **sequence (list)**.
Therefore excluded_filesystems holds an ordered collection of filesystem paths, i.e., a **list**, not a key-value map (mapping) or a tuple.
This matches the expected data-type for negative globbing patterns in Datadog's Agent configuration, where each entry of the list is treated individually.

References

YAML Specification – Lists and Sequences: https://yaml.org/spec/1.2.0/#lists
Datadog Agent Configuration – excluded_filesystems documentation:
https://docs.datadoghq.com/agent/configuration/#excluded_filesystems

Correct answer: A – a list.

Question: 16

Which file is used to specify the maximum number of log files that can be tailed in parallel?

A. auto_conf.yaml
B. datadog.yaml
C. conf.yaml

Answer: B

Explanation:

The correct answer is **B. datadog.yaml**.

Answer: B

Explanation:

Correct answer: B – to authorize a request being made to an API
Why this is the primary function:
An API key is a unique identifier that must be presented with each request to prove the caller's identity and entitlement to use the service.
It acts as a credential that grants or denies access based on the caller's permissions; without a valid key, the API rejects the request.
The key does **not** convey data payload or storage of request contents – its sole purpose is to enforce authentication/authorization.
Why the other options are unsuitable:
A – to specify the data to be returned by an API – Data shaping is controlled by request parameters (e.g., query strings, path variables, or request bodies), not by the API key itself.
C – to store the value of a request to an API – Storing request data is a function of logging or response handling mechanisms; an API key never stores request values.

References

Datadog API Authentication Overview: https://docs.datadoghq.com/api/latest/authentication/
Managing API Keys in Datadog: https://docs.datadoghq.com/api/latest/api_keys/

Question: 20

What is the benefit of attaching metadata to a monitored entities?

A. It provides additional context and attributes for an organizing and filtering data
B. It authenticates users accessing the monitoring dashboard
C. It encrypts sensitive information transmitted to the Datadog platform

Answer: A

Explanation:

Technical Justification

Option A is correct – Attaching metadata (tags, attributes) to monitored entities enriches each data point with contextual information (e.g., service name, environment, version). This enables more granular **organizing, filtering, and querying** of metrics, logs, and traces within Datadog, supporting precise alerting and analytics.
Option B is incorrect – Authentication to the Datadog dashboard is handled through API keys, OAuth tokens, or SSO, not by adding metadata to entities. Metadata does not verify user identity or access permissions.
Option C is incorrect – Encryption of data in transit is provided by TLS/SSL at the network layer. Metadata attached to entities does not encrypt information; it only adds descriptive tags that do not protect data confidentiality.

Conclusion – The primary benefit of metadata is to augment observability data with searchable context, making OptionA the only accurate statement.

References

Datadog – Tagging and Filtering
Datadog – Metadata for Metrics and Events

Answer: B

Explanation:

The autodiscovery template selects **%%port%%** to represent the **highest exposed container port** of the matched service, regardless of discovery order.
Variables such as **%%port_0%%** and **%%port_1%%** refer to the first and second discovered ports respectively; they do not automatically resolve to the highest value and can be lower than the maximum port.
With a container exposing ports **80, 443, and 8443**, **%%port%%** will evaluate to **8443**, while **%%port_0%%** and **%%port_1%%** would yield 80 and 443 (or whichever ports are discovered first), which are not the highest.
Consequently, option **B –%%port%%** is the only variable that reliably provides the highest exposed port, making it the correct answer.

References

Datadog Autodiscovery Template Variables:
https://docs.datadoghq.com/agent/autodiscovery/template_variables/
Service Types and Variables Overview: https://docs.datadoghq.com/agent/autodiscovery/service_types/?tab=windows#variables

Question: 23

To monitor an NGINX Docker container with the name custom_repo/ngnx:1.21, which value should be assigned to ad_identifiers in the Datadog Agent's Autodiscovery configuration file?

A. custom_repo/nginx:1.21
B. nginx
C. nginx:1.21

Answer: B

Explanation:

Technical justification

The ad_identifiers field in the Datadog Agent's Autodiscovery configuration expects the **service identifier** that the Agent uses to match a container to a built-in integration.
For the official NGINX integration the default identifier is simply **nginx**; the Agent looks for a service named nginx regardless of the image repository, tag, or full container name.
Option B – nginx matches exactly this identifier and therefore triggers the NGINX integration automatically.
Option A – custom_repo/nginx:1.21 includes the repository and tag, which are not part of the identifier name; the Agent would not recognize this as a built-in service and the integration would not be applied.
Option C – nginx:1.21 adds a tag after the colon, but tags are ignored when the identifier is a plain alphanumeric name; the extra :1.21 makes the string invalid as a recognized identifier, so the Agent would not associate it with the NGINX integration.

Hence, assigning **nginx** to ad_identifiers is the only option that aligns with the Agent's discovery rules and will correctly enable NGINX monitoring for the container custom_repo/ngnx:1.21.

References

best and most precise answer.

References

Datadog APM – Agent Port Configuration: https://docs.datadoghq.com/agent/apm/#agent-ports
APM Tracing – How Traces Are Sent to the Agent:
https://docs.datadoghq.com/tracing/trace_collection/trace_ingestion/

Both links are current as of the latest Datadog documentation (2024).

Question: 27

What is the purpose of the no-proxy setting in the Datadog Agent configuration file?

A. to specify metrics that must bypass the configured proxy
B. to specify log data that must bypass the configured proxy
C. to specify hosts that must bypass the configured proxy

Answer: C

Explanation:

Justification

The no-proxy setting in the Datadog Agent configuration (dogproxy.yaml or application.yaml) lists **hostnames or IP addresses** that should be contacted directly, without being routed through the HTTP(S) proxy configured for the Agent.

This is required for environments where certain internal services, metadata endpoints, or external APIs must be reachable even when a proxy is in place (e.g., AWS metadata service, internal monitoring endpoints, or DNS resolvers).

By explicitly naming those hosts, the Agent can bypass the proxy for them, preventing connectivity failures while still using the proxy for all other outbound traffic.

Option A – Incorrect; metrics are always sent by the Agent and are not controlled by a "no-proxy" list. Metrics are transmitted over the same transport (usually HTTPS to api.datadoghq.com) regardless of proxy settings.

Option B – Incorrect; log data is shipped via the same connection path as any other Agent telemetry (HTTP/HTTPS to the Logs collection endpoint). The no-proxy list does not filter specific log streams.

Option C – Correct; it directly matches the documented purpose of the no_proxy/no_provisional_proxy settings, which specify **hosts** that must be accessed without the proxy.

References

Datadog Agent Configuration – Proxy Settings: https://docs.datadoghq.com/agent/configuration/proxy/
Datadog Agent FAQ – No-Proxy Option: https://docs.datadoghq.com/agent/faq/no-proxy/