Datadog Fundamentals Exam Guide

# Introduction

The Datadog Fundamentals certification tests individuals’ fundamental

knowledge about the Datadog ecosystem. It covers a broad range of topics, from

Linux command line basics to metric visualization strategies. Certification

earners know how to use Datadog effectively for common use cases, and are

well equipped to learn more advanced topics specific to their roles.

This exam will test and validate knowledge on the following topics:

* Computer fundamentals such as CPUs, RAM, data storage
* Linux operating system basics
* YAML, JSON, Python, and shell scripts
* Networking basics including IP addresses and protocols such as HTTP,
* TCP, and UDP
* Install and configure the Datadog Agent in multiple environments
* Install and configure Datadog integrations
* Common techniques for troubleshooting Datadog
* Aggregate and visualize data in the Datadog application
* Search and filter events, logs and metrics using tags and facets
* Create Datadog monitors with targeted notifications
* Familiarity with the Datadog API

See Content outline, below for a complete list of the topics that this

exam covers.

Each exam has a maximum seat time of 2 hours.

# Target Candidates

Candidates should be entry-level Datadog users who understand

Observability principles. They should be frequent users of Datadog’s core

products and features.

RECOMMENDED DATADOG KNOWLEDGE

A Target Candidate should have the following knowledge:

* General computing knowledge (CPU, RAM, Data storage, etc)
* Linux Basics
* Reading YAML, JSON, Python, Shell Scripts
* Basic Networking
* Datadog
  + Agent
* Configuration
* Commands
  + Integrations
  + API
  + Metrics
  + Tags
  + Host Map
  + Monitors
  + Log Search & Filter
  + Metric Search & Filter

# Out of Scope

The following is a list (non-exhaustive) of specific job tasks that earners will

not be required to perform:

* Programming
* Systems Architecture
* Testing

# Exam Content

## QUESTION TYPE

The exam only contains one type of question:

* **Multiple choice**: Has one correct response and two or more incorrect responses (distractors)

Select the response that best answers the question posed. Incorrect

answers, also known as distractors, are options that can be a plausible

response for the question’s content area. Unanswered questions are scored

as incorrect.

The exam includes 75 questions that will affect your score.

## PRE TEST ITEMS

The exam includes 15 pre-test (unscored) questions that do not affect your

score. Datadog collects information about candidate performance on pretest

questions to evaluate them for future use on exam forms. Pretest questions

are not identified to the candidate on the exam.

# Key

* Need All
* Need Online Documentation and Udemy?
* Need Online Documentation
* Need Udemy?

# EXAM CONTENT OUTLINE

The Exam Content Outline includes the knowledge domains and related

subdomains of content on the exam.

### **1 Computer Fundamentals**

#### **A Config File Modification**

#### **B Operating Systems**

#### **C Programming Languages**

#### **D Hardware Concepts**

#### **E Shell**

#### **F Metadata**

#### **G Networking**

### **2 Infrastructure Development**

#### **A Agent Installation**

* Agents are installed depending on OS
* run from command line
* require API Key
* Docker Environment
  + Agent runs in its own container
  + Docker agent container relies on environment vars instead of config file

#### **B API Key**

* allows agent through the Server API

#### **C Application Key**

#### **D Running the Agent**

#### **E Agent Hostname**

* set in datadog.yaml
* if not explicitly set, heuristics will be used to tag each log / metric

### **3 Networking & Agent Configuration**

#### **A Datadog Ports**

#### **B Datadog IP Addresses**

#### **C Auto-discovery**

* auto-discovery label keys begin with
  + com.datadoghq.ad

### **4 Data Collection**

#### **A DogStatsD**

#### **B Crawlers**

#### **C Agent Integrations**

* configures datadog to understand specific types of apps / services
* categories of integrations
  + agent-based
    - installed with datadog agent and use a python class method called ‘Check’ to define metrics
  + Authentication (crawler) based
    - Enable communication between datadog and external services
    - Example
      * AWS, Slac, Azure, etc
  + Library
    - Datadog API client code you can import into your application
* Some integrations can be installed automatically if they require no configuration

#### **D API Endpoints**

#### **E Tagging Best Practices**

#### **F Metrics & Timeseries**

* Metrics
  + track about anything
  + overall picture of system
  + can come from
    - built in
    - integrations
      * can look up what is collected by viewing integration
    - custom
  + View in Metrics menu
  + Types of metrics
    - Work
      * Throughput
      * Success
      * Error
      * Performance
      * Only page on these types of metrics
    - Resource
      * Utilization
      * Saturation
      * Error
      * Availability
      * Some of these can be alerted on.
    - Events
      * Code changes
      * Alerts
      * Scaling events
      * etc
* SLO – Service Level Objective
  + Track metric and monitor trends over time
  + Framework for defining clear targets around application performance
  + Percentage of metric over time
* Alerts
  + Can set thresholds to sent notification of when metric breaches

### **5 Troubleshooting Datadog**

#### **A Agent Commands**

#### **B Agent Logs**

#### **C Agent Config Files**

* located in the install folder
* yaml files

### **6 Data Visualization & Utilization**

#### **A Host Map**

* visualize environment
* identify outliers
* detect usage patterns
* optimize resources

#### **B Dashboards**

* visually track, analyze and display info
* integrations create their own dashboard
* can clone and edit dashboards
* can copy widget from one dashboard to another
* can be shared internally or publicly available
* export as JSON
* widget
  + sections of dashboard

#### **C Using Metrics**

* Can graph directly from Metrics explorer

#### **D Using Tags**

* Tags come from each agent or integration. But you can add tags as needed

#### **E Monitors and Alerts**

* Monitors can watch a metric
* Use any metric reporting to datadog
* Multi alerts by device, host, etc
* Use @ in alert messages to direct notifications to the right people
* Schedule downtimes to suppress notifications for system shutdowns, off-line maintenance
* Watchdog
  + <https://docs.datadoghq.com/watchdog/#overview>

# Study and Preparation

It is expected that you have done some preparation for the exam. This can

include not only using the platform, but also taking some of Datadog’s recommended courses and reading relevant documentation. Below please find a

list of recommended materials to go over to get you started.

## Key

* Need to read

## COURSES

* Intro to Datadog: Dev + SRE
  + <https://learn.datadoghq.com/courses/dd-101-dev>
  + <https://learn.datadoghq.com/courses/dd-101-sre>
* Introduction to Monitoring
  + <https://learn.datadoghq.com/courses/intro-to-monitoring>
* Introduction to Integrations
  + <https://learn.datadoghq.com/courses/intro-to-integrations>

## DOCUMENTATION

* Getting Started
  + <https://docs.datadoghq.com/getting_started/>
* Datadog Agent
  + <https://docs.datadoghq.com/agent/>
* Metrics
  + <https://docs.datadoghq.com/metrics/#what-are-metrics>
* Infrastructure List
  + <https://docs.datadoghq.com/infrastructure/list/#aliases>
* API and Application Keys
  + <https://docs.datadoghq.com/account_management/api-app-keys/>
* Integrations
  + <https://docs.datadoghq.com/developers/integrations/new_check_howto/?tab=configurationtemplate>
* Docker Daemon
  + <https://docs.datadoghq.com/integrations/docker_daemon/>
* Query to the graph
  + <https://docs.datadoghq.com/dashboards/guide/query-to-the-graph/>
* Docker Agent
  + <https://docs.datadoghq.com/containers/docker/?tab=standard>
* Metrics API
  + <https://docs.datadoghq.com/api/latest/metrics/>
* Monitor types
  + <https://docs.datadoghq.com/monitors/types/>
* DogStatsD
  + <https://docs.datadoghq.com/developers/dogstatsd/?tab=hostagent#how-it-works>
* Ddtrace
  + <https://ddtrace.readthedocs.io/en/stable/advanced_usage.html>
* Canonical Hostnames
  + <https://github.com/DataDog/datadog-agent/blob/main/docs/agent/hostname_force_config_as_canonical.md>

Three Pillars of Observability

* Metrics
  + Baseline and anomilies
* Logs
  + What happened and how it happened
* Traces
  + Follow activity from request to function and service call