Oracle Functions FAQ

(Updated April 13, 2021)

General | **Key concepts** | **Security and access control**

General

What is Oracle Functions?

Oracle Functions is a serverless platform that lets developers create, run, and scale applications without managing any infrastructure. Functions integrate with Oracle Cloud Infrastructure, platform services and SaaS applications. Functions is based on the open source Fn Project. Code based on Functions typically runs for a short duration, and customers pay only for the resources they use while their function is running.

Which languages does Oracle Functions support?

Oracle Functions is based on open source Fn Project which provides function development kits (FDKs) for Java (including, the GraalVM native image), Python, Node, Go and Ruby, and in addition, lets you bring your own Dockerfile.

How do I scale my function?

You do not have to manually scale your function to handle the incoming request volume. Oracle Functions will automatically scale your function up (spin up new function containers to handle the incoming request volume) and then scale it down (terminate idle function containers as the incoming request volume decreases).

Do I need to provision compute instances for Oracle Functions to run on (like with the Oracle Container Engine for Kubernetes)?

No, all the compute resources necessary to run functions are automatically provisioned by Oracle Functions in the service tenancy. You don't need to provision any compute instances.

Can I run Oracle Functions on my own computer or in my data center?

Oracle Functions is a fully managed service that runs on Oracle Cloud Infrastructure. However, it is built on the open source Fn Project which can be run anywhere - in any cloud or on-premises. The advantage of Functions is that it is truly serverless and can operate at cloud scale. You can download and install the open-source distribution of Fn Project, develop and test a function locally, and then use the same tooling to deploy that function to Oracle Functions.

What can I do with this service?

With Oracle Functions, you can develop serverless solutions in Oracle Cloud. Functions has out-of-the-box integration with other services like API Gateway, Events Service, Notifications, Service Connector Hub, Streaming (Apache Kafka-compatible), Logging, and Oracle Integration. This lets you *run code in response to events*. You can move beyond simple notifications to taking action when something changes in your cloud environment. Using Oracle Functions, you can implement serverless event-driven architectures, web and mobile API backends, real-time file and stream processing solutions, DevOps, and enterprise security use cases.

How does Oracle Functions work?

You write your function code using an Fn Project function development kit (FDK), run the Fn Project CLI to package your function code as a Docker image, and push the image to Oracle Cloud Infrastructure Container Registry (OCIR). Optionally, you can configure one or more event triggers via the Events Service, API Gateway, Notifications, Service Connector Hub (Streaming, Logging), and Oracle Integration. Your function code runs only when the event occurs, or when it's invoked directly using the API/SDK/CLI.

What is the pricing for Oracle Functions?

With Oracle Functions you only pay for resources used while your function is running, and there is no charge when your function is idle. Please see the Oracle Functions page (pricing section) for details.

How do I get started with Oracle Functions?

To get started with Oracle Functions, please use the Oracle Functions Quick Start Guide. Once you have your first function up and running, you can use the other resources listed under Sample Functions, Solution Playbooks, Reference Architectures, and Developer Tutorials.

Where can I find the documentation for Oracle Functions?

Please see the Oracle Functions service documentation.

Which services can my function use?

Your function can access any Oracle Cloud Infrastructure service using the OCI SDK. Your function can interact with Compute, Storage, Networking, Vault, Autonomous Transaction

Processing (ATP), Autonomous Data Warehouse (ADW), MySQL, NoSQL, Registry, Streaming, Email, and other services. In addition, your function can access any service on the internet, as long as your network administrator has configured the appropriate network access for your function to use.

Which events can trigger a function?

Oracle Cloud Infrastructure services like API Gateway, Events Service, Notifications, Service Connector Hub, Streaming (Apache Kafka-compatible), Logging (Audit Logs, Network Flow Logs, etc.), and Oracle Integration can trigger a function. For a complete list of triggers and events sources, see Invoking Functions from other Oracle Cloud Infrastructure Services.

How do I monitor my function?

You can monitor the health, capacity, and performance of your function using built-in Oracle Cloud Infrastructure metrics, alarms and notifications. Oracle Functions monitors function execution, and collects and reports metrics such as:

- Invocations: number of times a function is invoked
- Duration: length of time a function ran for
- Errors: number of times a function failed
- Throttles: number of requests to invoke a function that returned a '429 Too Many Requests' error in the response

See Function Metrics for more information.

How do I access my function logs?

The Oracle Cloud Infrastructure Logging service is the default and recommended option for accessing, searching, and storing function logs. Alternatively, you can send your function logs directly to an external logging destination like Papertrail by configuring the syslog URL using Fn Project CLI.

See Storing and Viewing Function Logs for more information.

How do I access my function traces?

The Oracle Application Performance Management service is the default and recommended option for accessing, searching, and storing function traces.

See Distributed Tracing for Functions for more information.

Key concepts

What is an application?

In Oracle Functions, an application is both a unit of function runtime isolation and a logical grouping of related functions. It provides a context to store network configuration and environment variables that are available to all functions in the application. The Oracle Functions web console lists applications and their functions.

What is a function?

A function is a small block of code that does one thing. In Oracle Functions, functions are grouped into applications and are packaged as Docker images stored in Oracle Cloud Infrastructure Container Registry (OCIR). The function definition consists of memory and timeout settings, and a link to the corresponding Docker image. You can invoke a function directly using the CLI/API/SDK or using a service trigger. The Oracle Functions web console lists functions under the application they belong to.

What is a function invocation?

In Oracle Functions, a function's code is run when the function is invoked. You can invoke a function using:

- the Fn Project CLI
- the Oracle Cloud Infrastructure SDKs and CLI
- signed HTTP requests to the function's invoke endpoint
- other Oracle Cloud services (for example, triggered by an event via the API Gateway, Events service, etc.) or from external services

What happens when a function is invoked?

When a function is invoked for the first time, Oracle Functions pulls the function's Docker image from Oracle Cloud Infrastructure Container Registry, creates a container, and executes the function. Subsequent requests to the same function may be served by the same container. After a period of being idle, a function container is removed.

The Oracle Functions web console shows information about function invocations in metric charts and logs.

What are the key features of Oracle Functions?

Key features include:

- **No vendor lock-in** - Based on Fn Project, an open source FaaS platform, runs anywhere so no vendor lock-in.

- **Container native** - Docker is a first-class citizen. Package and run your functions in lightweight Docker containers.
- **Oracle Cloud integrated** - Native integration with Oracle Cloud Infrastructure (security, identity, networking, events, logs, metrics, traces, etc.).
- **Secure by design** - Fine grained permissions-based access, workload isolation across multiple tenants using OCI security features (OCI VM-based isolation, OCI networking, OCI identity, etc.).
- **Function triggers** - Direct invocation and from several other services, with more triggers to come.
- **Pay per use** - Fine grained billing, pay only for execution time, not for idle time.
- **Auto scaling -** Adjusts the number of running function instances to meet incoming request demand.
- **Highly available** - By default, functions run across available availability domains.
- **Function Development Kits (FDKs)** - Libraries to simplify authoring of functions in Java, Python, Go, Node, and Ruby.
- **Extensible** – By allowing you to bring your own custom Dockerfiles
- **Diagnostics** - Platform provides detailed metrics, execution logs and traces.

Security and access control

Can my function access resources on the internet?

If your function needs access to resources on the internet, your network administrator must set up a virtual cloud network (VCN) with ***public*** subnet(s) and an ***internet gateway***. You can select the subnet(s) while creating the application that contains your function. See Create the VCN and Subnets to Use with Oracle Functions for details.

How do I prevent my function from accessing resources on the internet?

If your function does ***not*** need access to resources on the internet, your network administrator must setup a virtual cloud network (VCN) with ***private*** subnet(s) and a ***service gateway***. And then you can select the subnet(s) while creating the application that contains your function. See Create the VCN and Subnets to Use with Oracle Functions for details.

How do I allow my function to access other Oracle Cloud Infrastructure resources?

To enable a function to access another Oracle Cloud Infrastructure resource, you can include the function in a dynamic group, and then create a policy to grant the dynamic group suitable permissions to access that resource. Having set up the policy and the dynamic group, you can use the resource principal provider, included in the Oracle Cloud Infrastructure SDK, in your function code. See Accessing Other Oracle Cloud Infrastructure Resources from Running Functions for more details.