Function Compute

FAQ

FAQ

FAQs

General

- What is Function Compute?
- What is FC function?
- What is trigger?
- What kind of programming languages I can use to write FC function?
- How long can a FC function run?
- When should I use FC versus ECS?
- How does Function Compute secure my code?
- How do I scale my FC function execution?
- Can I access the infrastructures that run my FC function?
- How do I monitor my FC function execution?

VPC

- What is VPC?
- How can I access my private VPC resources with FC function?
- How many VPCs a FC function can access?
- How can I access resources in classic network?
- How can I enable internet access for my VPC function?

Programming with FC

- How can I run code, such as C++, other than the supported programming languages?
- Docker is currently popular, why FC does not support custom Docker images?
- How to automatically install dependencies?

Function running

- Why do I receive “permission denied” error from FC, but not in local environment?
- What if my application generates a large file or a large disk space is required? How can I

request for more disk spaces?
- Where can I get the information of how much memory is consumed for function call?
- Can one function call another function?
- How do I kill a function call that causes an infinite loop?
- The default maximum concurrent call limit per user is 100. Does it mean that my service can process up to 100 requests per second?
- Is my execution environment isolated and secure? How is that guaranteed?

Performance

- Is execution environment released after functions return? Can I reuse the resource or status buffered by the last call?

Logging and monitoring

- How can I download logs generated by function execution?
- The function was called successfully, why can’ t I see metrics, such as number of calls, on the CloudMonitor console?

Resource access

- How does FC access a RDS database?
- Why do I get “The AccessKeyID does not exist” error by using access/secret key ID obtained from function context parameter?

General

What is Function Compute?

Function Compute, also known as FC, is an event driven computing service that provides a secure environment for you to execute your code with comprehensive monitoring and logging. With Function compute, you will be freed from managing infrastructures, network resources, instance scaling and system load balancing. You just need to upload your code to our service and we will take care the rest of it to run your code with high scale and availability. You can focus more on developing your business logic with fast software iteration. Function Compute seamlessly integrates with many other Alibaba Cloud services. Your functions can be triggered upon by the events that generated by the upstream services.

What is FC function?

FC function is a piece of code that you upload to our service. We provide many programming languages for you to write your FC function. See our documentation for a full list of supported programming languages.

What is trigger?

Trigger is the mechanism that invokes your function. You can invoke your function by calling the InvokeFunction API directly or setting trigger on the function. Function Compute integrates with many other Alibaba Cloud services to invoke your functions. You can set a trigger on a OSS bucket to invoke your function whenever there is object uploaded to the bucket. See our documentation for a full list of supported triggers.

What kind of programming languages I can use to write FC function?

We currently support NodeJS, Python and Java. More programming languages are coming down the roadmap. See our documentation for a full list of supported programming languages.

How long can a FC function run?

The default function timeout is 3 seconds. Function timeout can be set with any value between 1 and 600 seconds.

When should I use FC versus ECS?

ECS provides a wide range of virtual machine instance types and options to customize operating system, network, security settings, software logging, software monitoring and software load balancing, allowing you to customize your own software stack solution end to end. You have the full control and ownership of the ECS instances. You can customize your own scaling logic with ECS or utilize Container Service to manage container clusters over ECS.

Function Compute abstracts the underlying infrastructures and provides a unified compute environment for you. With Function Compute, you are freed for managing infrastructures and software stack. Function Compute will provision the infrastructures to execute your functions and provides comprehensive monitoring and logging behind the scene. You do not need to worry about scaling and load balancing as Function Compute will handle it automatically for you. Function Compute allows you focus on more developing your main business logic to process events and requests.

How does Function Compute secure my code?

Function Compute encrypts your code and stores it in OSS. We perform integrity checks whenever your code is in use. Code execution is isolated with its own view of file system and network namespace.

How do I scale my FC function execution?

One of the great benefits is that you do not need to provision any resource to scale your function execution. Function Compute will scale the infrastructures to execute your function on your behalf as more invocation requests you are sending to us.

Can I access the infrastructures that run my FC function?

No. Function Compute owns and maintains all the infrastructures, including health check, apply security patches, operation system updates and other routine maintenance.

How do I monitor my FC function execution?

Function Compute will generate metrics per execution and output it to Cloud Monitor for free. You can view the FC metrics in Cloud Monitor portal. You can also turn on function logging and view the logs in Log Service.

VPC

What is VPC?
Virtual Private Cloud (VPC) is an isolated cloud network built for private usage. It allows you to logically isolate your cloud resources in a virtual network environment. All FC functions are running in the FC owned VPC network environment. FC function cannot access your private VPC resources by default due to the nature of VPC network isolation.

How can I access my private VPC resources with FC function?
Function Compute allows you to configure your functions to access the specific VPC. You can grant Function Compute permissions to manipulate elastic network interfaces (ENIs) and provide VPC-specific configuration information that includes VPC ID, vswitch IDs and security group ID. Function Compute will peer the function execution environment with the specific VPC by using the ENIs. Once VPC configuration is enabled, your function will run as if it is running inside the specific VPC.

How many VPCs a FC function can access?
Each function can be configured to access at most one private VPC. If your function needs to access more than one private VPCs, you can peer your private VPCs together and grant your function access to any one of the peered VPCs.

How can I access resources in classic network?
FC functions run in the VPC network environment. If your function needs to access resources in the classic network, you can use classic link to link the resources in classic network to your private VPC, and then grant your FC function access to your private VPC. Your FC function can access the resource in the classic network through your private VPC.

How can I enable internet access for my VPC function?
Function Compute provides an option to enable or disable the internet access for your functions. Once the internet access is enabled, Function Compute will prepare an execution environment with internet access. You have the option to enabled the VPC access with or without internet access by setting the internet access option. You can grant your VPC function internet access by either enabling the FC function internet access or setup an internet NAT in your private VPC with the FC function internet access disabled.

Programming with FC

How can I run code, such as C++, other than the supported programming languages?
We support more languages based on the customers' need. Try the following methods if your code is written in a language that we do not support yet:

- Rewrite the code in languages that FC supports. Node.js and Python provide a wide range of libraries and are more efficient in terms of development.
- Compile C/C++ programs into executable files, and run the files through system calls like fork.

- Compile C/C++ modules as shared libraries, and use libraries by binding in Python.

The following table lists the advantages and disadvantages of the methods.

Method	Difficulty	Performance loss	Scenario
Rewrite logic	Depends on the logical complexity	Depends on the language and scenario	Applicable to relatively simple business logic
Fork executable files	Low	High	Applicable to non latency sensitive functions, e.g. batch processing
Import shared library	High	Low	Applicable to latency sensitive functions

If the suggested methods do not satisfy your requirement, contact us. We can help to propose other solutions.

Docker is currently popular, why FC does not support custom Docker images?

To guarantee real-time scalability, the system must start the container within a short period of time. It could take several minutes to download a large image, which makes performance unacceptable. We recommend that you develop your application as micro-services, and decouple functions so that each function has clear responsibility. Thus, function code generally does not involve complex dependencies and is easy to build.

How to automatically install dependencies?

Function Compute requires users to upload code as a compressed file that contains all the dependencies. Dependency management method varies in different languages. For example, in Node.js, you can use npm to install the dependent packages to code directory, compress to zip, and create/update function with the compressed file. You can also use CLI tool fcli to install and zip the dependencies. See the detailed procedure in the examples.

Function running

Why do I receive “permission denied” error from FC, but not in local environment?

The permission denied error may happen during the zipping process, especially in Windows. Some compression tool does not retain the permissions of the original files, FC returns permission denied if the code is decompressed. Make sure that permission, 755 or 664, is set on all files inside the zip package.

How can I request for more disk spaces if my application generates a large file or a large disk space is required?

There is a temporary storage space of 512 MB available for function execution. If this space is

insufficient, consider using a streaming pattern to process the data. For example, using a stream processing can reduce the memory consumption of a function from 1 GB to 256 MB and the duration from several minutes to 10 seconds. Contact us if this method does not satisfy your requirement,

Where can I get the information of how much memory is consumed for function call?

You can find information such as highest memory consumption in the LogResult header of Sync call. Alternatively, you can view the resource consumption in CloudMonitor (CMS).

Can one function call another function?

Functions can call other functions. You can use FC InvokeFunction API to call specified functions.

How do I kill a function call that causes an infinite loop?

Two types of infinite loops exist: a infinite loop caused by logic of a single function, and an infinite recursion in calling multiple functions that cannot be aborted. For example, function A calls function B, and function B calls function A. The first case is easier to handle. Once a function is timed out, the system terminates the execution automatically, and a timeout error message is returned without any further financial loss. The second case can be tricky since this kind of call is allowed by the service. In this case, you must set up the monitoring alarm based on function-level metrics. Also, concurrent call limit is enforced to make your financial cost managable.

The default maximum concurrent call limit per user is 100. Does it mean that my service can process up to 100 requests per second?

QPS (Requeset Per Second) is related to the maximum concurrent call limit to some extend, but they are also different concepts. You can estimate QPS by the following formula: QPS * FunctionDuration = ConcurrentCall. For example, if you expect 10,000 requests per second, and the average request processing time is 1 second, the required concurrency limit must be greater than 10,000. If the average request processing time is 10 milliseconds, the required concurrent call limit is 10,000 x 0.01 (second) = 100. For more information, see related documentation. In addition, the concurrency is set to protect users from spending unexpected cost such as the infinite call recursion. If the default value is not enough, contact us to adjust the limit.

Is my execution environment isolated and secure? How is that guaranteed?

FC takes security as the highest priority in every aspect of the product and system design. Your function execution environment has the same isolation level as Alibaba Cloud ECS instances, that is, isolation is at virtual machine level instead of at container level. Further more, we provide strict protection on the network, storage, codes, and anti-DDoS attacks to guarantee the security of your function.

Performance

Is execution environment released after functions return? Can I reuse the resource or status buffered by the last call?

Functions are running inside containers. Containers are not immediately released after functions return. Containers are only released when no request is received within a certain period of time. If requests are received consecutively, functions can considered as **long-living**. Therefore, you can buffer resources, such as global variables, to optimize the performance. However, the correctness of your program cannot always rely on the assumption that the buffer is available. For example, the buffered data can become unavailable when the container or the machine is down or released.

Logging and monitoring

How can I download logs generated by function execution?

If your logs are saved in LogStore of Alibaba Cloud Log Service, you can view and download related contents using Log Service APIs.

The function was called successfully, why can’ t I see metrics, such as number of calls, in the CloudMonitor console?

Check if the account you use is a sub-account and if the account has read-only permission of the CloudMonitor Service.

Resource access

How does FC access a RDS database?

Since FC containers are not running on fixed IPs, you must allow the access from all IPs, which can be a security risk, thus we recommend that you not do so. FC supports VPC. Once the feature is available, users can access resources inside VPC securely after granting VPC access to FC.

Why do I receive “The AccessKey ID does not exist” error when using access/secret key ID obtained from function context parameter?

Function context parameter provides temporary credentials to access Alibaba Cloud resources which consist of AccessKey ID, AccessKey secret and security token. Make sure that security token is provided. The following is an example for accessing OSS in python function.

```
import json
import oss2
def my_handler(event, context):
evt = json.loads(event)
creds = context.credentials
# Do not forget to fill out "security_token"!
auth = oss2.StsAuth(creds.access_key_id, creds.access_key_secret, creds.security_token)
bucket = oss2.Bucket(auth, evt['endpoint'], evt['bucket'])
bucket.put_object(evt['objectName'], evt['message'])
return 'success'
```