

FaaS Platform Configuration Options

Configuration area	Configuration option	Description	AWS Option	AWS semantic value	AWS reference	GCP Option	GCP semantic value	GCP reference	Azure Option	Azure semantic value	Azure reference
Deployment	Region	Geographic region of the data center the function is running in	one of currently 24 Lambda deployment regions	Reference to a provider-specified region, e.g. "eu-central-1", "us-west-1"	Lambda Endpoints	one of the regions available to the current account / project (may depend on pricing tier)	Reference to one of the regions available	Set region List regions	region is set on the associated "resource group" (logical cloud entity)	Reference to one of the regions available	Regions Resource Groups
	Source code upload	Way of providing the source code artifacts	Upload of ZIP archive or Reference to pre-built Docker image	Enumeration option, package type "Zip" or "Image"	ZIP-based deployment Image-based deployment	Upload of source code as ZIP archive, no image-based container deployments	no explicit option, implicitly handled by deployment tooling	Deployment options	Upload of ZIP archive, source code files or reference of pre-built image	no explicit option, implicitly handled by deployment tooling	Deployment Options
Runtime	Runtime OS / Architecture	OS / Processor architecture of the computing entity executing the function	arm64 (ARM-based) or x86_64 (Intel-based)	Enumeration option	Lambda Instruction set architecture	<i>not configurable</i>	---	---	Choice between Linux or Windows	Enumeration option	Azure Functions hosting options
	Language Runtime	Language runtime environment executing the function	one of the provided runtimes (e.g. nodejs10.x) or "provided"	Reference to provider-specified runtimes	Lambda runtimes	one of the provided runtimes (e.g. nodejs10)	Reference to provider-specified runtimes	Cloud Functions Execution Environments	one of the provided runtimes (e.g. node), specific language version is configured separately (language-dependent)	Reference to provider-specified runtimes	Setting Language Runtime Setting Node Version
	Custom Language Runtime	Option to provide a custom runtime for languages not supported by default	binary can be supplied as part of the source code artifact or via "lambda layer"	Language runtime setting references "provided", binary must be called "bootstrap"	Custom Lambda runtimes	<i>not available</i>	---	---	binary can be supplied as part of deployment package	"custom-handler" setting contains reference to executable	Azure Functions custom handlers
Execution	Execution timeout	Maximum execution time	number of seconds, 1-900, default 3 seconds	quantitative numeric integer value	Lambda quotas	number of seconds, 1-540, formatted as time string, default "60s"	quantitative numeric integer value	Cloud Functions Timeouts	time in minutes/seconds, formatted as time string, limit depends on payment plan, theoretically unbounded except for HTTP response functions (230 second limit)	quantitative numeric time value	Function App Timeout duration
	Memory	Allocated memory	desired value in MB, 128-10240, 1MB steps, also adjusts the allocated CPU power, default 128MB	quantitative numeric integer	Lambda quotas	pre-defined memory size slots, e.g. 128MB, 256MB (default), highest slot: 8192MB, also adjusts the allocated CPU power	Enumeration value	Cloud Functions Memory MemoryCPU ricingins ins	memory limit depends on payment plan, no specific configuration possible	---	Payment Plan service limits
	Retry behavior	If and how the function should re-execute on failure	configuration of number of retries (0-2), maximum event age (60-21600 seconds), dead letter queue for failure events	number of retries, maximum event age: quantitative numeric integer dead letter queue: String reference	Error handling & retries	retries can be enabled or disabled, if enabled may retry indefinitely	(Boolean) enumeration value	Retrying Event-Driven Functions	configuration of maximum retry count, retry timing strategy (static or dynamic increase with min/max values)	retry count: numeric , strategy choice: enumeration , strategy config: numeric time string	Azure functions retry
	Runtime Environment Values	Configuring environment key/value pairs to be used in code	can be set, some values are reserved	key-value pair of names , some reserved	AWS Environment Values Reserved variables	can be set, some values are reserved	key-value pair of names , some reserved	Using Environment Variables	can be set (called "Application Settings"), some values reserved	key-value pair of names , some reserved	Work with application settings
Invocation	Invocation Event Definition	Configuration of which events trigger which functions	definition depends on event source, most configuration done at service creating the event (instead of the function) - exception: "event source mapping" option to configure messages from queues and streams	service-specific configuration options, no general type	Lambda Event Source Mappings	one of four event types: storage bucket change events, HTTP trigger, pub/sub topic, or custom service event, specified by reference of resource and event type	enumeration option of which type to configure, some of which contain a reference to specific resources and event types	Deployment Trigger Options	configured using "bindings", language-dependent: C# / Java use annotations, JS/TS / PowerShell/Python use separate key-value configuration file	reference to resource invoking the service	Azure Triggers and Bindings Binding Example
	Entry Point	Which source code function to invoke	name of file + name of method	String reference including file and function	Handler Configuration	name of method	String reference to method in code	Endpoint configuration option	part of invocation configuration ("binding"), language-dependent: C# / Java use annotations, JS/TS / PowerShell/Python use default method names + optional explicit name of file / method	if applicable: String reference to file / method in code	Triggers and Bindings
Concurrency	Running instances	Minimal concurrency, amount of "stand-by" instances	"provisioned concurrency", running function instances ready to serve immediately, limit depends on available "reserved concurrency", depends on account permissions (default: 1000 concurrent executions)	quantitative numeric integer value in the bounds of the account permissions	Lambda Provisioned Concurrency Lambda quotas	option to set a number of running function instances	quantitative numeric value, no limit specified, but must not interfere with maximum number of instances	Using minimum instances	option to configure number of "always running" instances and number of "pre-warmed" instances, acting as buffer when scaling	quantitative numeric values	Always ready instances Pre-warmed instances
	Concurrency limit	Maximal concurrency, limit of concurrently running instances	"reserved concurrency" sets an upper limit for concurrent instances	quantitative numeric integer, limit depends on number of reserved instances already allocated on the current account	Lambda Reserved Concurrency Lambda quotas	option to set a maximum number of concurrently running function instances	quantitative numeric value, no bounds specified	Using maximum instances	option to set a maximum number of instances, upper limit depends on payment plan	quantitative numeric value with bounds	Limit scale out
Networking	Virtual network access	Connection to a virtual network of resources with access policies	reference to security groups or subnets the function should be part of, upper limit (max. 5 security groups, 16 subnets)	lists of references to security groups and subnets	Lambda networking and VPC configurations Lambda access resources in VPC	connection to VPCs using previously created "VPC connector" (holds reference to subnet etc.)	reference to VPC connector	Connecting to a VPC network	option to add Function to a VNet (virtual network)	reference to VNet	Azure Functions Networking Options
	IP-based access	Access policies based on IP values and ranges	configured via "security groups" as part of VPC configuration	domain-specific numeric values : IP addresses, ranges and ports	VPC Security	configured via "VPC connectors"	domain-specific numeric values : IP addresses, ranges and ports	Configuring Serverless VPC access	configured via "Network Security Groups" or "App Service Restrictions"	domain-specific numeric values : IP addresses, ranges and ports	Network Security Groups App Service access restrictions
Permissions	Permissions to access the function	Who (user or other service component) can access the function	configured by adding permissions (permitted actions) on the function resource (the object) to a given role (collection of permissions)	references to resources, enumerative permission actions, pattern matching may not be too restrictive	AWS Lambda permissions	configured by adding a principal (user or service account) with role (collection of permissions) to the function	reference to function, reference to principal, enum/reference to role	GCP IAM Granting access to function	configured by assigning a role (with permissions) to an assignee (user/s or service/s) with the scope including the function		Azure RBAC Azure Function Permissions
	Permissions of the function	What the function can access	configured by adding permissions (actions on resources) to the function's role (collection of permissions)	references to resources, enumerative permission actions, pattern matching may not be too restrictive	AWS Lambda permissions	configured by giving the function's principal (its associated service account) a role (with permissions) on the resource that needs to be accessed	reference to function principal, reference to resource, enum/reference to role	GCP IAM Granting access to function	with the function as assignee, assign a role (with permissions), scoped to the accessed resource		Azure RBAC Azure Function Permissions