aws_lambda_cheatsheet.md

AWS Lambda Cheatsheet

Runtime Versions			
Type	**Versions**	**AWS SDK**	**Operating System**
Node.js	nodejs10.x	(JavaScript) 2.712.0	Amazon Linux 2
	nodejs12.x	(JavaScript) 2.712.0	Amazon Linux 2
Java	java11	(JDK) amazon-corretto-11	Amazon Linux 2
	java8.a12	(JDK) amazon-corretto-8	Amazon Linux 2
	java8	(JDK) java-1.8.0-openjdk	Amazon Linux
Python	python3.8	(Python) boto3-1.14.40 botocore-1.17.40	Amazon Linux 2
	python3.7	(Python) boto3-1.14.40 botocore-1.17.40	Amazon Linux
	python3.6	(Python) boto3-1.14.40 botocore-1.17.40	Amazon Linux
	python2.7	(Python) boto3-1.14.40 botocore-1.17.40	Amazon Linux
Ruby	ruby2.7	(Ruby) 3.0.3	Amazon Linux 2
	ruby2.5	(Ruby) 3.0.3	Amazon Linux
.NET Core	dotnetcore3.1	--	Amazon Linux 2
	dotnetcore2.1	--	Amazon Linux

Go	go1.x	--	Amazon Linux
Custom Runtime	provided.al2	--	Amazon Linux 2
	provided	--	Amazon Linux

Available Operating System		
Type	**Image**	**Kernel**
Amazon Linux	amzn-ami-hvm-2018.03.0.20181129-x86_64-gp2	4.14.171-105.231.amzn1.x86_64
Amazon Linux 2	Custom	4.14.165-102.205.amzn2.x86_64

Settings | Limits		
Description	**Settings | Limits | Explained**	**Can be increased**
Writable Path & Space	/tmp/ 512 MB	--
Default Memory & Execution Time	128 MB Memory 3 Second Timeout	--
Max Memory & Execution Time	3008 MB Memory (64 MB increments) 15 Minutes Timeout	--
Number of processes and threads (Total)	1024	--
Number of File descriptors (Total)	1024	--
Maximum deployment		--

package size	50 MB (zipped, direct upload) 250 MB (unzipped, including layers)	
Maximum deployment package size for console editor	3 MB	--
Total size of deployment package per region	75 GB	Can be increased upto Terabytes
Maximum size of environment variables set	4 KB	--
Maximum function Layers	5 layers	--
Environment variables size	4 KB	--
Maximum test events (Console editor)	10	--
Invocation payload Limit (request and response)	6 MB (synchronous) 256 KB (asynchronous)	--
Elastic network interpaces per VPC	250	Can be increased upto Hundreds
Lambda Destinations	• It sends invocation records to a destination (SQS queue, SNS topic, Lambda function, or EventBridge event bus) when the lambda function is invoked asynchronously	Can be increased upto Hundreds

	• It also supports stream invocation	
Monitoring tools	• (Default) CloudWatch Logs stream • AWS X-Ray • CloudWatch Lambda Insights (preview)	--
VPC	• When you enable VPC, your Lambda function will lose default internet access • If you require external internet access for your function, ensure that your security group allows outbound connections and that your VPC has a NAT gateway	--
Concurrency	• Concurrent Execution refers to the execution of number of function at a given time. By default the limit is 1000 across all function within a given region • AWS Lambda keeps 100 for the unreserved function • So, if there are 1000 then you can select from 900 and reserve concurrency for selected function and rest 100 is used for the unreserved function	Can be increased upto Hundreds of thousands
DLQ (Dead Letter Queue)	• Failed Lambda is invoked twice by default and the event is discarded • DLQ instruct lamnda to send unprocessed events to AWS SQS or AWS SNS • DLQ helps you troubleshoot and examine the unprocessed request	--
Throttle	• Throttle will set reserved concurrency of the function to zero and it will throttle all future invocation • If the function is throttled then it will fail to run	--

	• If the fucntion is ran from Lambda console then it will throw "Calling the Invoke API failed with message: Rate Exceeded."	
File system	• File system will allow you to add Amazon EFS file system, which provides distributed network storage for the instances of the function • To connect to the file system, you need to connect your lambda function to VPC	--
State machines	• Step Functions state machines which orchestrate this function • The Step Functions state machines page lists all state machines in the current AWS region with at least one workflow step that invokes a Lambda function	--
Database proxies	• Database proxy manages a pool of database connections and relays queries from a function • It uses Secrets Manager secret to access credentials for a database • To connect to the file system, you need to connect your lambda function to VPC	--

Execution Role (Common Execution Role Available)	
AWSLambdaBasicExecutionRole	Grants permissions only for the Amazon CloudWatch Logs actions to write logs.
AWSLambdaKinesisExecutionRole	Grants permissions for Amazon Kinesis Streams actions, and CloudWatch Logs actions.
AWSLambdaDynamoDBExecutionRole	Grants permissions for DynamoDB streams actions and CloudWatch Logs actions.
AWSLambdaVPCAccessExecutionRole	Grants permissions for Amazon Elastic Compute Cloud (Amazon EC2) actions to manage elastic network interfaces (ENIs).

AWSXrayWriteOnlyAccess	Grants permission for X-ray to to upload trace data to debug and analyze.

Add new permission

```
import boto3
client = boto3.client('lambda')

# Role ARN can be found on the top right corner of the Lambda function
response = client.add_permission(
    FunctionName='string',
    StatementId='string',
    Action='string',
    Principal='string',
    SourceArn='string',
    SourceAccount='string',
    EventSourceToken='string',
    Qualifier='string'
)
```

Execution | Invoke | Tweaks	
A Lambda can invoke another Lambda	Yes
A Lambda in one region can invoke another lambda in other region	Yes
A Lambda can invoke same Lambda	Yes
Exceed 15 minutes execution time	Yes (Can Tweak around)
How to exceed 5 minutes execution time	Self-Invoke , SNS, SQS
Asynchronous Execution	Yes (Async Exec)
Invoke same Lamba with different version	Yes

Setting Lambda Invoke Max Retry attempt to 0	Yes

Triggers	Description	Requirement
API Gateway	Trigger AWS Lambda function over HTTPS	API Endpoint name API Endpoint Deployment Stage Security Role
AWS IoT	Trigger AWS Lambda for performing specific action by mapping your AWS IoT Dash Button (Cloud Programmable Dash Button)	DSN (Device Serial Number)
Alexa Skill Kit	Trigger AWS Lambda to build services that give new skills to Alexa	--
Alexa Smart Home	Trigger AWS Lambda with desired skill	Application ID (Skill)
Application Load Balancer	Trigger AWS Lambda from ALB	Application Load Balancer Listener (It is the port that ALP receivce traffice) Host Path
CloudFront	Trigger AWS Lambda based on difference CloudFront event.	CloudFront distribution, Cache behaviour, CloudFront event (Origin request/response, Viewer request/response). To set CloudFront trigger, one need to publish the version of Lambda. **Limitations:** Runtime is limited to Node.js 6.10 /tmp/ space is not available Environment variables, DLQ & Amazon VPC's cannot be used

CloudWatch Events	Trigger AWS Lambda on desired time interval (rate(1 day)) or on the state change of EC2, RDS, S3, Health.	Rule based on either Event Pattern (time interval) Schedule Expression (Auto Scaling on events like Instance launch and terminate AWS API call via CloudTrail
CloudWatch Logs	Trigger AWS Lambda based on the CloudWatch Logs	Log Group Name
Code Commit	Trigger AWS Lambda based on the AWS CodeCommit version control system	Repository Name Event Type
Cognito Sync Trigger	Trigger AWS Lambda in response to event, each time the dataset is synchronized	Cognito Identity Pool dataset
DynamoDB	Trigger AWS Lambda whenever the DynomoDB table is updated	DynamoDB Table name Batch Size(The largest number of records that AWS Lambda will retrieve from your table at the time of invoking your function. Your function receives an event with all the retrieved records)
Kinesis	Trigger AWS Lambda whenever the Kinesis stream is updated	Kinesis Stream Batch Size
S3	Trigger AWS Lambda in response to file dropped in S3 bucket	Bucket Name Event Type (Object Removed, Object Created)
SNS	Trigger AWS Lambda whenever the message is published to Amazon SNS Topic	SNS Topic
SQS	Trigger AWS Lambda on message arrival in SQS	SQS queue Batch size **Limitation:** It only works with Standard queue and not FIFO queue

Troubleshooting		
Error	**Possible Reason**	**Solution**
File "/var/task/lambda_function.py", line 2, in lambda_handler return event['demoevent'] KeyError: 'demoevent'	Event does not have the key 'demoevent' or either misspelled	Make sure the event is getting the desired key if it is receiving the event from any trigger. Or if the not outside event is passed than check for misspell. Or check the event list by printing event.
botocore.exceptions.ClientError: An error occurred (AccessDeniedException) when calling the GetParameters operation: User: arn:aws:dummy:🔢assumed-role/role/ is not authorized to perform: ssm:GetParameters on resource: arn:aws:ssm:dummy	Lacks Permission to access	Assign appropriate permission for accessibility
ImportError: Missing required dependencies [‘module']	Dependent module is missing	Install/Upload the required module
sqlalchemy.exc.OperationalError: (psycopg2.OperationalError) could not translate host name "host.dummy.region.rds.amazonaws.com" to address: Name or service not known	RDS Host is unavailable	Make sure the RDS instance is up and running. Double check the RDS hostname
[Errno 32] Broken pipe	Connection is lost (Either from your side or may be some problem from AWS)	Make sure if you are passing the payload of right size.

	While invoking another Lambda, if the payload size exceed the mentioned limit	Check for the connection.
Unable to import module ‘lambda_function/index’ No module named ‘lambda_function'	Handler configuration is not matching the main file name	Update the handler configuration as per your filename.function_name
OperationalError: (psycopg2.OperationalError) terminating connection due to administrator command SSL connection has been closed unexpectedly	RDS/Database System has been rebooted. In a typical web application using an ORM (SQLAlchemy) Session, the above condition would correspond to a single request failing with a 500 error, then the web application continuing normally beyond that. Hence the approach is “optimistic” in that frequent database restarts are not anticipated.	Give second try
Error code 429	The function is throttled. Basically the reserved concurrency is set to zero or it have reach the account level throttle. (The function that is invoked synchronous and if it is throttled then it will return 429 error. If the lambda function is invoked asynchronously and if it is throttled then it will retry the throttled event for upto 6 hours.)	Check for the reserved concurrency limit or throttle status for the individual function. Or check for the account level concurrent execution limit

AWS Lambda CLI commands

Add Permission
It add mention permission to the Lambda function

Syntax

```
  add-permission
--function-name <value>
--statement-id <value>
--action <value>
--principal <value>
[--source-arn <value>]
[--source-account <value>]
[--event-source-token <value>]
[--qualifier <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Example

```
add-permission --function-name functionName --statement-id role-statement-id --action lambda:CreateFunction --principa
```

Create Alias

It creates alias for the given Lambda function name

Syntax

```
  create-alias
--function-name <value>
--name <value>
--function-version <value>
[--description <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Example

```
create-alias --function-name functionName --name fliasName --function-version version
```

Create Event Source Mapping

It identify event-source from Amazon Kinesis stream or an Amazon DynamoDB stream

```
  create-event-source-mapping
--event-source-arn <value>
--function-name <value>
[--enabled | --no-enabled]
[--batch-size <value>]
--starting-position <value>
[--starting-position-timestamp <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Example

```
create-event-source-mapping --event-source-arn arn:aws:kinesis:us-west-1:1111 --function-name functionName --starting-
```

Create Function

It creates the new function

Syntax

```
  create-function
--function-name <value>
--runtime <value>
--role <value>
--handler <value>
```

```
  [--code <value>]
  [--description <value>]
  [--timeout <value>]
  [--memory-size <value>]
  [--publish | --no-publish]
  [--vpc-config <value>]
  [--dead-letter-config <value>]
  [--environment <value>]
  [--kms-key-arn <value>]
  [--tracing-config <value>]
  [--tags <value>]
  [--zip-file <value>]
  [--cli-input-json <value>]
  [--generate-cli-skeleton <value>]
```

Example

```
  create-function --function-name functionName --runtime python3.6 --role arn:aws:iam::account-id:role/lambda_basic_exec
   --handler main.handler
```

Delete Alias

It deletes the alias

Syntax

```
    delete-alias
  --function-name <value>
  --name <value>
  [--cli-input-json <value>]
  [--generate-cli-skeleton <value>]
```

Example

```
delete-alias --function-name functionName --name aliasName
```

Delete Event Source Mapping

It deletes the event source mapping

Syntax

```
  delete-event-source-mapping
--uuid <value>
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Example

```
delete-event-source-mapping --uuid 12345kxodurf3443
```

Delete Function

It will delete the function and all the associated settings

Syntax

```
  delete-function
--function-name <value>
[--qualifier <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Example

```
delete-function --function-name FunctionName
```

Get Account Settings

It will fetch the user's account settings

Syntax

```
  get-account-settings
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Get Alias

It returns the desired alias information like description, ARN

Syntax

```
  get-alias
--function-name <value>
--name <value>
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Example

```
get-alias --function-name functionName --name aliasName
```

Get Event Source Mapping

It returns the config information for the desired event source mapping

Syntax

```
  get-event-source-mapping
--uuid <value>
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Example

```
get-event-source-mapping --uuid 12345kxodurf3443
```

Get Function

It returns the Lambda Function information

Syntax

```
  get-function
--function-name <value>
[--qualifier <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Example

```
get-function --function-name functionName
```

Get Function Configuration

It returns the Lambda function configuration

Syntax

```
  get-function-configuration
--function-name <value>
[--qualifier <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Example

```
  get-function-configuration --function-name functionName
```

Get Policy

It return the linked policy with Lambda function

Syntax

```
  get-policy
--function-name <value>
[--qualifier <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Example

```
get-policy --function-name functionName
```

Invoke

It invoke the mention Lambda function name

```
    invoke
  --function-name <value>
  [--invocation-type <value>]
  [--log-type <value>]
  [--client-context <value>]
  [--payload <value>]
  [--qualifier <value>]
```

Example

```
  invoke --function-name functionName
```

List Aliases

It return all the aliases that is created for Lambda function

Syntax

```
    list-aliases
  --function-name <value>
  [--function-version <value>]
  [--marker <value>]
  [--max-items <value>]
  [--cli-input-json <value>]
  [--generate-cli-skeleton <value>]
```

Example

```
list-aliases --function-name functionName
```

List Event Source Mappings

It return all the list event source mappings that is created with create-event-source-mapping

Syntax

```
  list-event-source-mappings
[--event-source-arn <value>]
[--function-name <value>]
[--max-items <value>]
[--cli-input-json <value>]
[--starting-token <value>]
[--page-size <value>]
[--generate-cli-skeleton <value>]
```

Example

```
list-event-source-mappings --event-source-arn arn:aws:arn --function-name functionName
```

List Functions

It return all the Lambda function

Syntax

```
  list-functions
[--master-region <value>]
[--function-version <value>]
[--max-items <value>]
[--cli-input-json <value>]
```

```
  [--starting-token <value>]
  [--page-size <value>]
  [--generate-cli-skeleton <value>]
```

Example

```
    list-functions --master-region us-west-1 --function-version ALL
```

List Tags

It return the list of tags that are assigned to the Lambda function

Syntax

```
    list-tags
  --resource <value>
  [--cli-input-json <value>]
  [--generate-cli-skeleton <value>]
```

Example

```
    list-tags --resource arn:aws:function
```

List Versions by functions

It return all the versions of the desired Lambda function

Syntax

```
    list-versions-by-function
  --function-name <value>
```

```
  [--marker <value>]
  [--max-items <value>]
  [--cli-input-json <value>]
  [--generate-cli-skeleton <value>]
```

Example

```
  list-versions-by-function --function-name functionName
```

Publish Version

It publish the version of the Lambda function from $LATEST snapshot

Syntax

```
    publish-version
  --function-name <value>
  [--code-sha-256 <value>]
  [--description <value>]
  [--cli-input-json <value>]
  [--generate-cli-skeleton <value>]
```

Example

```
    publish-version --function-name functionName
```

Remove Permission

It remove the single permission from the policy that is linked with the Lambda function

Syntax

```
 remove-permission
--function-name <value>
--statement-id <value>
[--qualifier <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Example

```
remove-permission --function-name functionName --statement-id role-statement-id
```

Tag Resource

It creates the tags for the lambda function in the form of key-value pair

Syntax

```
 tag-resource
--resource <value>
--tags <value>
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Example

```
tag-resource --resource arn:aws:arn --tags {‘key’: ‘pair’}
```

Untag Resource

It remove tags from the Lambda function

Syntax

```
 untag-resource
--resource <value>
--tag-keys <value>
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Example

```
untag-resource --resource arn:aws:complete --tag-keys [‘key1’, ‘key2’]
```

Update Alias

It update the alias name of the desired lambda function

Syntax

```
  update-alias
--function-name <value>
--name <value>
[--function-version <value>]
[--description <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Example

```
update-alias --function-name functionName --name aliasName
```

Update Event Source Mapping

It updates the event source mapping incase you want to change the existing parameters

Syntax

```
  update-event-source-mapping
--uuid <value>
[--function-name <value>]
[--enabled | --no-enabled]
[--batch-size <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```

Example

```
update-event-source-mapping --uuid 12345kxodurf3443
```

Update Function Code

It updates the code of the desired Lambda function

Syntax

```
  update-function-code
--function-name <value>
[--zip-file <value>]
[--s3-bucket <value>]
[--s3-key <value>]
[--s3-object-version <value>]
[--publish | --no-publish]
[--dry-run | --no-dry-run]
```

```
  [--cli-input-json <value>]
  [--generate-cli-skeleton <value>]
```

Example

```
  update-function-code --function-name functionName
```

Update Function Configuration

It updates the configuration of the desired Lambda function

Syntax

```
    update-function-configuration
  --function-name <value>
  [--role <value>]
  [--handler <value>]
  [--description <value>]
  [--timeout <value>]
  [--memory-size <value>]
  [--vpc-config <value>]
  [--environment <value>]
  [--runtime <value>]
  [--dead-letter-config <value>]
  [--kms-key-arn <value>]
  [--tracing-config <value>]
  [--cli-input-json <value>]
  [--generate-cli-skeleton <value>]
```

Example

```
  update-function-configuration --function-name functionName
```

References

- AWS Lambda Docs
- Boto 3 Docs
- AWS CLI Docs
- SQLAlchemy Docs