# Serverless Prerequisite for windows:

1. NVM (Node Version Manager) : For Managing Node version (**OPTIONAL STEP**)
   1. Download link : <https://github.com/coreybutler/nvm-windows>
   2. Verify nvm version : ***nvm –v***
   3. Install Node using nvm: ***nvm install 8.10.0***
2. Node : Install node version 8.10.0
   1. Verify Node version : ***node -v***
   2. Verify npm version : ***npm –v***
3. Python: Install Python version 3.7
   1. Verify Python version : ***python –V***
4. Install awscli using pip:
   1. ***pip install awscli***
   2. ***pip install --upgrade --user setuptools***
   3. ***pip install aws-sam-cli –user***
5. Configure AWS Access Key and Secret Key at local: (optional)

Run Command***: aws configure***

Now you have to give your access key and secret key, below is the sample data -

***AWS Access Key ID [None]: AKIAIOSFODNN7EXAMP***

***AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMP***

***Default region name [None]: us-west-2***

***Default output format [None]: ENTER***

# Serverless Installation for windows:

1. Install Serverless module using npm:
   1. ***npm install -g serverless***
   2. Verify serverless version ***: serverless -v***
2. Create first Serveless java project: Below command create skeleton project
   1. ***serverless create --template aws-java-maven –path(-p) firstProject***

Check below link for more serverless commands-https://serverless.com/framework/docs/providers/aws/guide/quick-start/

Or

Run Command: ***severless help***

# Run your first serverless project:

When you create serverless project (using serverless create command), it create a default Handler for Lambda function. To execute Handler code locally, below command need to run -

***sls invoke local –f hello***

## In case you face below error while running above command -

Serverless: In order to get human-readable output, please implement "toString()" method of your "ApiGatewayResponse" object.

java.lang.reflect.InvocationTargetException

        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

        at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)

        at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)

        at java.lang.reflect.Method.invoke(Unknown Source)

        at com.serverless.InvokeBridge.invoke(InvokeBridge.java:82)

        at com.serverless.InvokeBridge.<init>(InvokeBridge.java:36)

        at com.serverless.InvokeBridge.main(InvokeBridge.java:133)

Caused by: java.lang.NoSuchMethodError: com.amazonaws.services.lambda.runtime.LambdaLogger.log([B)V

        at com.amazonaws.services.lambda.runtime.log4j2.LambdaAppender.append(LambdaAppender.java:74)

        at org.apache.logging.log4j.core.config.AppenderControl.tryCallAppender(AppenderControl.java:156)

        at org.apache.logging.log4j.core.config.AppenderControl.callAppender0(AppenderControl.java:129)

        at org.apache.logging.log4j.core.config.AppenderControl.callAppenderPreventRecursion(AppenderControl.java:120)

        at org.apache.logging.log4j.core.config.AppenderControl.callAppender(AppenderControl.java:84)

        at org.apache.logging.log4j.core.config.LoggerConfig.callAppenders(LoggerConfig.java:448)

        at org.apache.logging.log4j.core.config.LoggerConfig.processLogEvent(LoggerConfig.java:433)

        at org.apache.logging.log4j.core.config.LoggerConfig.log(LoggerConfig.java:417)

        at org.apache.logging.log4j.core.config.LoggerConfig.log(LoggerConfig.java:403)

        at org.apache.logging.log4j.core.config.AwaitCompletionReliabilityStrategy.log(AwaitCompletionReliabilityStrategy.java:63)

        at org.apache.logging.log4j.core.Logger.logMessage(Logger.java:146)

        at org.apache.logging.log4j.spi.AbstractLogger.logMessageSafely(AbstractLogger.java:2091)

        at org.apache.logging.log4j.spi.AbstractLogger.logMessage(AbstractLogger.java:2005)

        at org.apache.logging.log4j.spi.AbstractLogger.logIfEnabled(AbstractLogger.java:1876)

        at org.apache.logging.log4j.spi.AbstractLogger.info(AbstractLogger.java:1421)

        at com.serverless.Handler.handleRequest(Handler.java:18)

        ... 7 more

## Solution:

Replace below dependency version with give-

from

<dependency>

<groupId>com.amazonaws</groupId>

<artifactId>aws-lambda-java-log4j2</artifactId>

<version>1.1.0</version>

</dependency>

to

<version>1.0.0</version>

# Implement Layer over Lambda Function:

## Configure Layer in serverless.yml file:

To configure layer we need to configure below set of lines, here “layerone” is the name of the layer and “artifact” is the path of zip file which needs to upload over layer.

layers:

layerone:

name: ${self:provider.stage}-layerone

description: Test for Lambda Layer

compatibleRuntimes:

- java8

package:

artifact: lambdaLayer/layer/package.zip

## Attach Layer with Lambda Function:

By using “Ref” keyword with Layer name (name should be followed by LambdaLayer) we can refer our defined layer with the function

functions:

hello:

handler: com.serverless.Handler

layers:

- {Ref: LayeroneLambdaLayer}