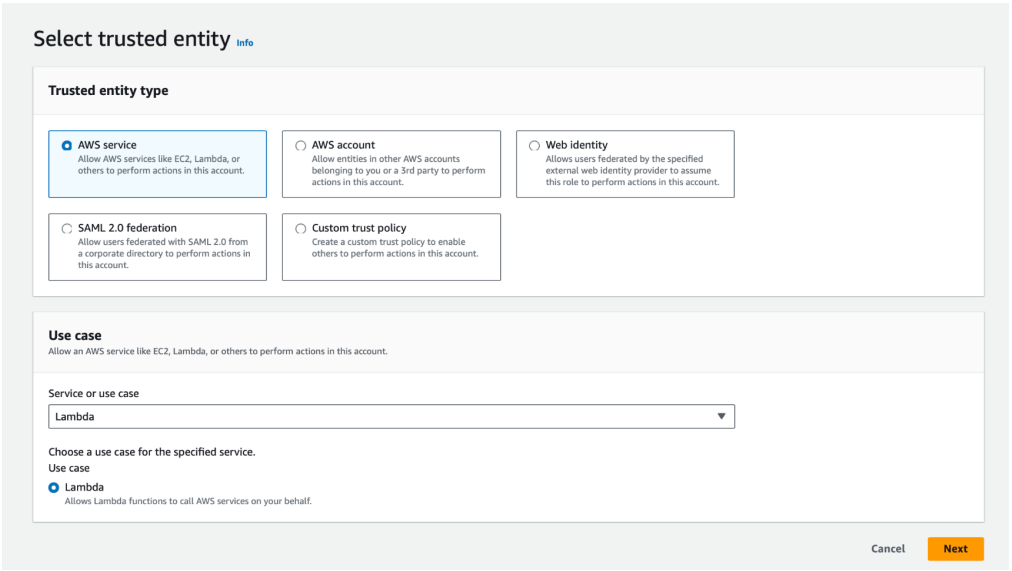
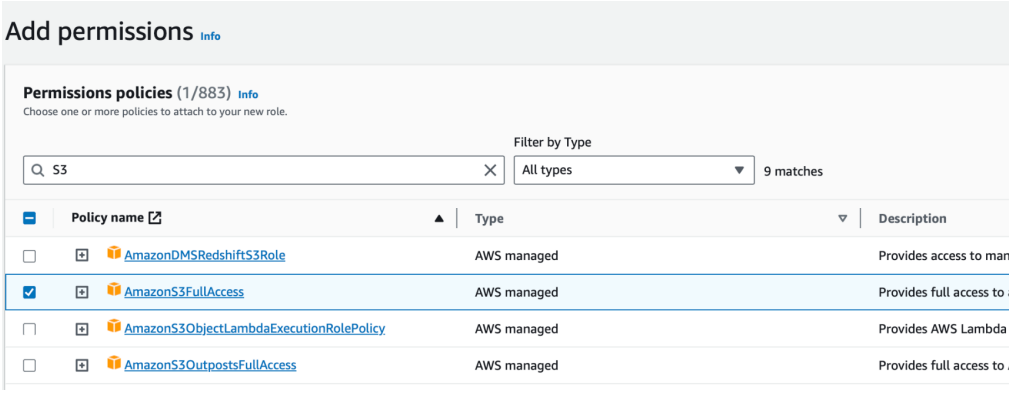
Instructions for creating AWS lambda function for caching the images.

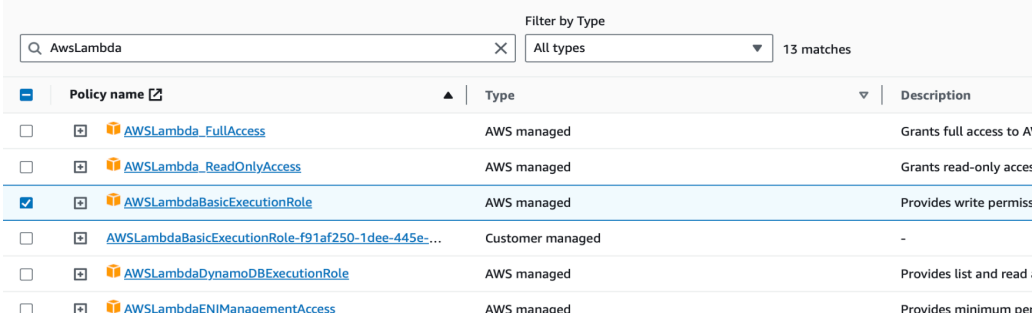
1. Creating IAM Role:

Before creating a lambda function we need to create an IAM Role for our lambda function to get full access to S3.
Roles:

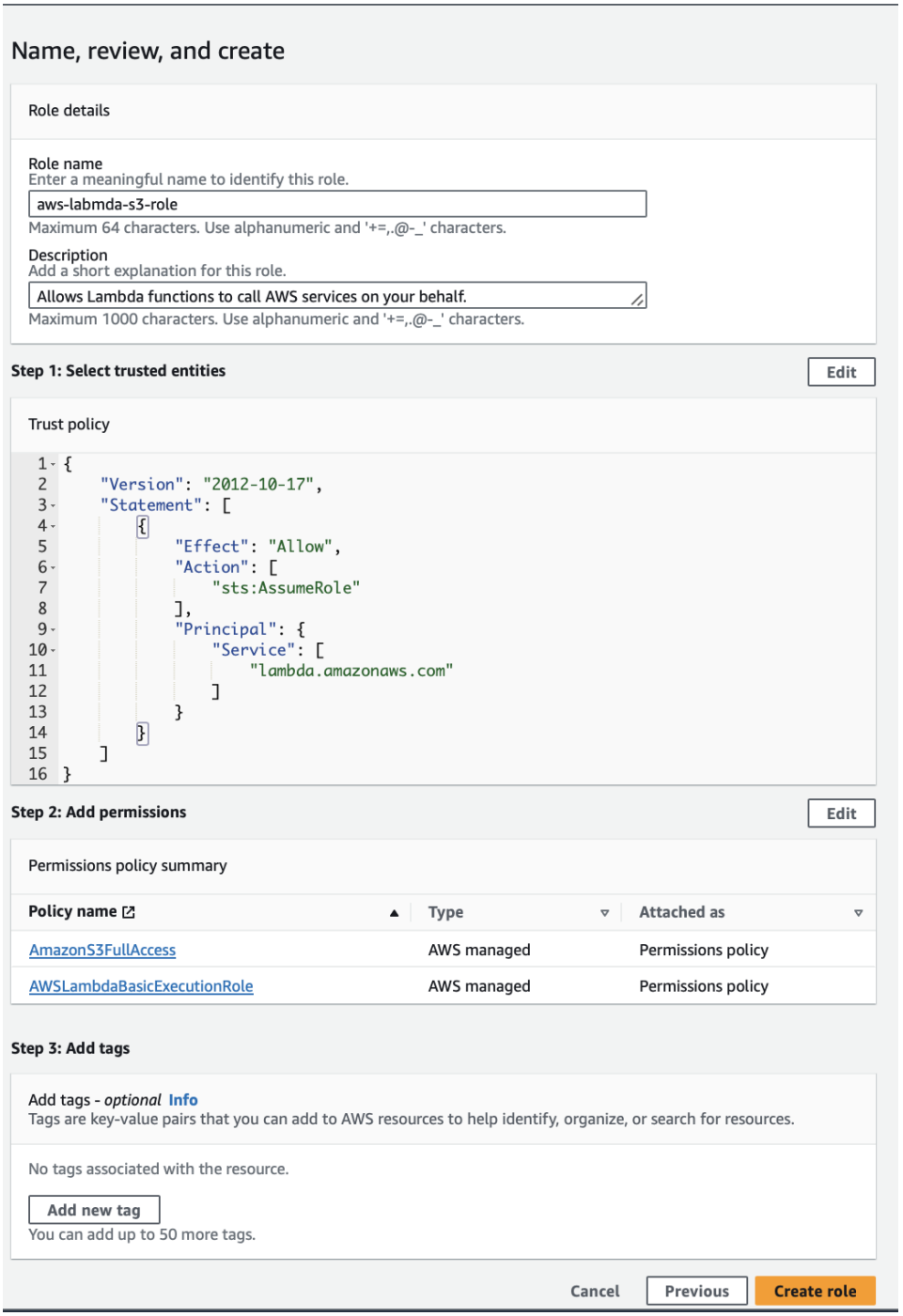
- AWSLambdaBasicExecutionRole
- AmazonS3FullAccess





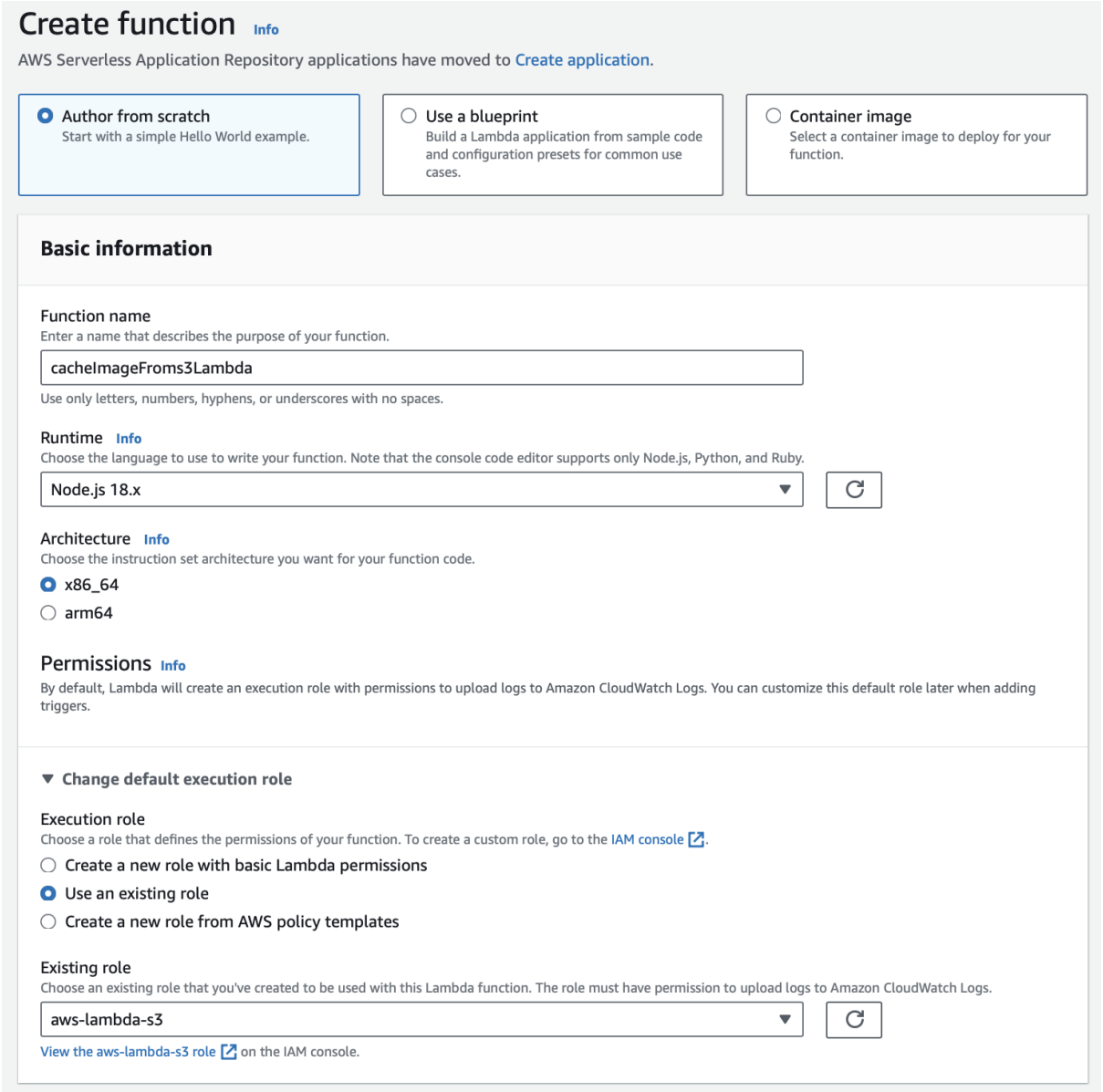


After configuring the permission our role should look like this. Give it the name you want and hit create role to finish the process.

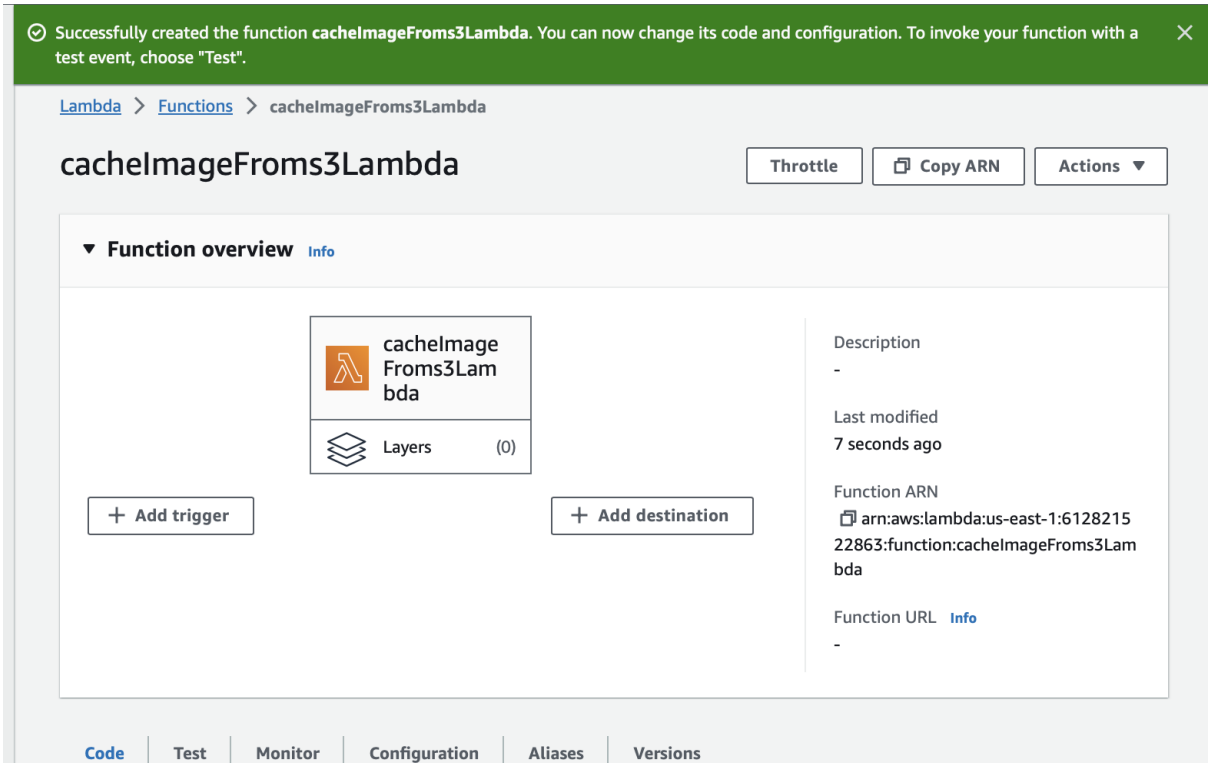


2. Creating lambda function

Step 1:
Here we are using the role that we created. Give it a name of your choice and make sure lambda has the same configuration as follows.

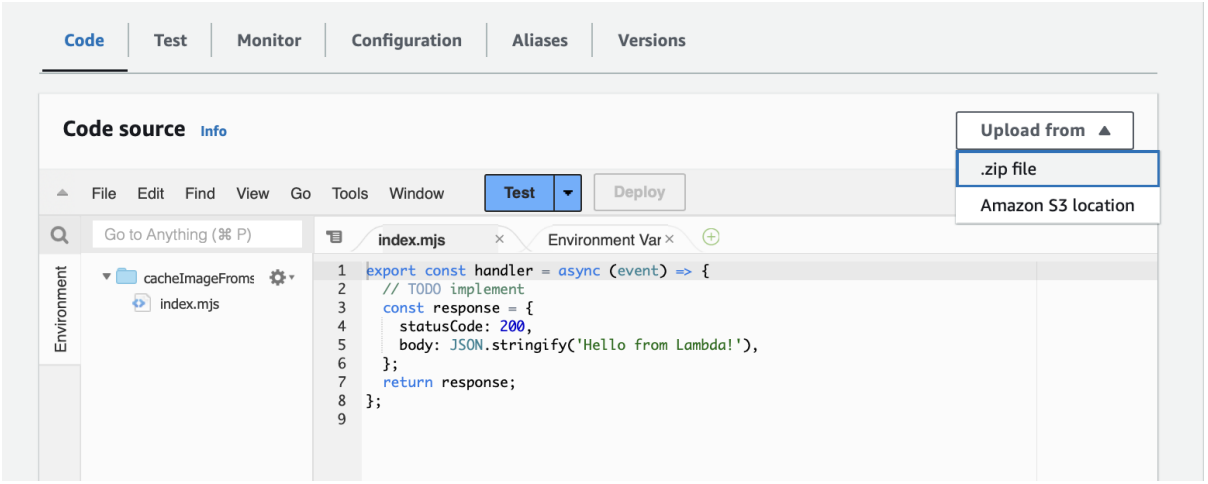


Function overview.



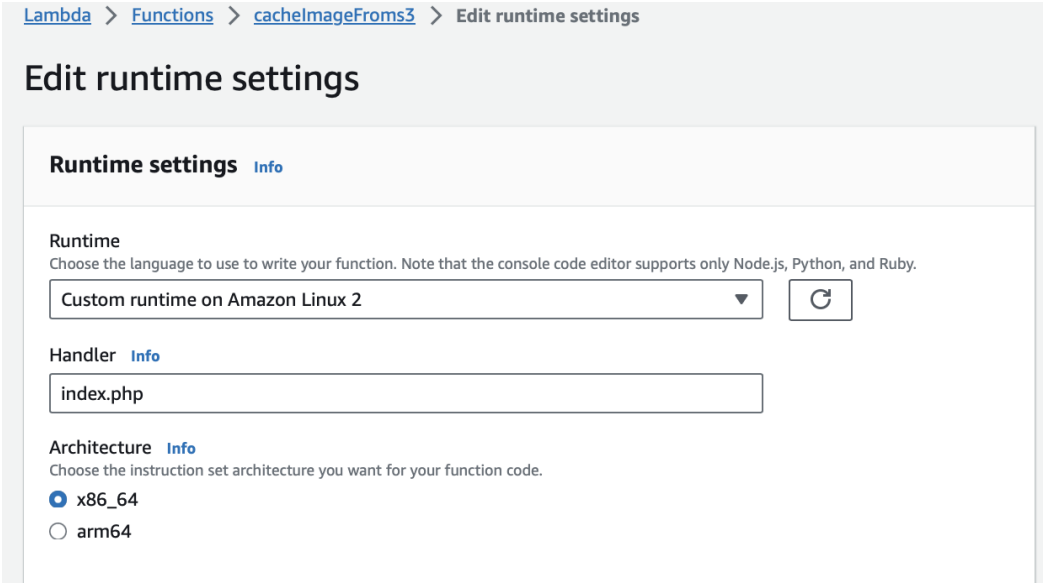
Add our zipped code from https://github.com/Tsolmonx/resize-image-lambda-php using command

```
zip -r9q index.zip .
```



3. Runtime settings

Edit Runtime settings as follows. Index.php is our entry point.

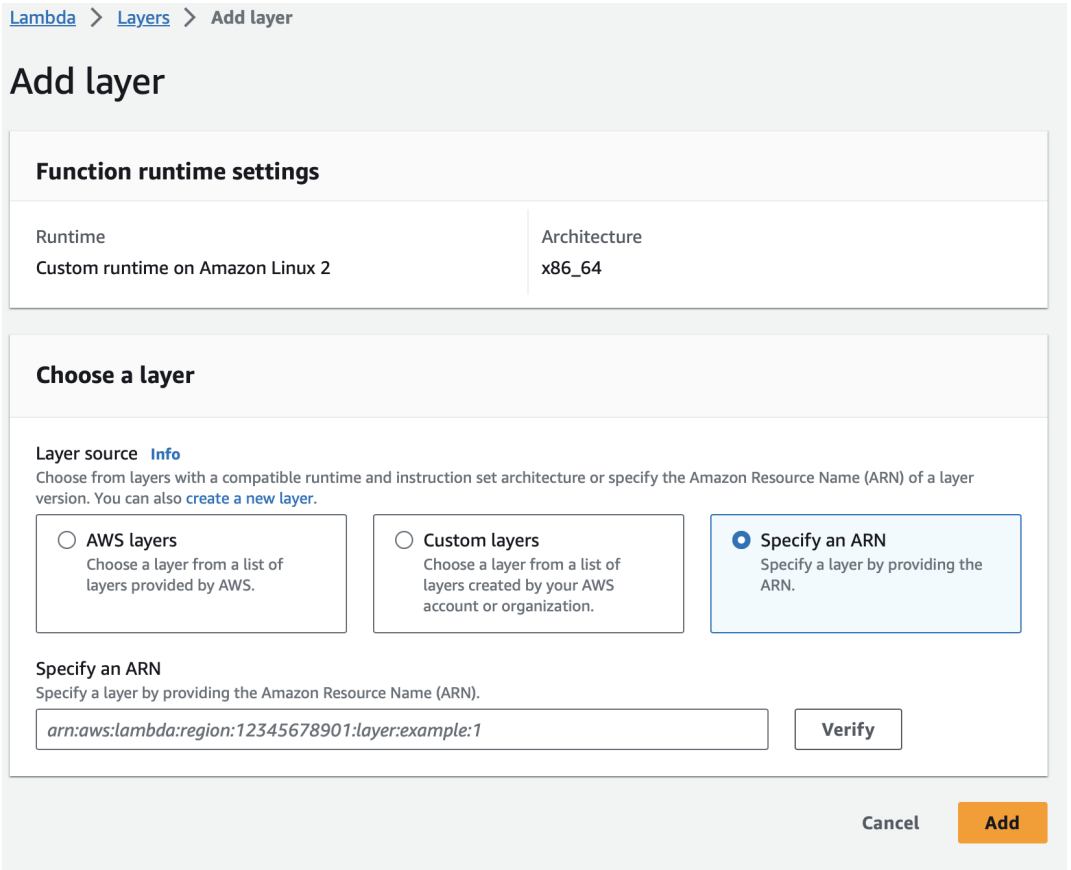


4. Layers

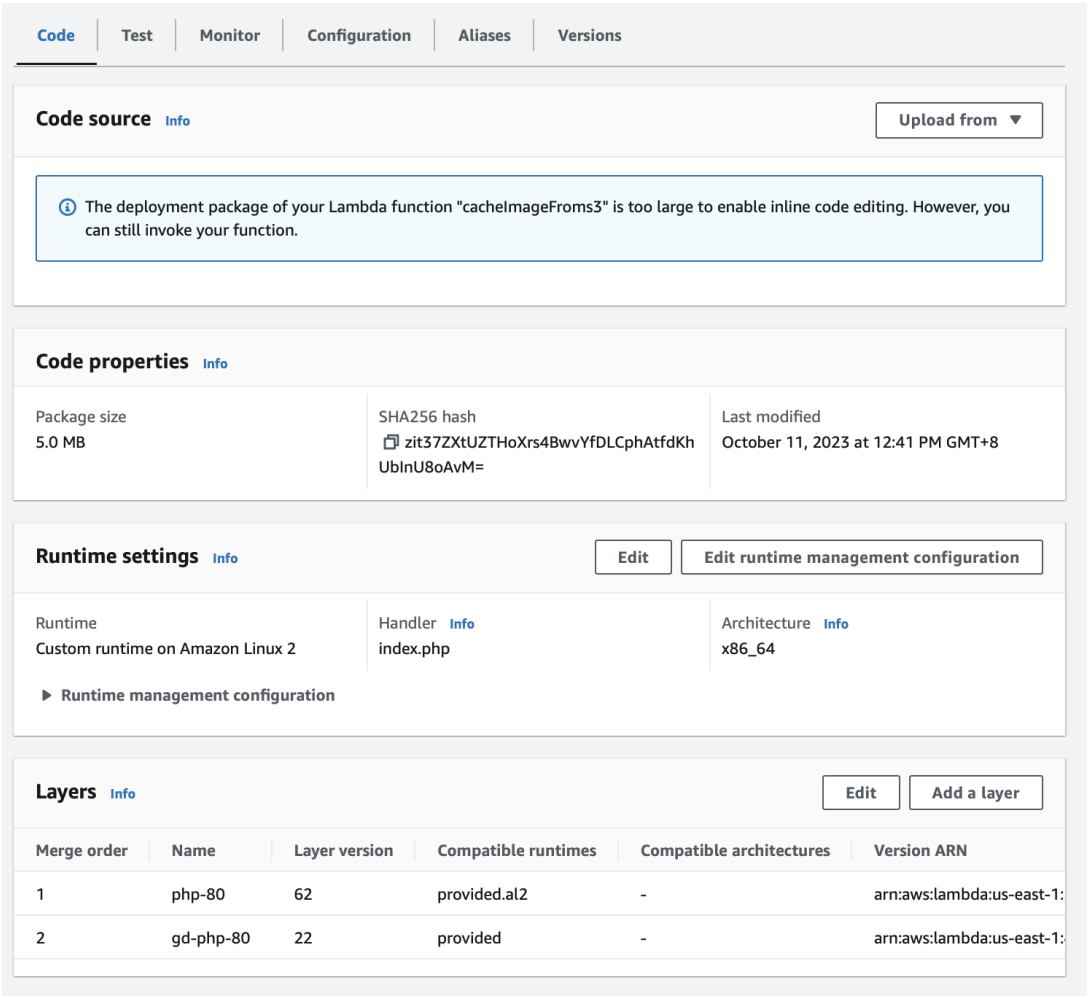
We need two layers for our function to work properly.

1: arn:aws:lambda:us-east-1:534081306603:layer:php-80:62

2: arn:aws:lambda:us-east-1:403367587399:layer:gd-php-80:22

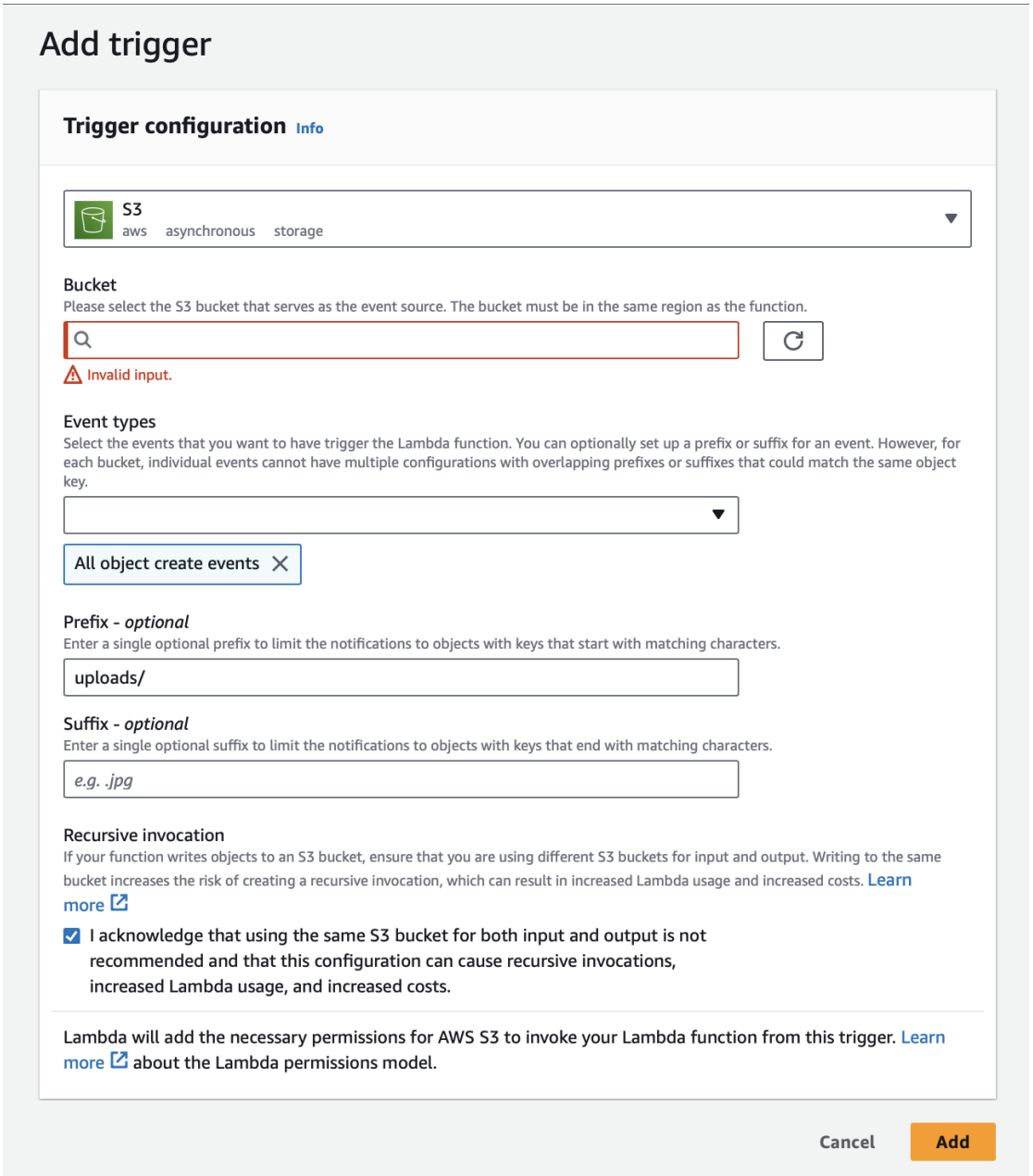


Configuration overview

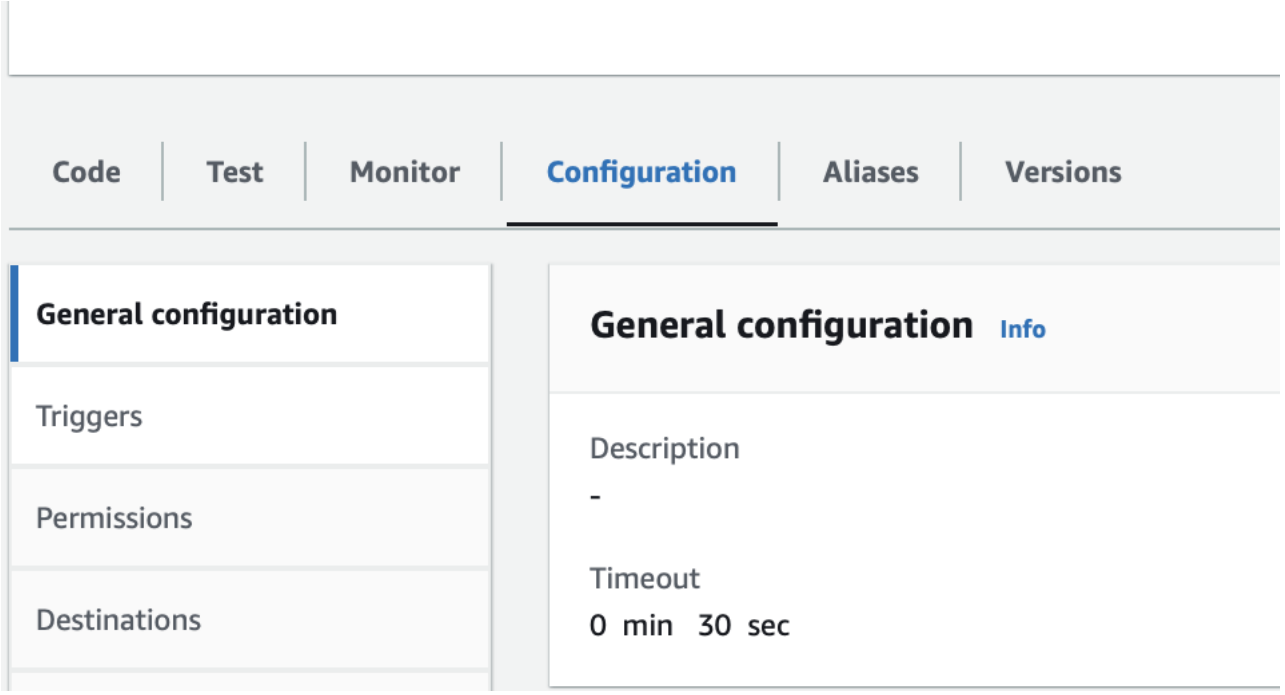


5. Adding trigger to our function

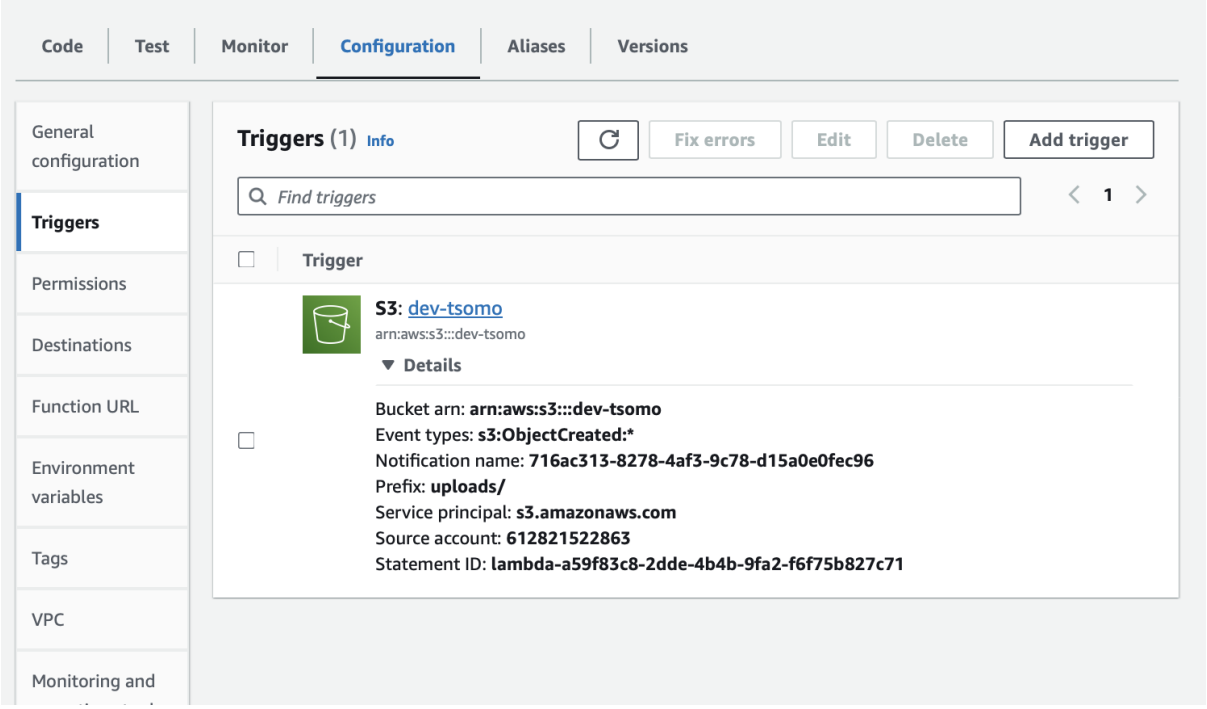
Whenever images are uploaded to S3 we will resize the image to 3 different sizes and upload it back to S3. To prevent an infinite loop when resizing and uploading it back to S3 we need to be careful and add a prefix for our trigger(uploads/). Adding this prefix will ensure our function works it's intended way and our resized image will be added in cache/ directory. And provide the bucket and subscribe to All object create event.



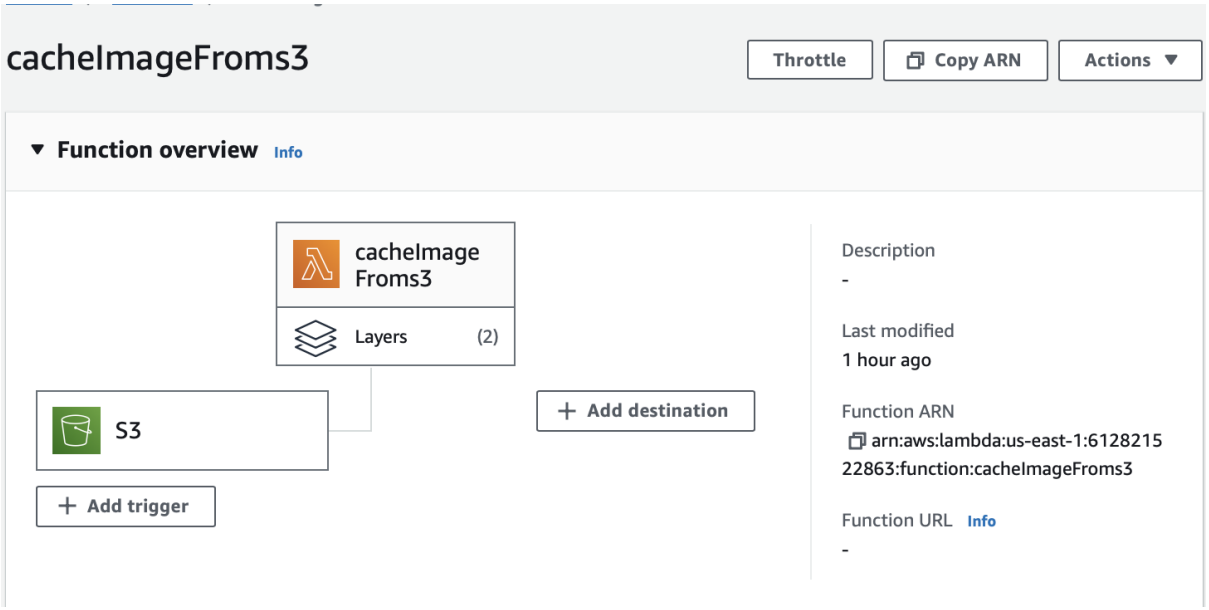
Increase the timeout to at least 30 second



Trigger overview

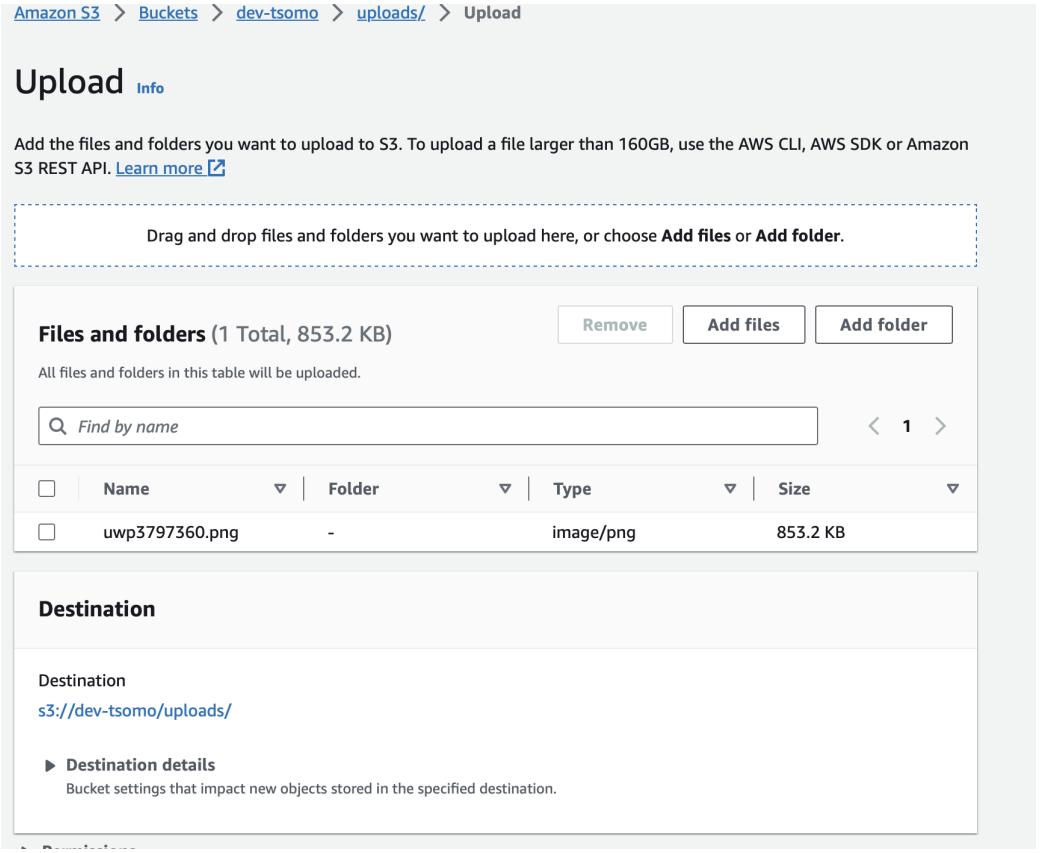


Function overview



6. Test

Test your lamba by uploading image to uploads directory



Here in cache/ directory we can see our lambda function created our desired resized images to its respective folders.