Section 5: Creating an image resize function

***Section Description (from the outline):*** In the last section we’ll jump into creating and deploying an image resize function. This API will be able to resize images on the fly. Meaning, it will only resize an image once the user requests it.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Video Number** | **Video Title** | **Problem / Solution (Not more than 50 words)** | **Step 1 (Not more than 10 words)** | **Step 2(Not more than 10 words)** | **Step 3(Not more than 10 words)** |
| 5.1 | Creating an S3 bucket for storing images | Before starting we need to create a resource on AWS, more precisely an S3 Bucket. Additionally, we need to configure public read permissions and static website hosting. | Creating an S3 Bucket. | Configure permissions. | Enable static website hosting. |
| 5.2 | Creating the image resize service | Start with creating a new service, and configuring the serverless.yml. We’ll be adding all the needed configuration settings including IAM role statements and environment variables. | Create a new service. | Configure the service. | Add environment variables and IAM role statements. |
| 5.3 | Writing the image resize function | Begin with installing modules for image resizing and interacting with AWS. Interaction is done with AWS-SDK while the resizing will be done by Sharp. The rest of the video will be hands on coding. | Installing modules. | Writing the logic. | Testing it with serverless offline. |
| 5.4 | Deploying the image resize service with Docker | Start the process of deploying the service. For this we need to install all modules on the same OS as the Lambda service is running on. Meaning we need to install it on Amazon Linux. To achieve this we’ll use Docker to build and deploy our image resize service. | Install Docker. | Build the Docker container. | Deploy the service from within the container. |
| 5.5 | Testing the image resize service | Go through testing the image resize service. And summarizing the whole course. | Testing the API. | Make sure new resized images are stored in the S3 bucket. | Create a course summary. |