

## CONFIGURING LAMBDA FUNCTION TO CONVERT VIDEO FORMAT

**By : Aadrika Siddharth Ishika**

- In wordpress name folder where you have to upload video as: video/
- create folder in s3 bucket named: convertd-video
- got to SNS create 2 topic - error and complete using standard type and also create subscription using mail id for both
- confirm subscription in mail for both
- now go to elastic transcoder service
  - click create new pipeline
  - give name
  - select your bucket which you have created previously  
select create console default role
  - further for all option select your bucket created previously and storage as standard
  - under notification service in completion and error event use existing sns topic which we previously created- complete and error respectively
  - create
  - copy pipeline id (will be used in environmental variable of lambda inside configuration section)  
###1650618903718-jpn6j5
- now go to IAM role and create role for Lambda (select lambda service under role not ec2) with following access
  1. arn:aws:iam::aws:policy/AmazonElasticTranscoder\_FullAccess
  2. AmazonS3FullAccess

### 3. CloudWatchFullAccess

### 4. AmazonSNSFullAccess

- Now come to lambda service and create a function
  - select "author from scratch"..nodejs12
  - select "use existing role"
  - select role created in previous step
  - create
  - now scroll down add following code in index.js

```
'use strict';
```

```
var AWS = require('aws-sdk'),
```

```
    transcoder = new AWS.ElasticTranscoder({
```

```
        apiVersion: '2012-09-25',
```

```
        region: 'ap-south-1'
```

```
    });
```

```
exports.handler = (event, context, callback) => {
```

```
    let fileName = event.Records[0].s3.object.key;
```

```
    var srcKey =
```

```
    decodeURIComponent(event.Records[0].s3.object.key.replace(/\+/g,  
    " "));
```

```
    var newKey = fileName.split('.')[0];
```

```
    console.log('New video has been uploaded:', fileName);
```

```
    transcoder.createJob({
```

```
        PipelineId: process.env.PIPELINE_ID,
```

```
Input: {
  Key: srcKey,
  FrameRate: 'auto',
  Resolution: 'auto',
  AspectRatio: 'auto',
  Interlaced: 'auto',
  Container: 'auto'
},
Output: {
  Key: getOutputName(fileName),
  ThumbnailPattern: '',
  PresetId: '1351620000001-300040',
  Rotate: 'auto'
}
}, function(err, data){
  if(err){
    console.log('Something went wrong:',err)
  }else{
    console.log('Converting is done');
  }
  callback(err, data);
});
};
function getOutputName(srcKey){
```

```

let baseName = srcKey.replace('videos/', '');
let withoutExtension = removeExtension(baseName);
return 'audios/' + withoutExtension + '.mp3';
}

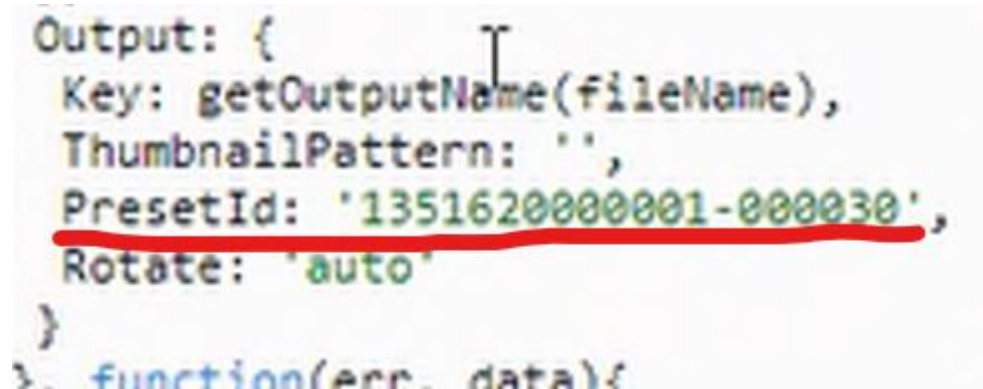
function removeExtension(srcKey){
    let lastDotPosition = srcKey.lastIndexOf(".");
    if (lastDotPosition === -1) return srcKey;
    else return srcKey.substr(0, lastDotPosition);
}

```

**change region, preset-id in code**

1351620000001-000030

#480 4:3 pixel



```

Output: {
  Key: getOutputName(fileName),
  ThumbnailPattern: '',
  PresetId: '1351620000001-000030',
  Rotate: 'auto'
}
3. function(app, data){

```

and click "deploy" button

now go to configuration tab

- in left pane you can see option of environment variable, where you can add you pipeline id, in following format

PIPELINE\_ID <value of id>

Environment variables (1)		Edit
The environment variables below are encrypted at rest with the default Lambda service key.		
Key	Value	
PIPELINE_ID	1650623217807-ujxvfx	

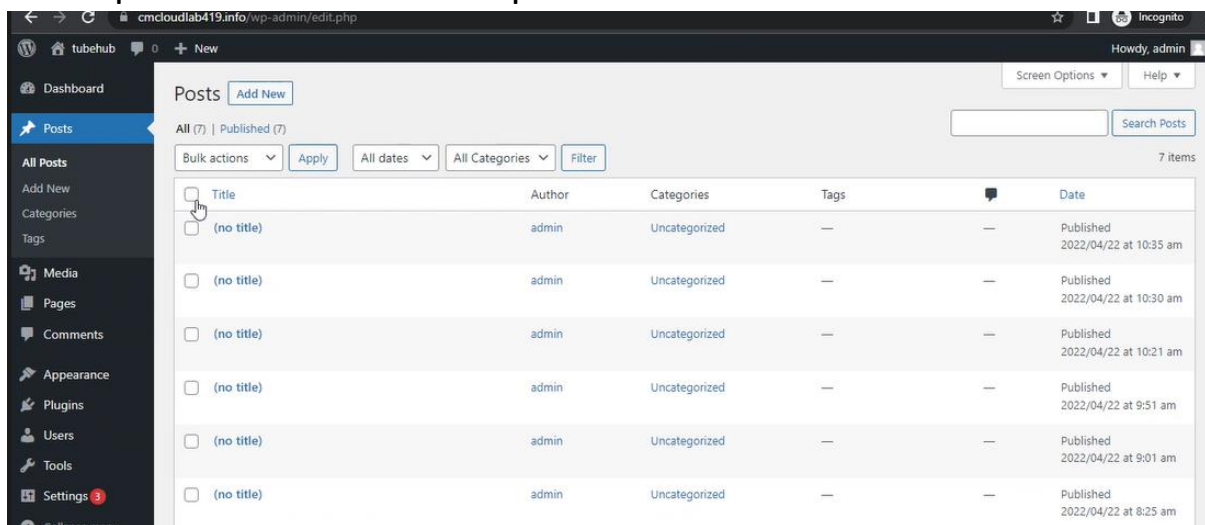
Now click "Add trigger" presented above

- select S3
- select bucketname
- in prefix add your folder name where your video will get uploaded by wordpress(in our case video/)
- in suffix add folder name where you converted video will get saved(in our case convertd-video/)

The screenshot shows the AWS Lambda console for a function named 'conversion'. The 'Function overview' tab is selected, showing a diagram of the function's configuration. An S3 bucket is selected as the trigger, and the function is linked to it. The right sidebar provides details about the function, including its ARN and last modified time. The bottom navigation bar shows various tabs for managing the function.

- now click create

now upload video from wordpress less than 2MB



then check elastic transcoder status, it should be completed. also you can see cloudwatch logs and s3 folder created with converted video.

