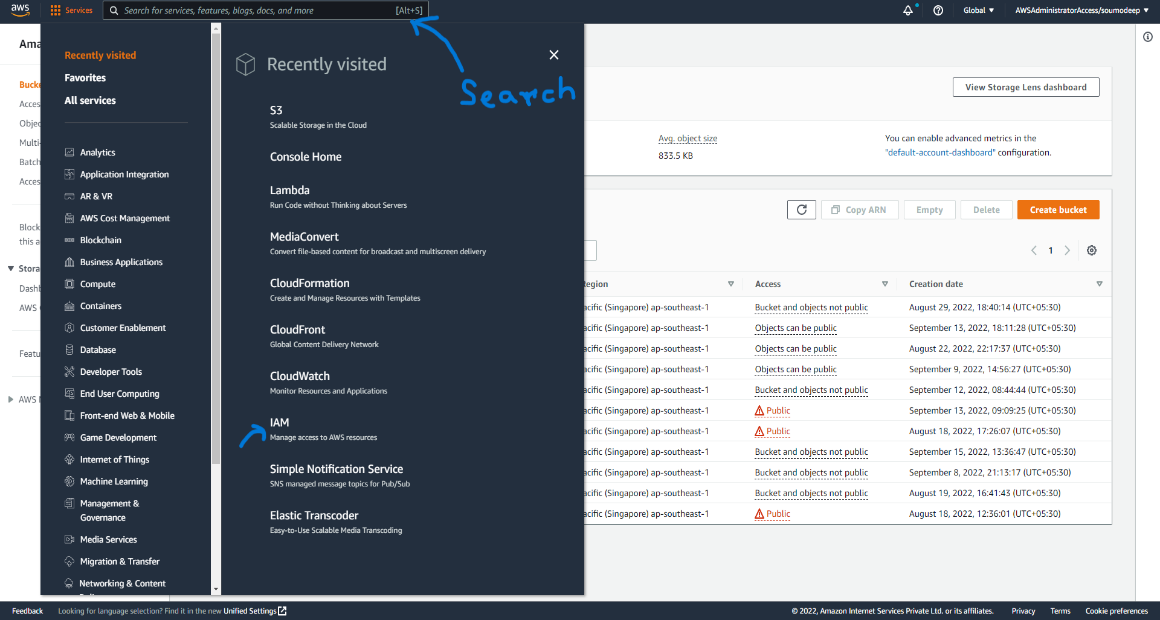
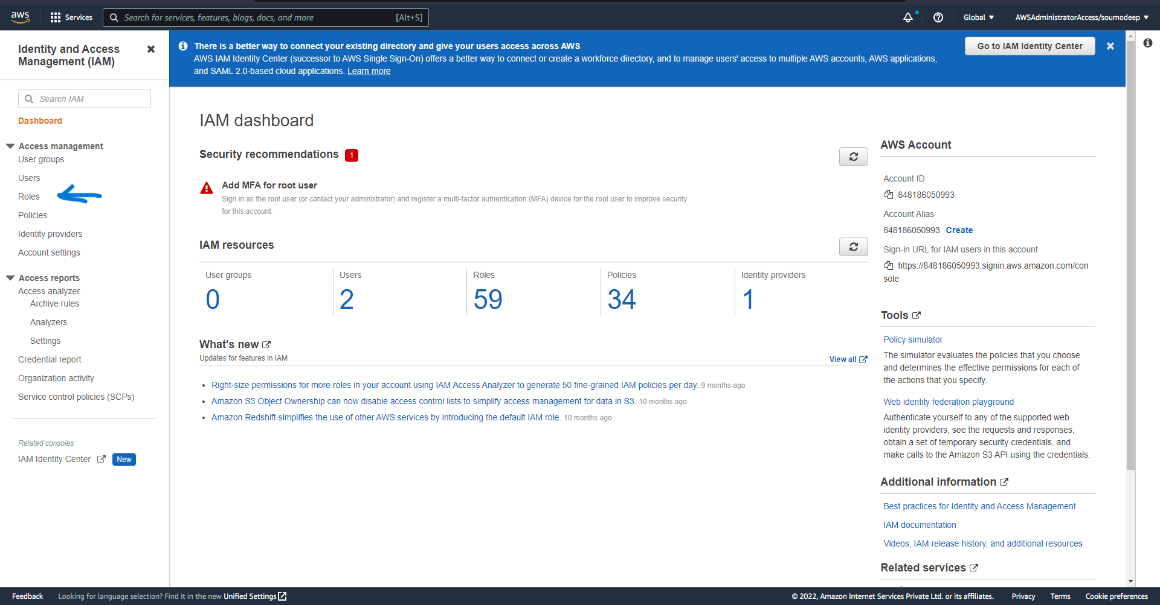
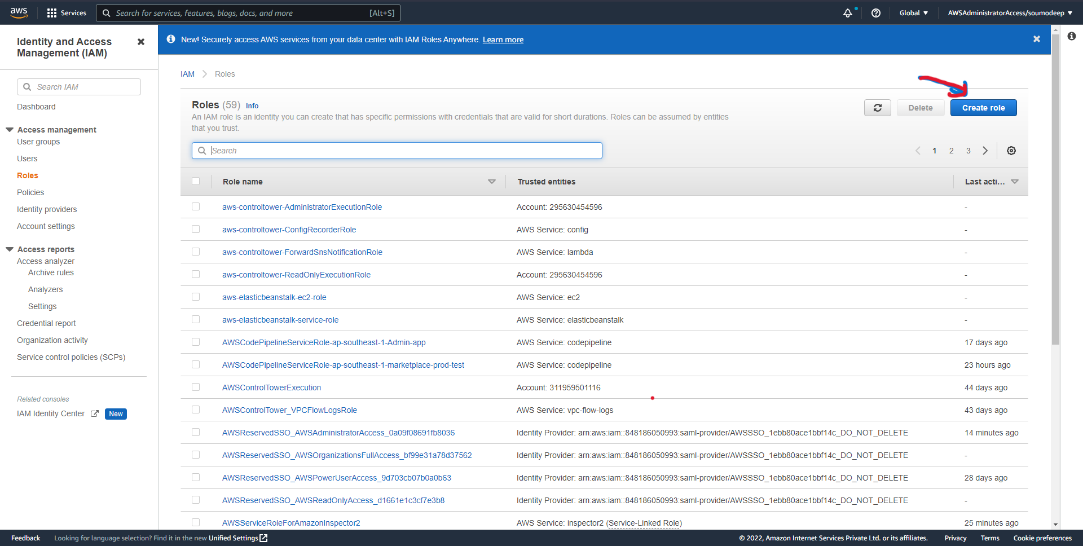
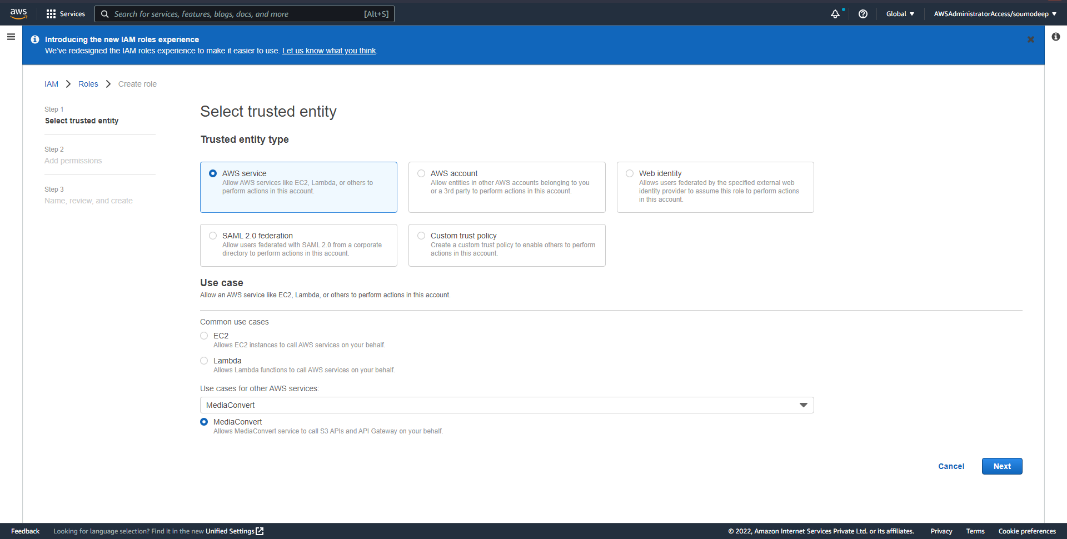
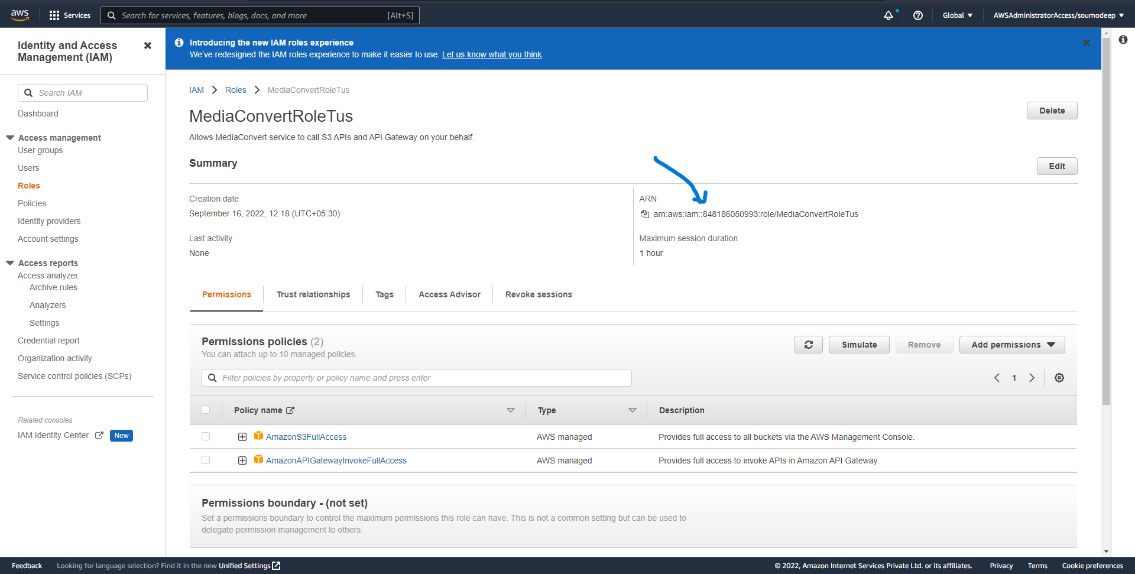
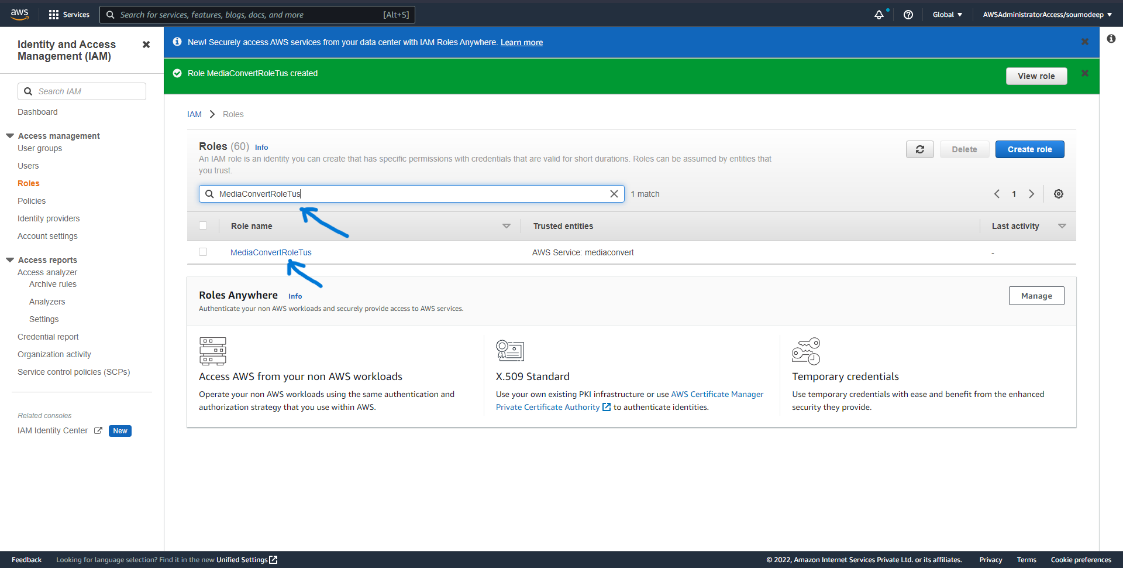
1. **Need a S3 bucket**

If you already have a bucket then no need to create a bucket, use the existing one.

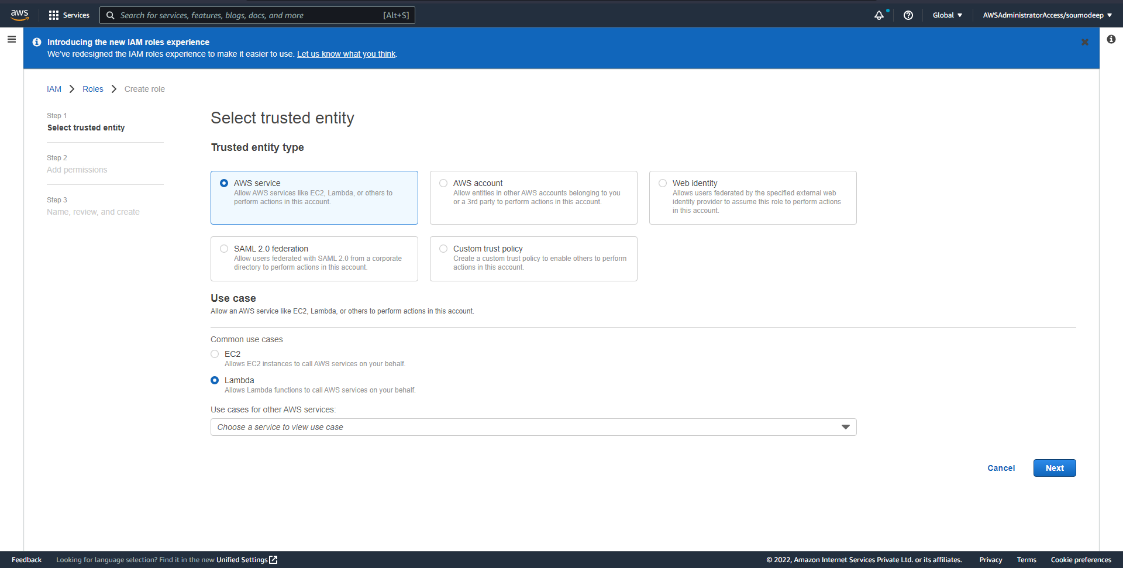
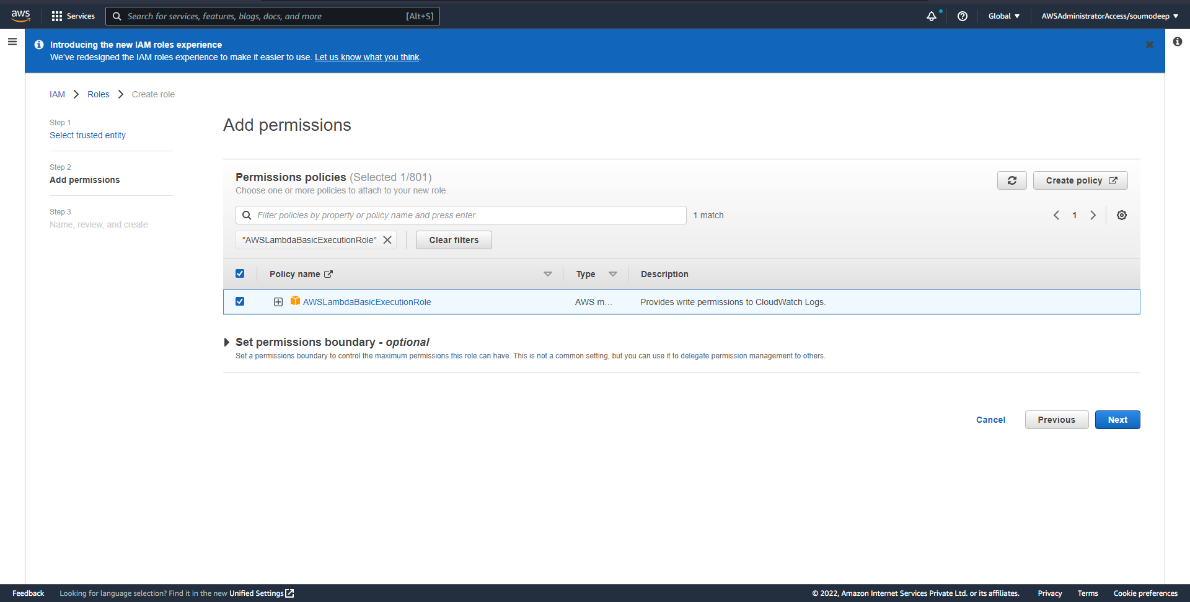
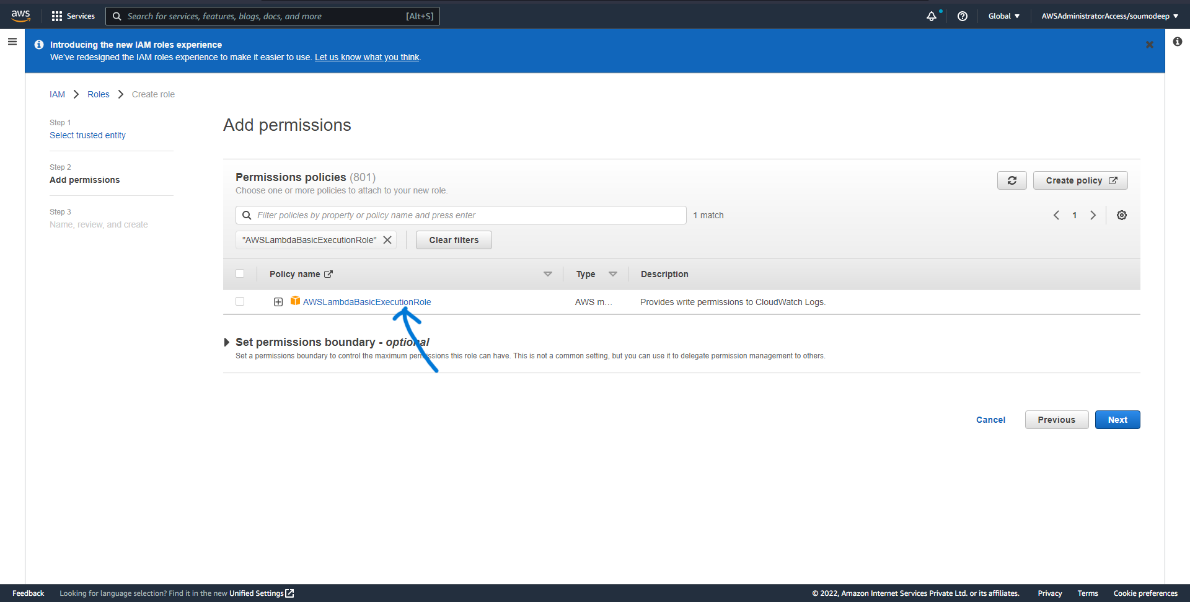
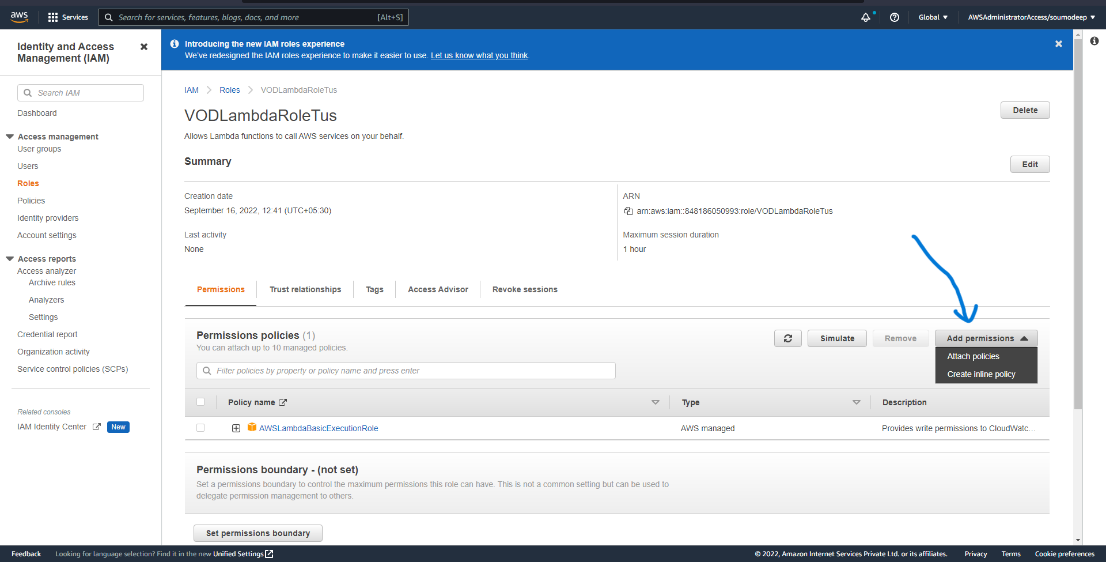
1. **Create an IAM role to pass to MediaConvert**

* Go to Services and select IAM option



* In IAM console you can see on the left side Roles option under Access Management
* Select Roles and then choose Create role
* Select AWS Service as Trusted entity type and select MediaConvert as Use case and click on next
* In Add Permission page again choose next button
* In Name, review and create page enter Role name under Role details and click on create role button
* Search Role name you have just created on click on that and save the ARN for use later

1. **Create and IAM Role for your Lambda function**

* Creating IAM role for Lambda function is same as creating IAM role for MediaConvert. In this case you have select Lambda option and click on next button
* In Add permissions page search AWSLambdaBasicExecutionRole in filter section and click on the checkbox then click on next button
* On Name, review and create page enter the role name under the Role details and click on create role button
* Search the IAM role name you have just created for your Lambda function and click on that
* On the right hand side you can see a drop down button Add permissions. Click on that and select Create inline policy

On the Create policy page select the JSON tab and paste the code and replace the ARNforMdeiaConvertRole with ARN of mdeiaConvertRole then click on the Review policy

{

"Version": "2012-10-17",

"Statement": [

{

"Action": [

"logs:CreateLogGroup",

"logs:CreateLogStream",

"logs:PutLogEvents"

],

"Resource": "\*",

"Effect": "Allow",

"Sid": "Logging"

},

{

"Action": [

"iam:PassRole"

],

"Resource": [

"ARNforMediaConvertRole"

],

"Effect": "Allow",

"Sid": "PassRole"

},

{

"Action": [

"mediaconvert:\*"

],

"Resource": [

"\*"

],

"Effect": "Allow",

"Sid": "MediaConvertService"

},

{

"Action": [

"s3:\*"

],

"Resource": [

"\*"

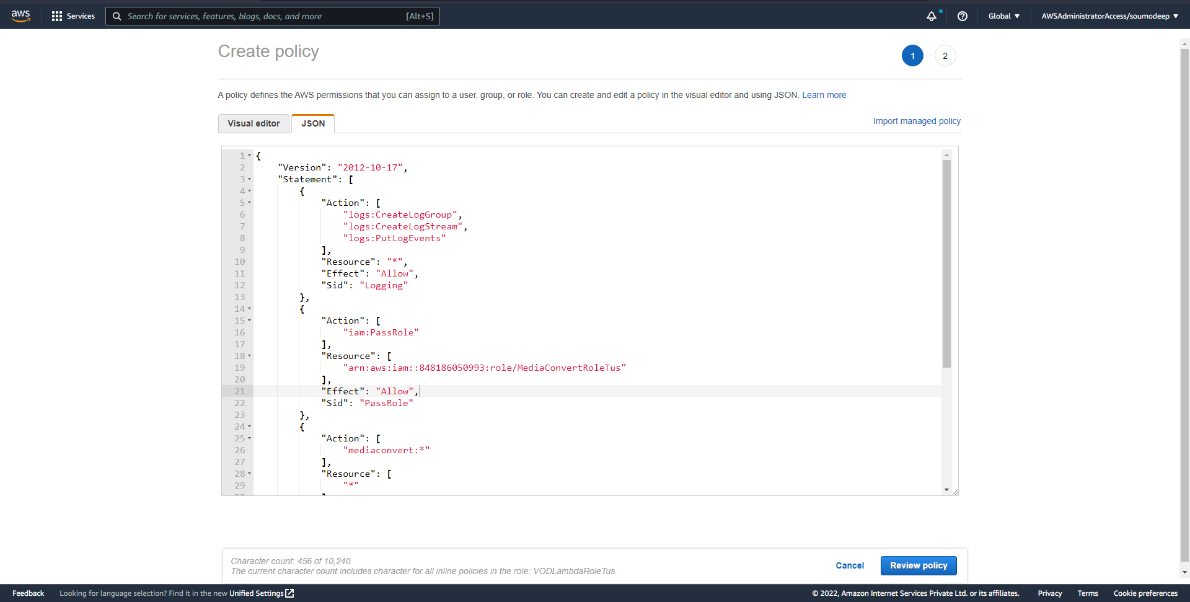
],

"Effect": "Allow",

"Sid": "S3Service"

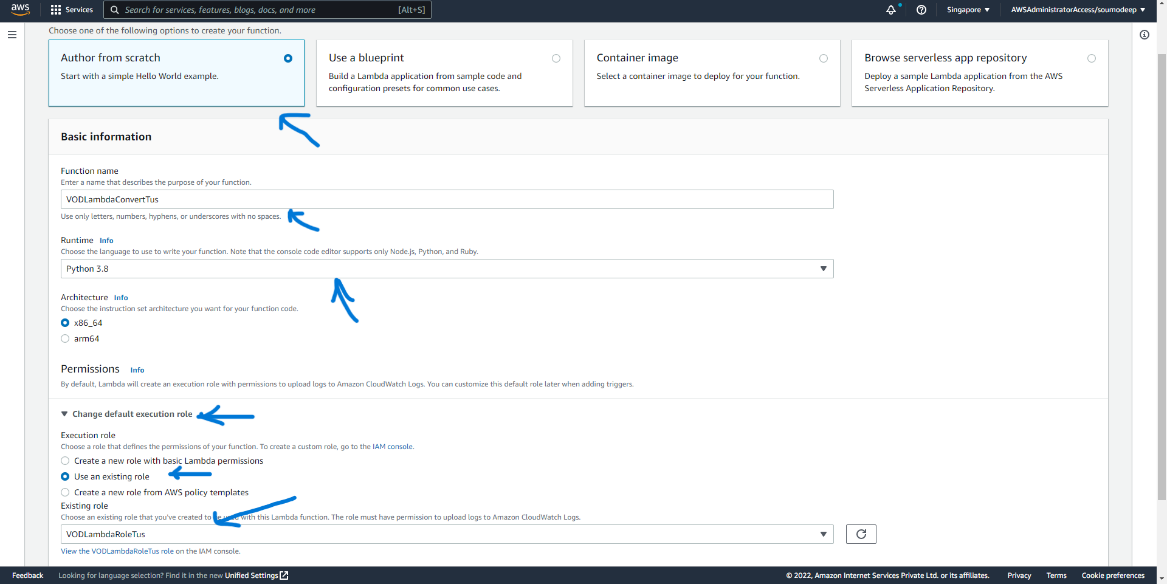
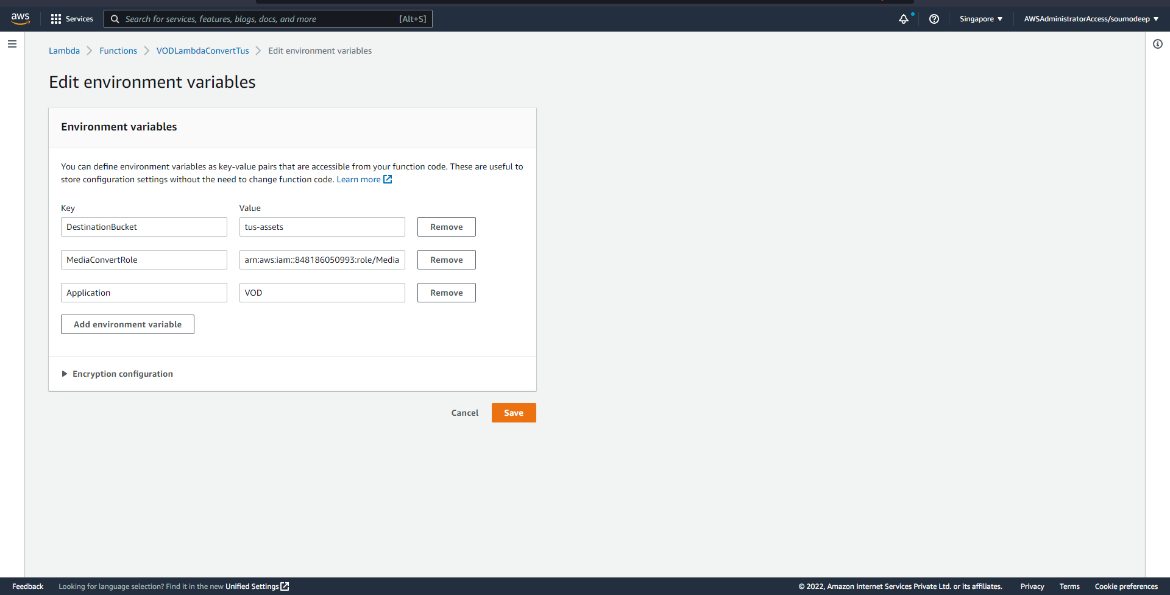
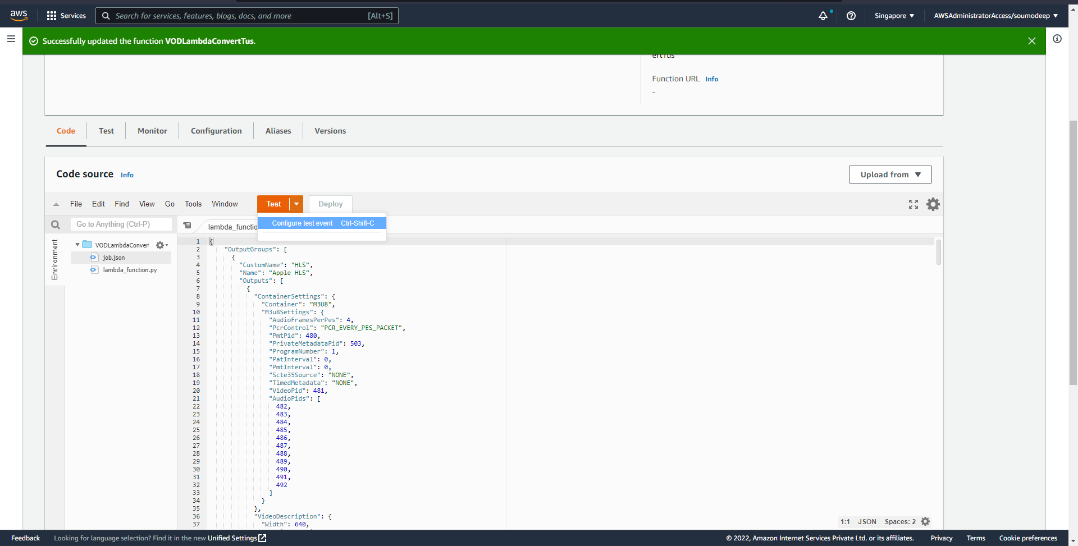
}

]

}

* On Create policy page enter the name of the policy under the Review policy section and then click on create policy button

1. **Create a Lambda function for converting videos**

* Choose Service then select Lambda
* Click on Create function button
* Choose Author from scratch button
* Enter the function name
* Select the Runtime
* Click on Dropdown Change default execution role
* Click on Use an existing role and choose your IAM role for lambda****
* Click on Create function
* On code section in .py file insert the Lambda function written in python
* Create a new file name job.json and insert the jobs configuration in json format
* In configuration tab click on Environment variable option and add environments 
* Click on Save button
* Go to Code tab Click on Test dropdown button and select Configure test event
* Enter the event name and paste the json file [Event JSON](https://github.com/aws-samples/aws-media-services-vod-automation/blob/master/MediaConvert-WorkflowWatchFolderAndNotification/README-tutorial.md#test-the-lambda)
* Replace the bucket name with yours and arn name and region
* Click on save
* Click on Test button and check the output contains status code 200 or not

1. **Create a S3 Event Trigger for Convert Lamdba**

* Click on Add Trigger button
* Select S3 first
* Select all the options and fill the fields according to you
* Click on Add Button