

Developer Manual

For setting up AWS Lambda, Microsoft Video Indexer, Box, and VSCode

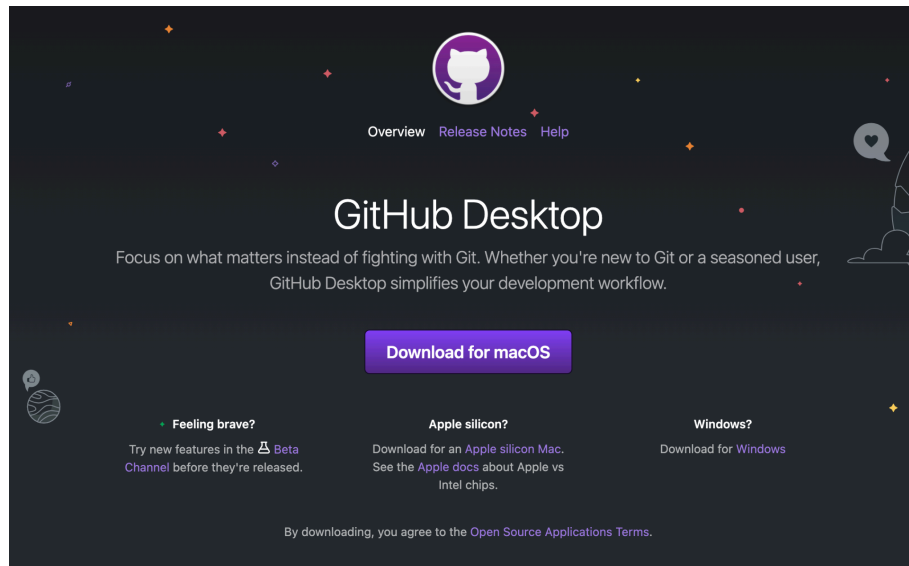
Table of Contents

Table of Contents.....	2
Prerequisites.....	3
GitHub Desktop: Downloading + Setting up your account.....	3
Github Desktop: Creating a Github Branch and working from it.....	4
(!)Github Desktop: Origin, Committing, Pushing.....	6
VSCODE: Configuring VSCode.....	7
VSCODE: Installing NodeJS (in the case that you don't have it installed).....	8
VSCODE: Installing ESLint, Prettier, JavaScript and TypeScript Nightly.....	8
VSCODE: Installing Serverless.....	8
Box.com.....	9
Setting up the Dev Console (from admin account).....	9
Configuring a new Box Skill.....	10
Deploying your Box Skill to the Custom Apps Manager.....	13
<u>Finding</u> Box Keys (Primary and Secondary).....	16
Adding 'config.json' to your VSCode.....	17
Video Indexer.....	18
<u>Finding</u> the Account ID.....	18
<u>Finding</u> the Account Keys.....	19
AWS.....	20
Creating an IAM User.....	20
Creating an Access Key.....	22
Setting Permissions and Adding User to a Group.....	24
Deploying to AWS with Serverless (for the first time).....	28
<u>Finding</u> APIGateway.....	29
Environment Variables Key.....	31
Configuring BOX, S3, and LAMBDA.....	32
Making sure your lambda function and S3 bucket was created.....	32
Configuring Box.....	33
Configuring Lambda.....	34
Post-Deployment Information.....	35
How to test your AWS Lambda Function?.....	35
What do I do if I change the code?.....	37

Prerequisites

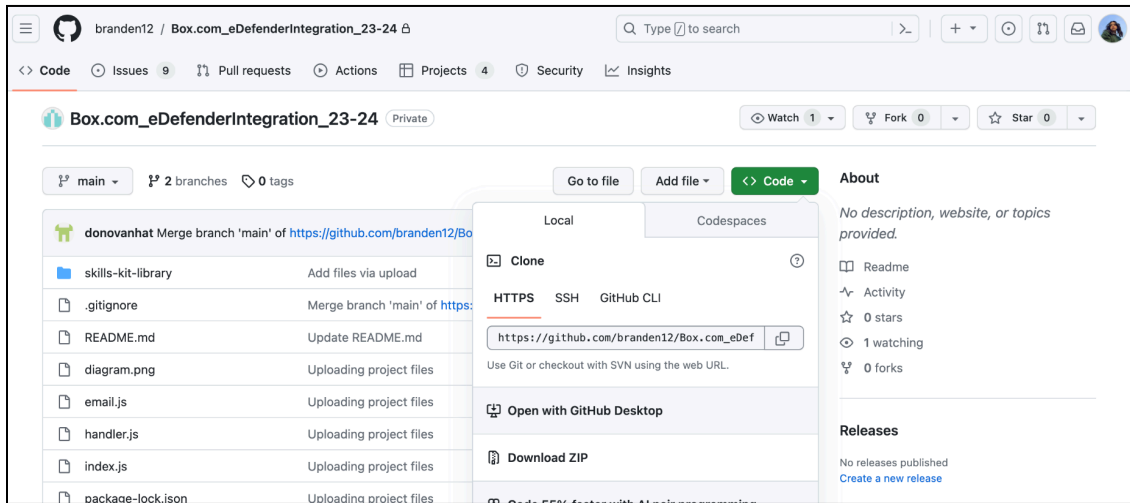
GitHub Desktop: Downloading + Setting up your account

Download Github Desktop and set up your account.

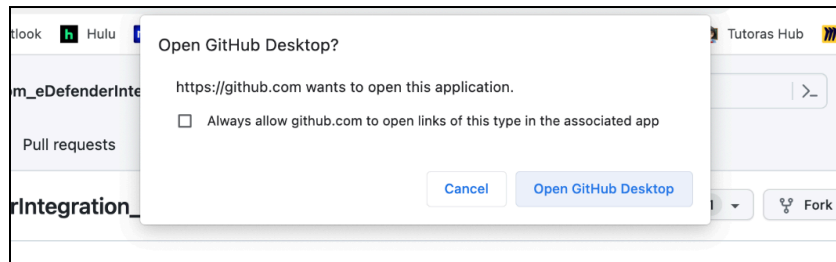


Github Desktop: Creating a Github Branch and working from it

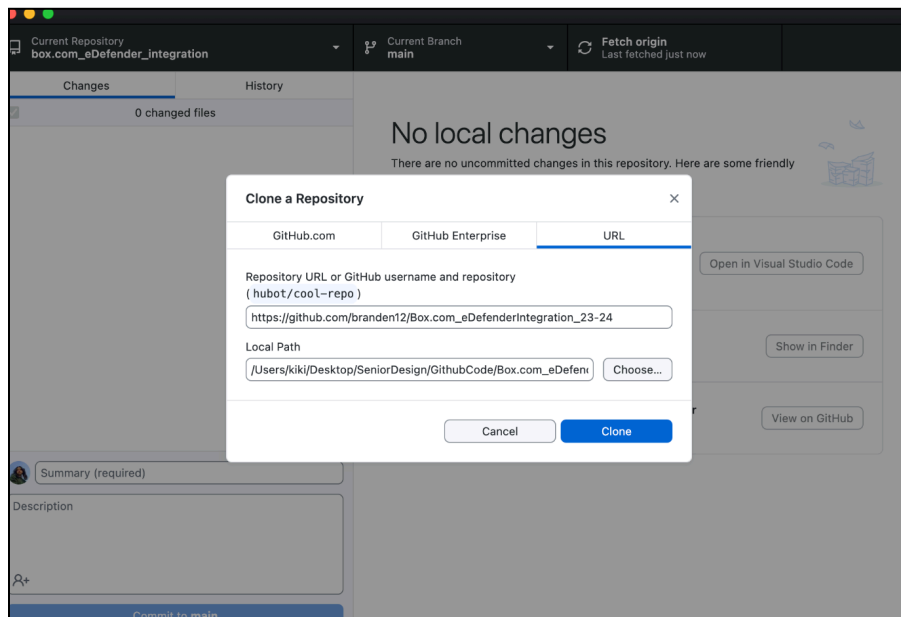
1. Click the green 'Code' button and select 'Open with Github Desktop'



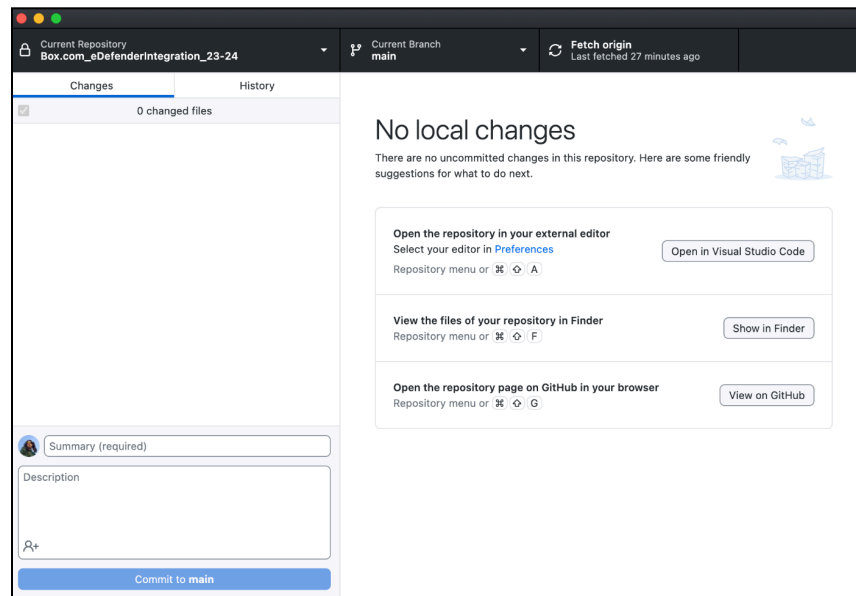
2. Open Github Desktop from the pop-up



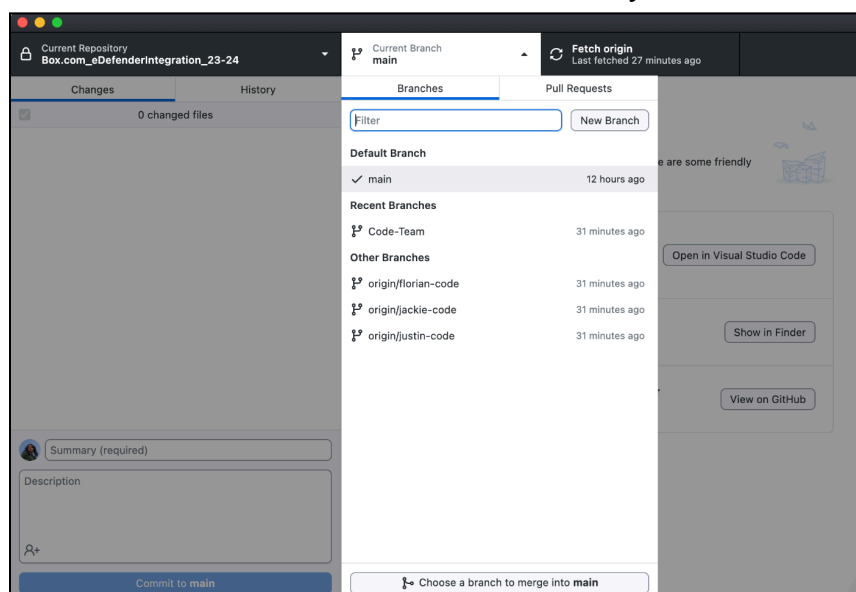
3. Configure where you want the **local path** to be



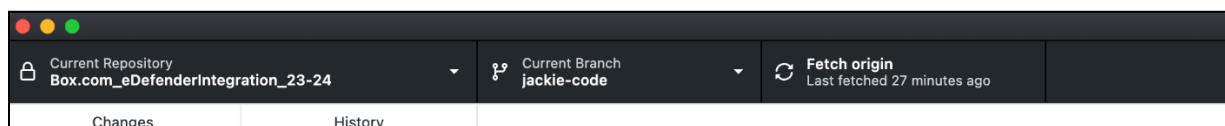
4. When it opens you should see the correct Repository name and the current branch you are in



5. If needed, click on 'Current Branch' and move to the branch you're meant to be on



5. You're done! Now you can open wherever your recently saved local files are and it should automatically make changes here when you edit your files in VSCode!



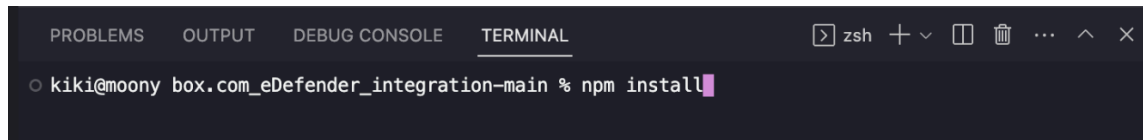
(!)Github Desktop: Origin, Committing, Pushing

If you've never worked with Github Desktop before, these are the most important things to know

1. 'Fetch Origin' - updates any changes made to the branch you're currently on (if any are available)
2. 'Commit' - sort of works like saving your file, these changes are not yet reflected on the Github branch you're working on
3. 'Push' - assures that the changes you make reflect upon the code up on Github

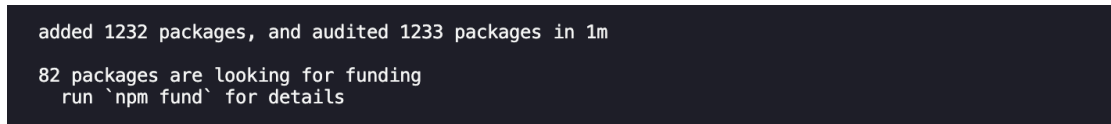
VSCODE: Configuring VSCode

1. Open the files on VSCode (wherever you just stored them)
2. Open a terminal and make sure you're in the correct workspace ('cd' if you have to)
3. Run 'npm install'



A screenshot of a terminal window within a VS Code editor. The terminal title bar shows 'zsh' and standard window controls. The prompt is 'kiki@moony box.com_eDefender_integration-main %' and the command 'npm install' is being typed.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
○ kiki@moony box.com_eDefender_integration-main % npm install
```



A screenshot of the terminal output showing the result of the 'npm install' command. It indicates that 1232 packages were added and audited in 1 minute, and that 82 packages are looking for funding.

```
added 1232 packages, and audited 1233 packages in 1m

82 packages are looking for funding
  run `npm fund` for details
```

VSCODE: Installing NodeJS (in the case that you don't have it installed)

Check if node is installed with the command `Node --version`

1. <https://nodejs.org/en/download>
2. Follow all the steps
3. Open a terminal on your computer and type `Node --version` to check if installed

VSCODE: Installing EsLint, Prettier, JavaScript and TypeScript Nightly

Go to 'Extensions' in VSCode

- Install EsLint
- Install Prettier
- Install JavaScript and TypeScript Nightly

VSCODE: Installing Serverless

<https://www.serverless.com/framework/docs/getting-started>

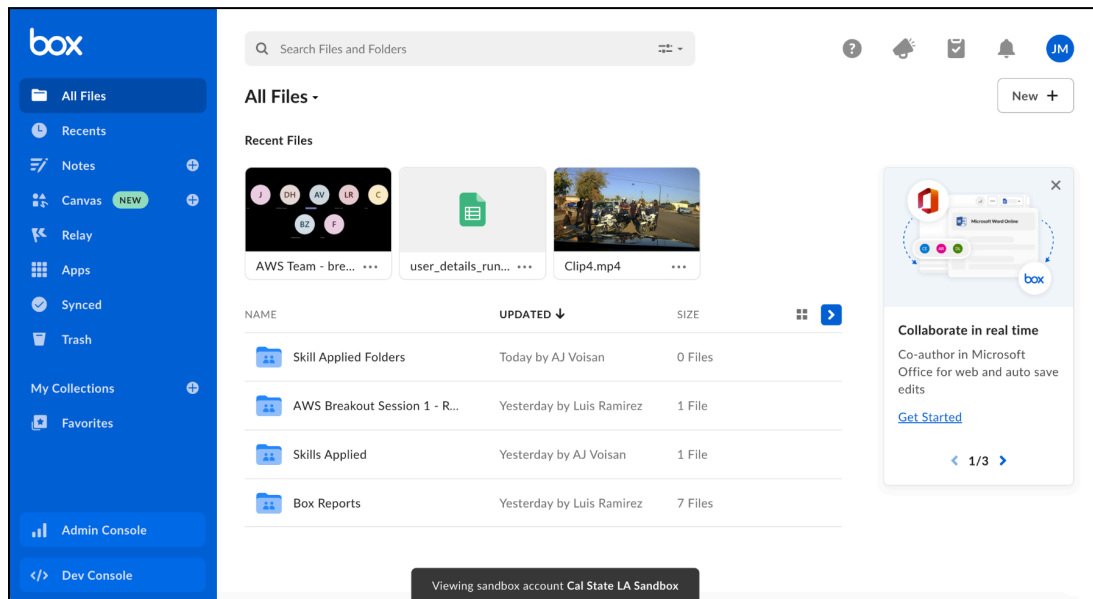
1. Open a terminal in VSCode
2. Type `npm i -g serverless` to install serverless
3. If you get an 'access denied' message type `sudo npm i -g serverless`

Box.com

Setting up the Dev Console (from admin account)

In case you don't see both the Dev Console and Admin console.

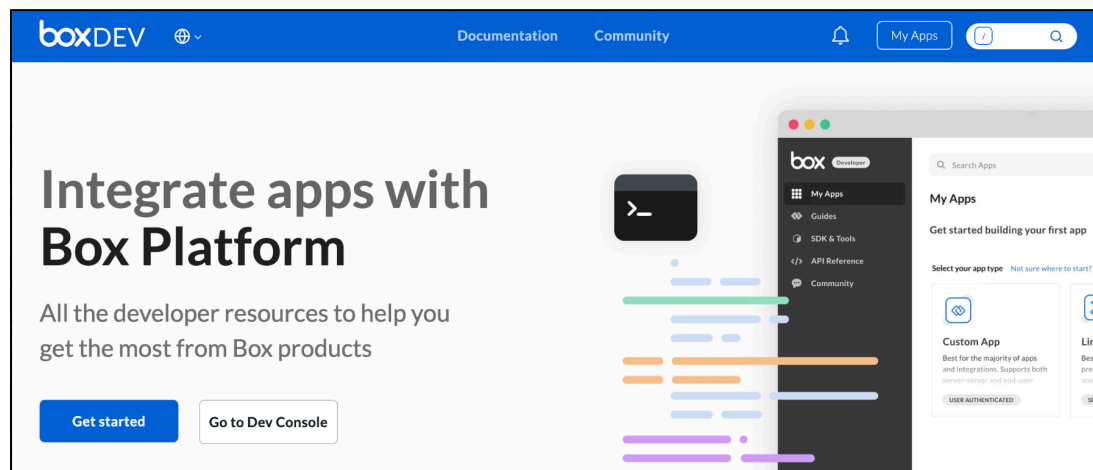
1. Open up your Box account. If it already has both admin and dev consoles. You're done!



2. If you only see the admin console but not the dev console. Go to this link.

<https://developer.box.com/>

3. Click the 'Go to dev console'

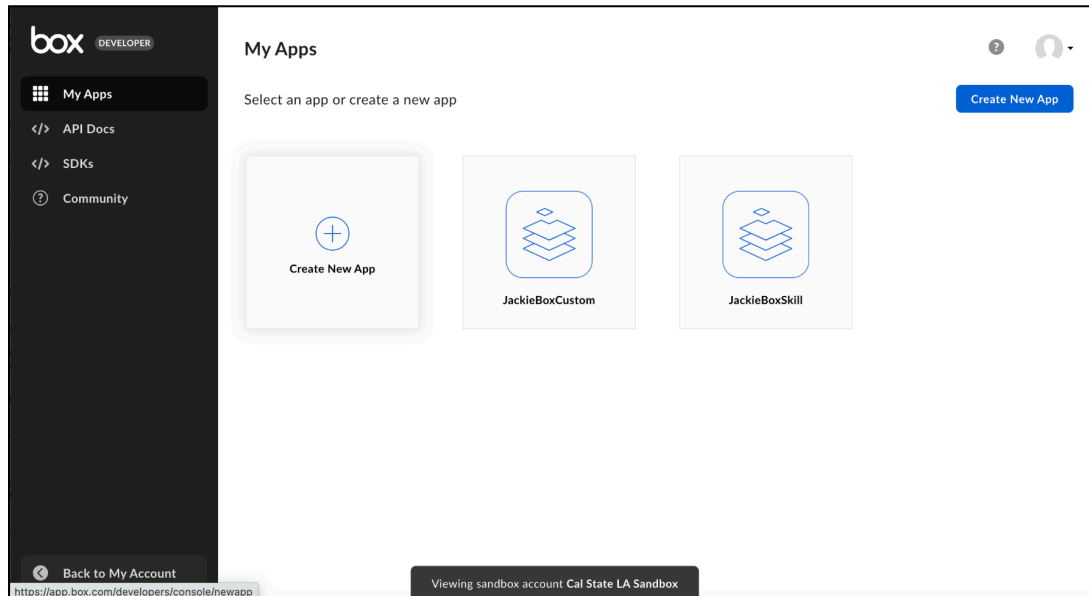


4. And now click 'Go back to my account'. You should see both the admin and dev consoles.

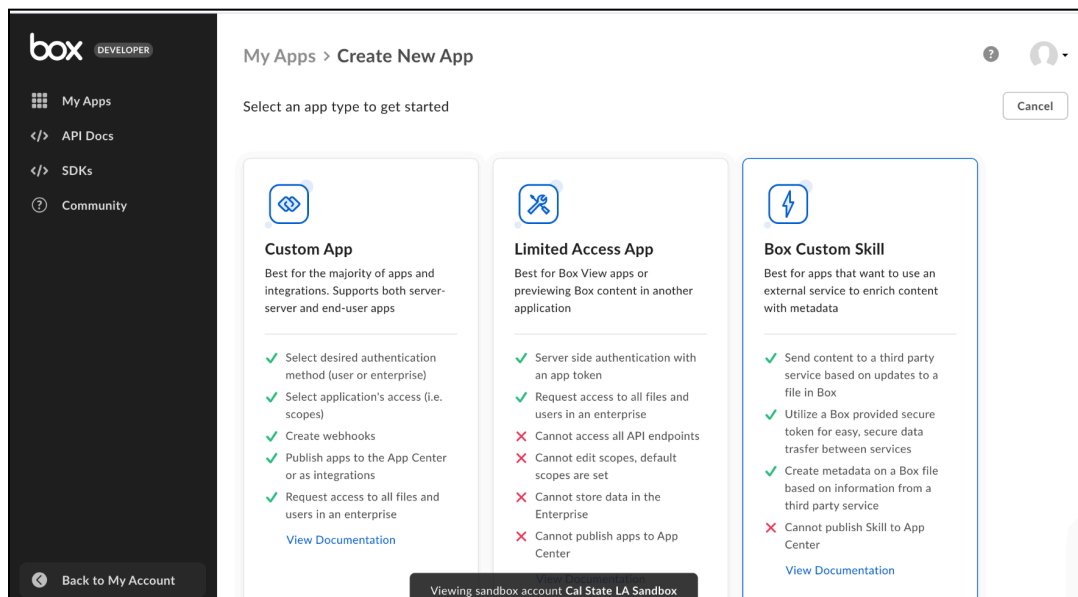
Configuring a new Box Skill

<https://developer.box.com/guides/applications/custom-skills/setup/>

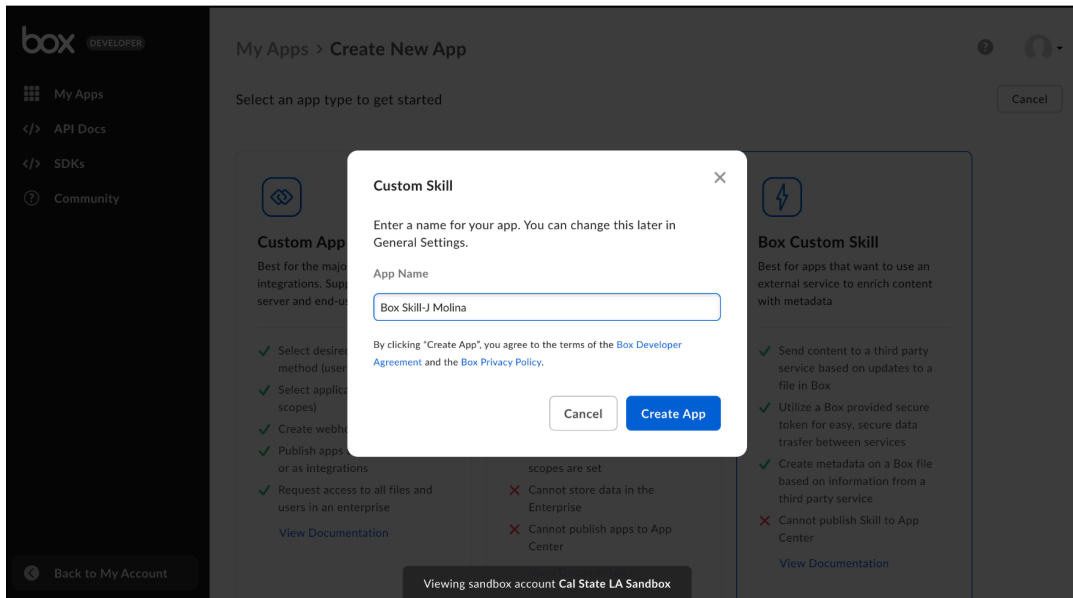
1. Open the 'Dev Console' (should take you to a new tab)



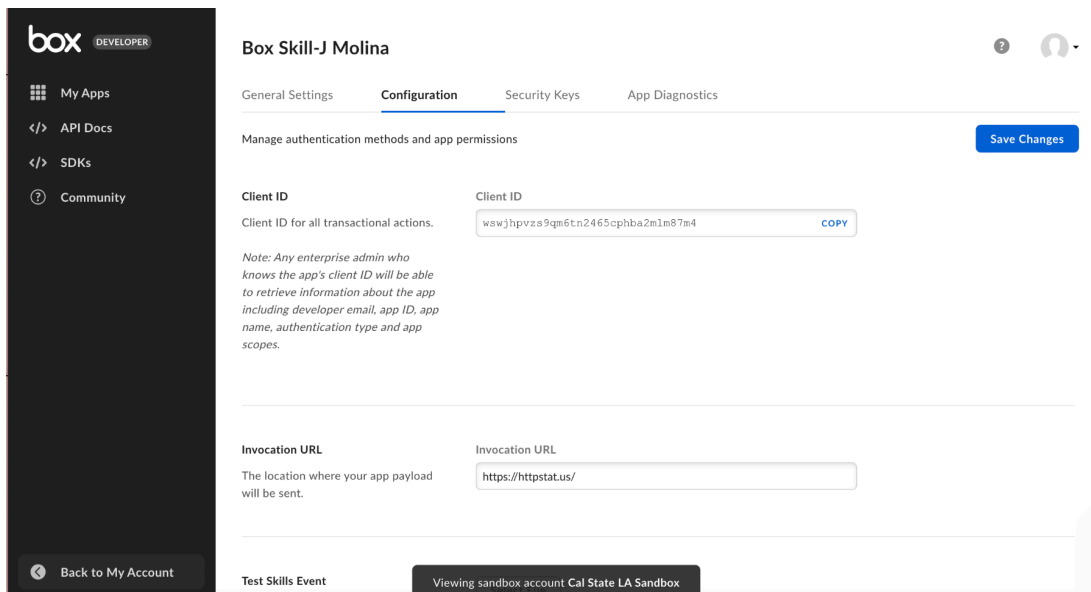
2. Click 'Create New App' and select Box Custom Skill



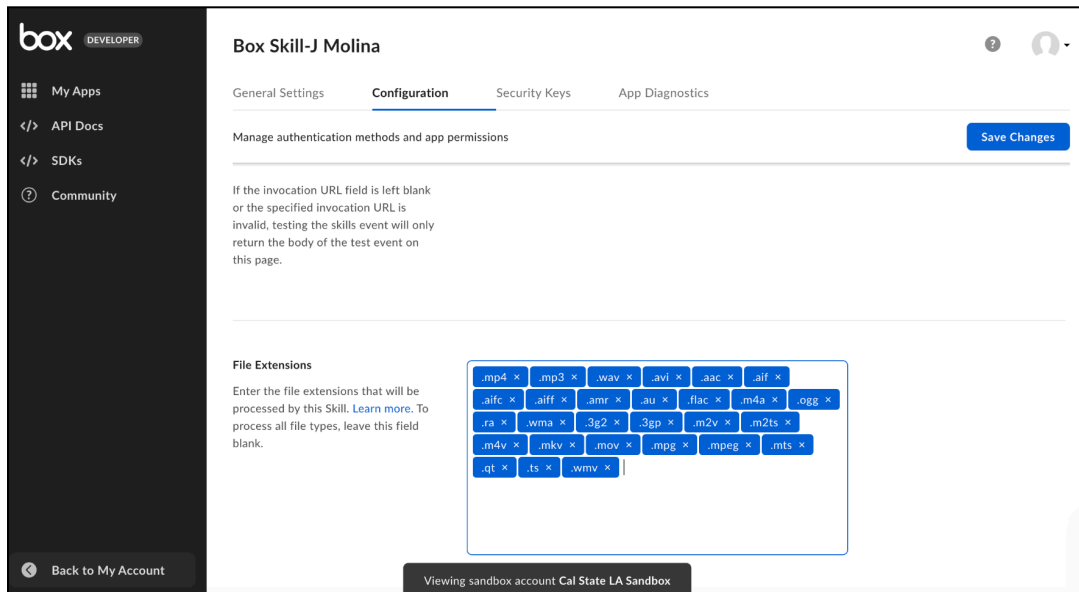
3. Name your Custom Skill and 'Create App'



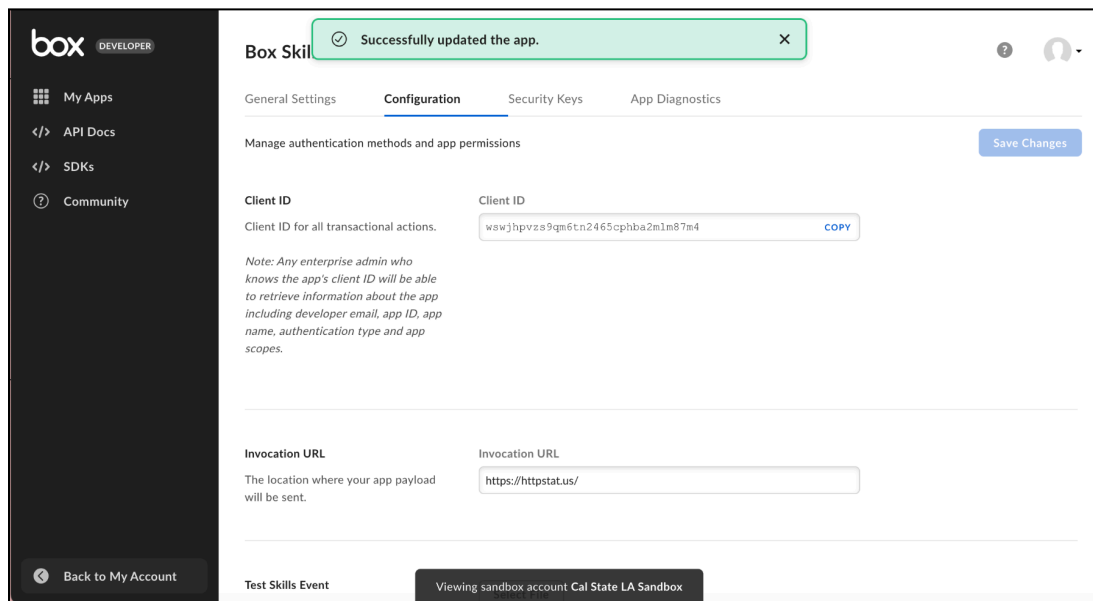
4. Go to the configurations tab in your skill app and add a temporary Invocation URL



5. Scroll down and add these to the File Extensions



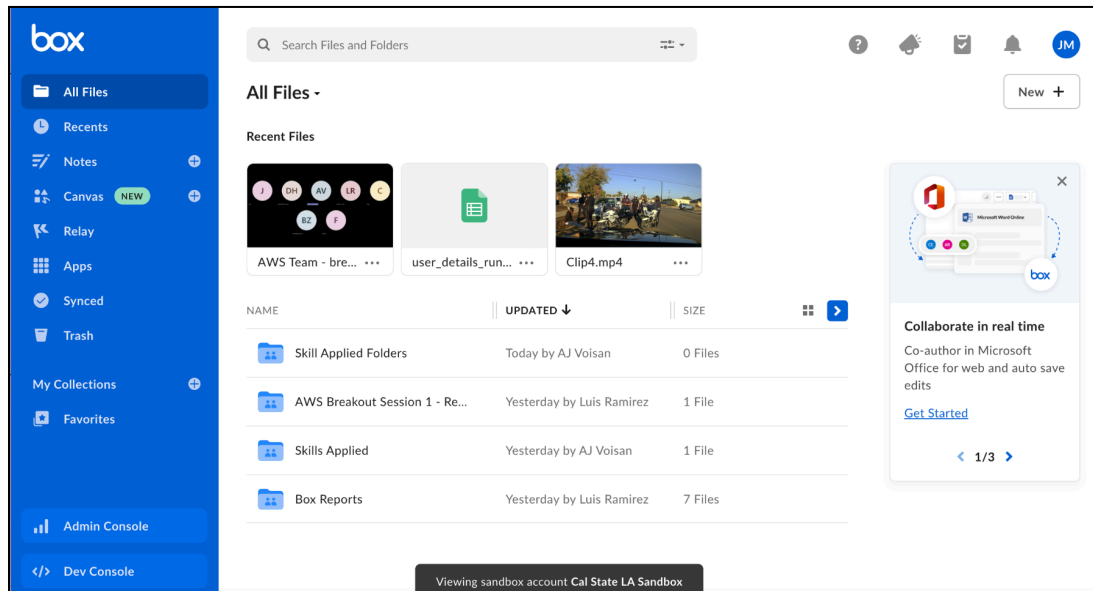
6. Save changes and copy the ClientID (you need this for the next part).



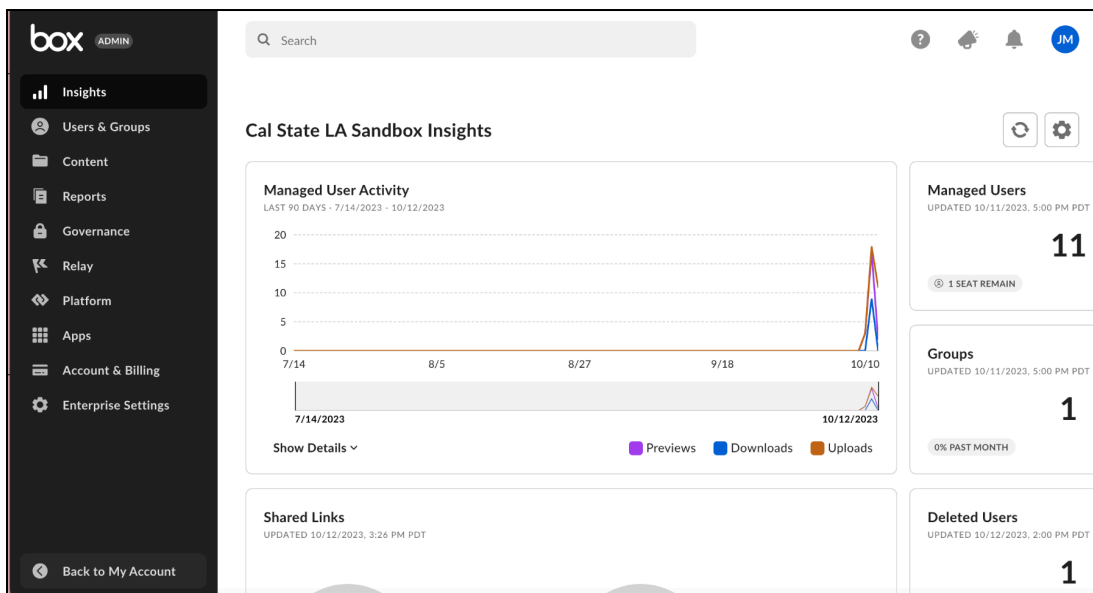
You've created a Custom Box Skill!

Deploying your Box Skill to the Custom Apps Manager

1. From your dashboard, click on the Admin console (you will be redirected)



2. On your Admin console, navigate to the 'Apps' tab



3. Click on the 'Custom Apps Manager' tab

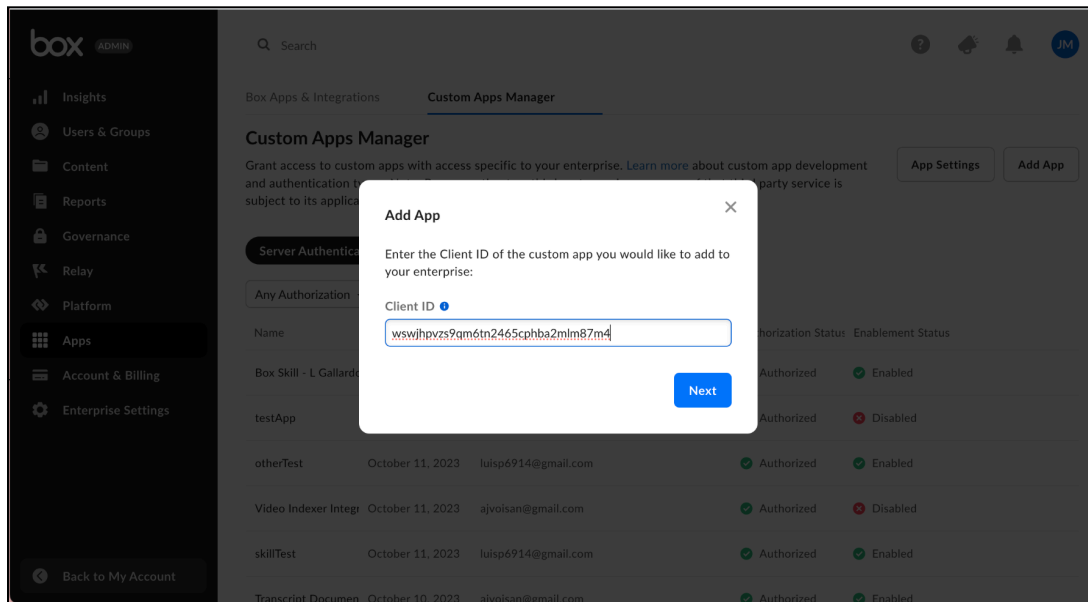
The screenshot shows the Box Admin console interface. On the left is a dark sidebar with navigation links: Insights, Users & Groups, Content, Reports, Governance, Relay, Platform, Apps (highlighted), Account & Billing, and Enterprise Settings. At the bottom of the sidebar is a 'Back to My Account' link. The main content area has a top bar with a search bar and user profile 'JM'. Below this, the 'Box Apps & Integrations' section is active, with a sub-tab 'Custom Apps Manager'. The 'Box Apps & Integrations' header includes a 'Save' button. The content area is divided into two columns. The left column contains text about 'Official Box Apps' and a note about contacting support. The right column features five expandable sections: 'Desktop Apps' (4 of 4 enabled), 'Android Apps' (5 of 5 enabled), 'iOS Apps' (6 of 6 enabled), 'Server Apps' (1 of 1 enabled), and 'Mobile Web and Accessibility' (2 of 2 enabled).

4. Click 'Add App'

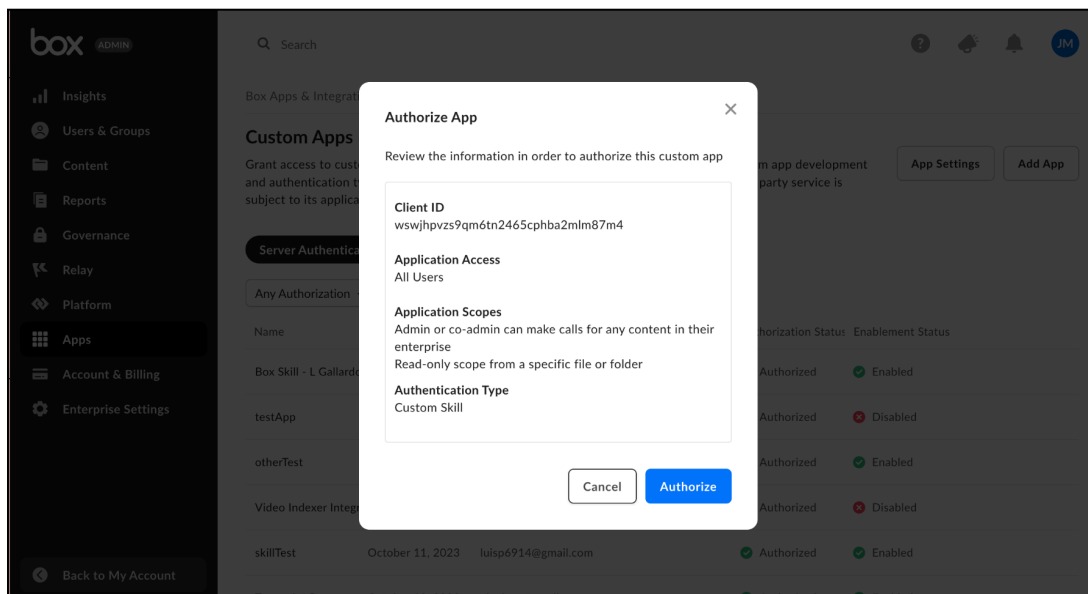
The screenshot shows the Box Admin console interface, specifically the 'Custom Apps Manager' tab. The sidebar is identical to the previous screenshot. The main content area has a top bar with a search bar and user profile 'JM'. Below this, the 'Custom Apps Manager' section is active. It includes a header with 'Add an app using Client ID' and buttons for 'App Settings' and 'Add App'. A note explains that custom apps grant access to your enterprise and that connecting to third-party services is subject to their terms. Below this, there are two tabs: 'Server Authentication Apps' (selected) and 'User Authentication Apps'. A dropdown menu shows 'Any Authorization'. A table lists installed apps with columns for Name, Last Activity, Developer Email, Authorization Status, and Enablement Status.

Name	Last Activity	Developer Email	Authorization Status	Enablement Status
Box Skill - L Gallard	October 12, 2023	leo.gallardo2002@gmail.com	✓ Authorized	✓ Enabled
testApp	October 11, 2023	jmolin84@calstatela.edu	✓ Authorized	✗ Disabled
otherTest	October 11, 2023	luisp6914@gmail.com	✓ Authorized	✓ Enabled
Video Indexer Integr	October 11, 2023	ajvoisan@gmail.com	✓ Authorized	✗ Disabled
skillTest	October 11, 2023	luisp6914@gmail.com	✓ Authorized	✓ Enabled
Transcript Documen	October 10, 2023	ajvoisan@gmail.com	✓ Authorized	✓ Enabled

5. Paste your Client ID and click Next

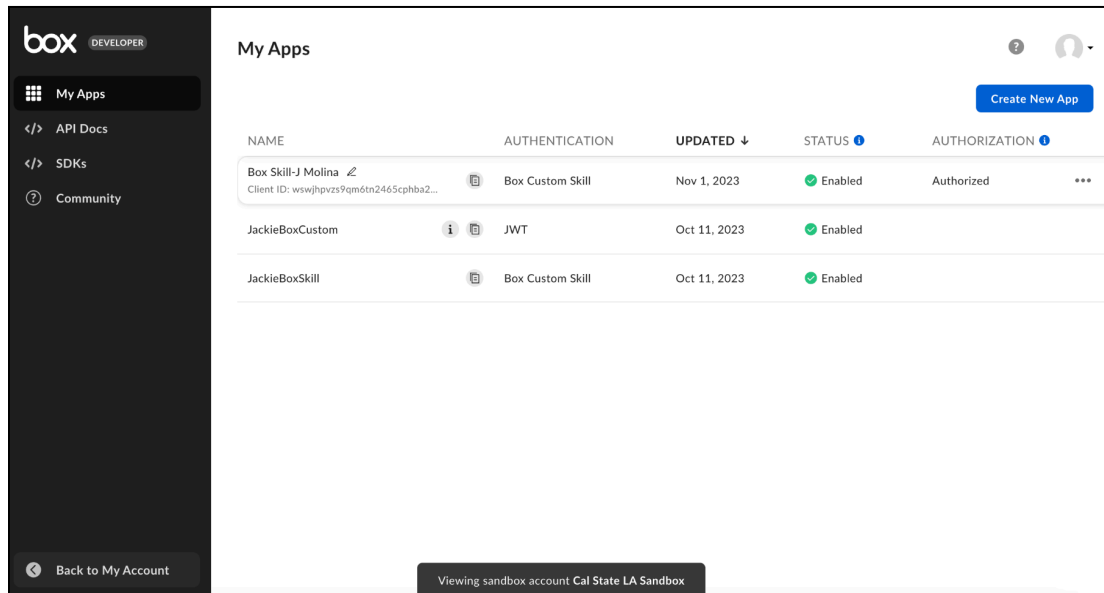


6. Authorize and you're done!

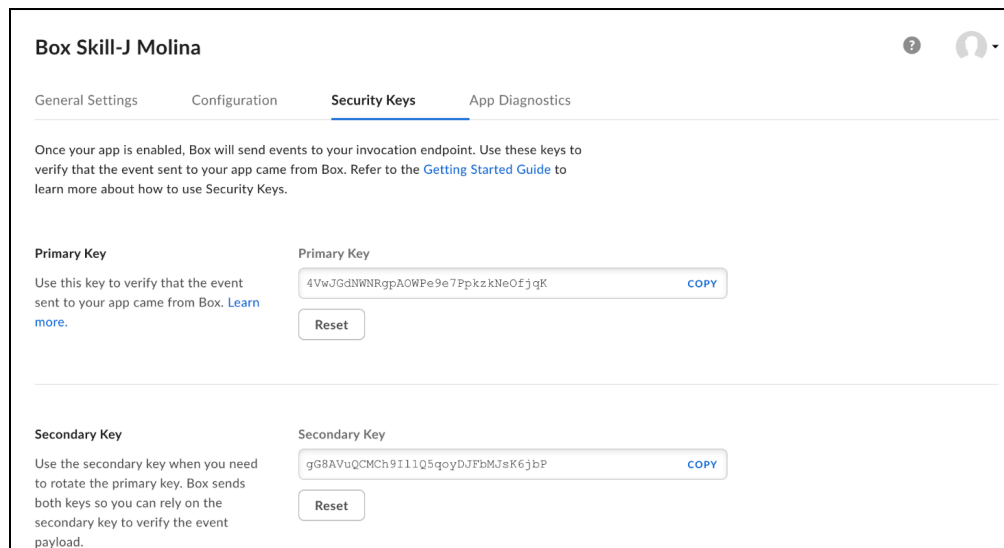


Finding Box Keys (Primary and Secondary)

1.Go to the Developer Portal and Click on Your Skill



2.Go to the Security Tab



You've found your Security Keys!

Adding 'config.json' to your VSCode

(You need this file or else your entire program won't run).

1. Create a new file and name it 'config.json'
2. The information inside of the config file will be given to you. Make sure you copy and paste all information given to you into your config.json file.

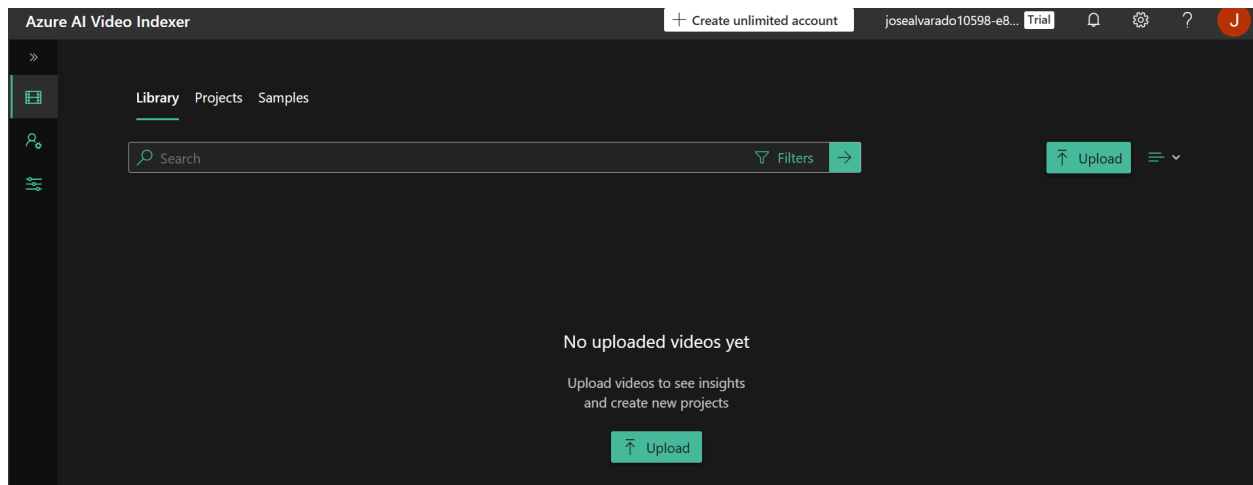
You've added your config file!

Video Indexer

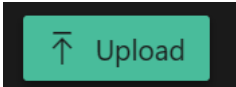
Azure Video Indexer Guide

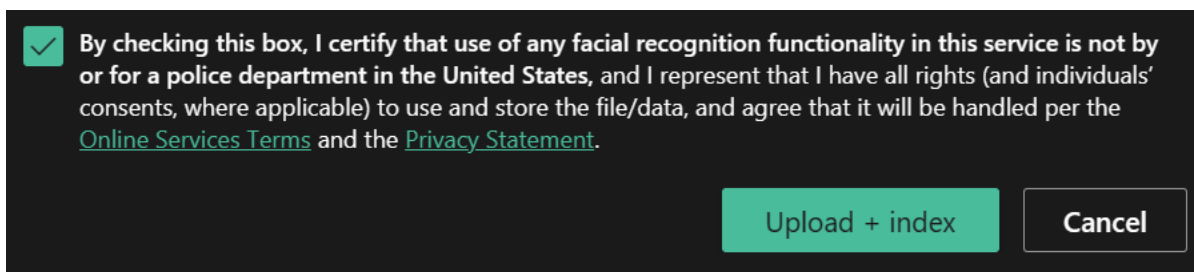
1. Create a Video Indexer Account

- i. Go to [Microsoft Azure AI Video Indexer - Unlock Video Insights](#), and sign up with your designated email
- ii. Your indexer portal should look like this after



2. Uploading videos Directly to Azure Indexer

- i. Click on  and look for the file that you want to upload.
 - i. If you're having trouble uploading you might want to drag the file onto the screen
- ii. Click on the checkbox and "Upload + index"

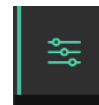


- iii. It should now be uploading to your Indexer Library

	Name	Indexed by	Last modified
<div><div>88%</div></div>	Sergio_NicolasCarno	Jose Alvarado	3 seconds ago
<div><div></div></div>	230703_0107	Jose Alvarado	3 minutes ago

3. Creating a Model

- On the left tab click on Model customization
- Then click on Language(Classic)



Speech (new)

Language (classic)

Language ID

People

Brands

Language

Teach Video Indexer to recognize specific vocabulary or i
Upload a .txt, .utt, .ttml, or .srt file up to 500 KB, and Vide
when videos are uploaded and indexed. [Learn more](#)

Search language models...

English 0 trained models

+

 Create new model

Arabic 0 trained models

+

 Create new model

- Under the English model click on

+

 Create new model and name it. You have now created a model

English 0 trained models

SBPDCustomModel

+

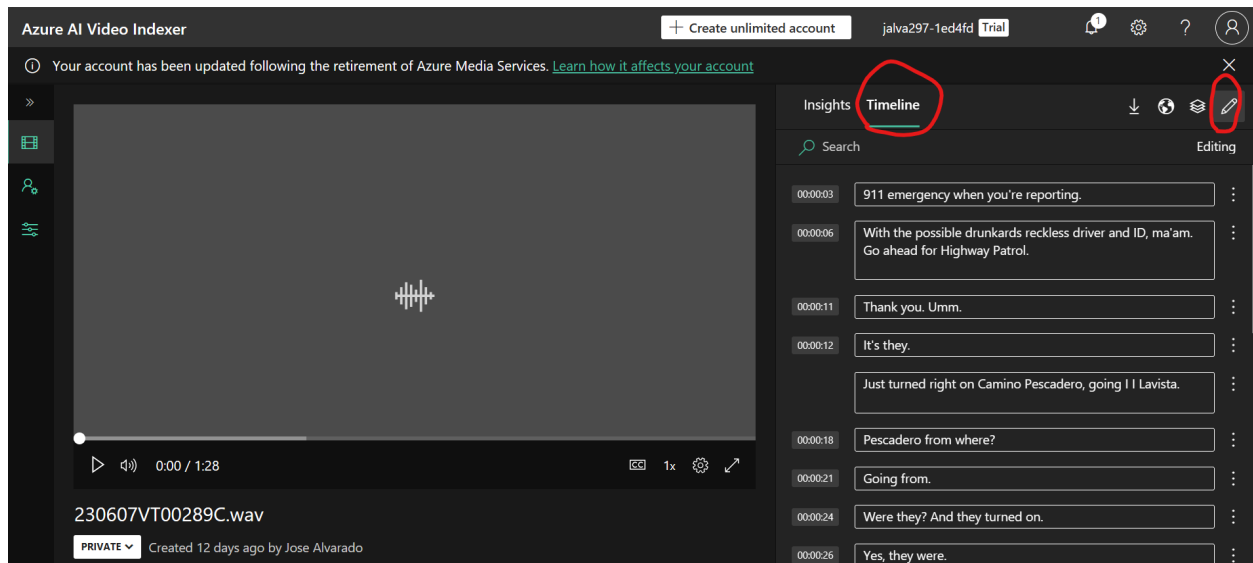
 Add file

+

 Create new model

4. Training a Model

- i. There are two ways to train a model.
 - i. Make changes to a video and then train the model
 - ii. Upload a file with your changes to the model
- ii. Go to your Indexer library and click on a video and you should see a video playing along with the insights
- iii. To edit, click on timeline and the editing icon on the top right



- iv. This will allow you to make any changes to the captions in the video that will be used to train the model.
- v. If you made changes you will have to go back to your model and click on the "Train" icon that appears next to the model



5. Using your model when uploading a video

- i. To use your model while uploading a video, simply choose it under the “Video Source language” before uploading.

Upload and index

① Trial accounts have limited access to face models. [Create an account with an Azure subscription](#)

File name

Sergio Cano

Add files

Privacy

Private

Streaming quality

Single bitrate

Video source language

English (SBPDCustomModel)

English

English (Australia)

English (SBPDCustomModel)

English (United Kingdom)

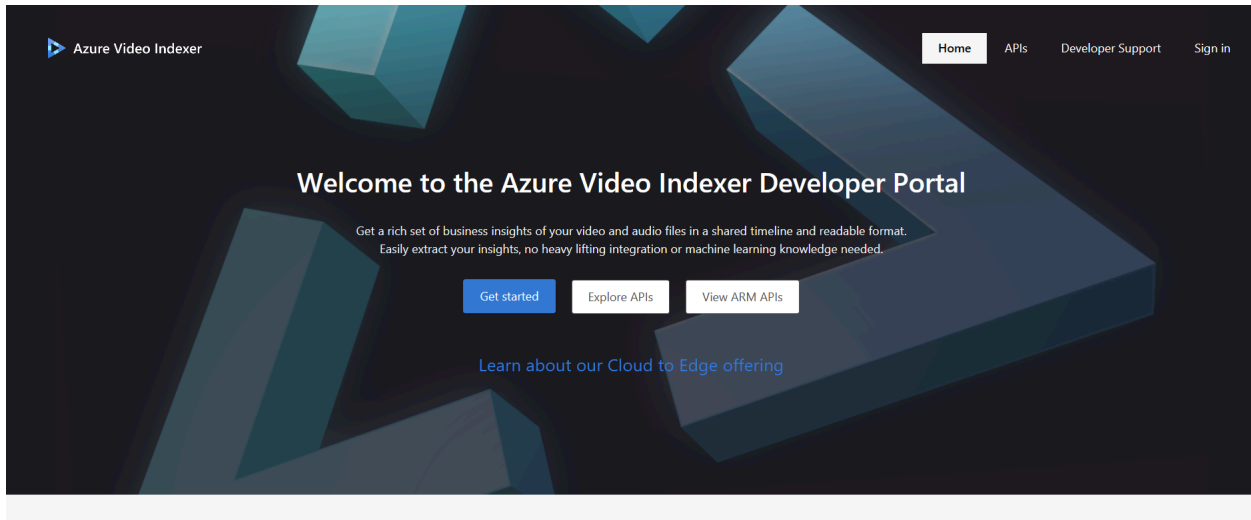
☒ By checking this box, I certify that use of any facial recognition functionality in this service is not by or for a police department in the United States and I represent that I have all rights (and individuals' consents, where applicable) to use and store the file/data, and agree that it will be handled per the [Online Services Terms](#) and the [Privacy Statement](#).

Upload + index

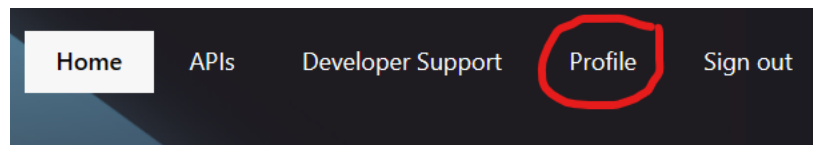
Cancel

6. Retrieving Subscription Keys

- i. Go to [Home - Microsoft Azure API Management - developer portal \(videoindexer.ai\)](https://home.microsoft.com/en-us/product/azure-ai-api) and sign in



- ii. Once you sign in a new tab 'Profile' should appear. Click 'Profile'.



- iii. This should take you to a new page that shows you two subscription keys. 'Primary key' and 'secondary key'.

Account details

Email	jalva297@calstatela.edu
First name	Jose
Last name	Alvarado
Registration date	10/27/2023

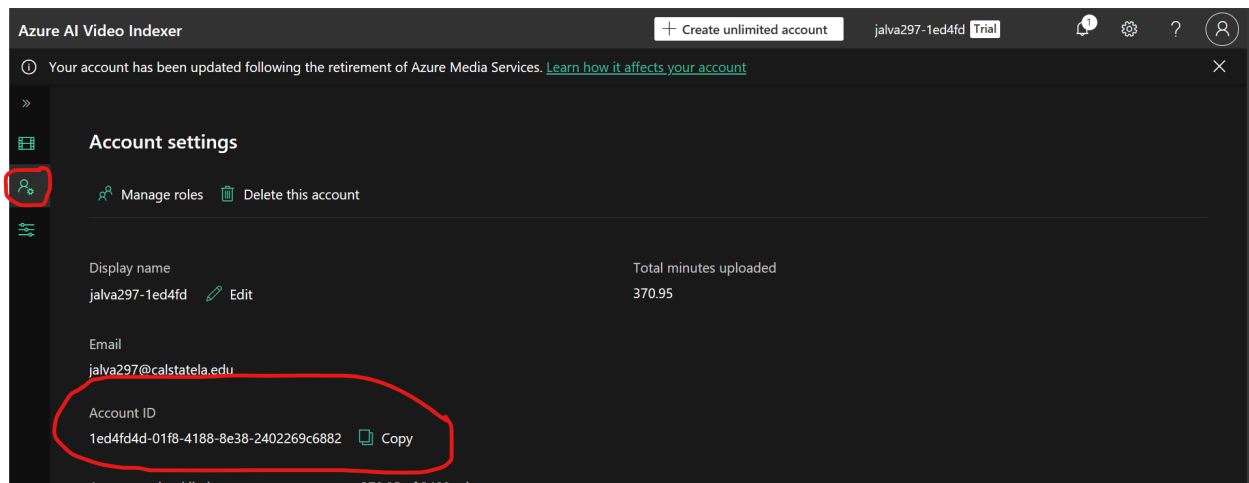
[Close account](#)

Subscriptions

Subscription details			Product	State	Action
Name	Product Authorization subscription	Rename	Authorization	Active	Cancel
Started on	10/27/2023				
Primary key	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	Show Regenerate			
Secondary key	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	Show Regenerate			

7. Retrieving VI Account ID


- i. Choose account details from your VI portal



- ii. You should be given an option to copy your VI account ID

8. Retrieving the Linguistic Model ID

- i. Go into your custom models

- ii. Click on the  icon in the Model whose linguistic Id your want to retrieve

»

Speech (new)

Language (classic)

Language ID

People

Brands

Teach Video Indexer to recognize specific vocabulary or industry terms. Upload a .txt, .utt, .html, or .srt file up to 500 KB, and Video Indexer will apply the new language models when videos are uploaded and indexed. [Learn more](#)

Search language models...

English 1 trained models

SBPDCustomModel		✓ Trained
ExpectedModel	Rename	
Baseline	Delete	
+ Add file	+ Add file	
⊕ Create new model	Linguistic model ID (trained model): 5287cc02-c0ed-4585-9c70- 18489382ad58	

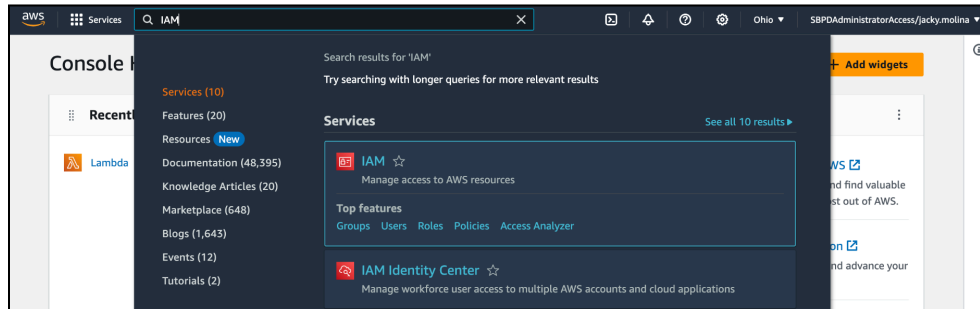
Arabic 0 trained models

iii. Linguistic Id should be visible afterwards

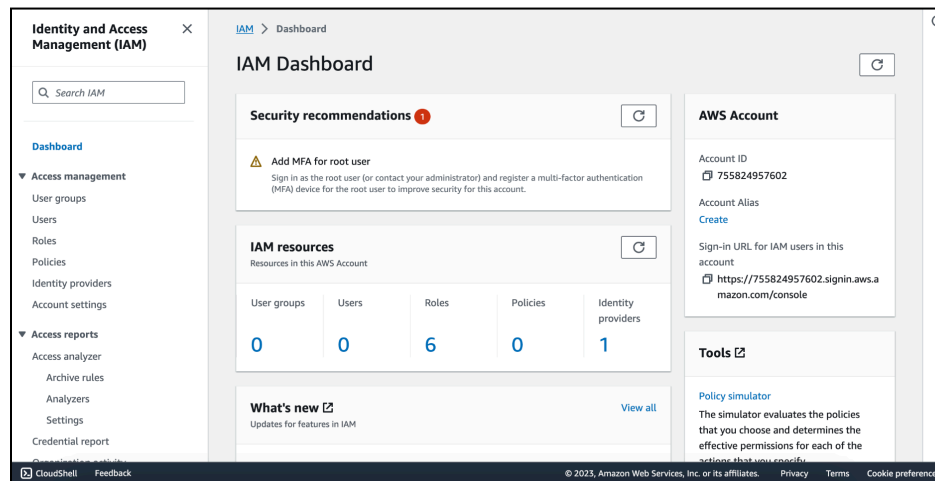
AWS

Creating an IAM User

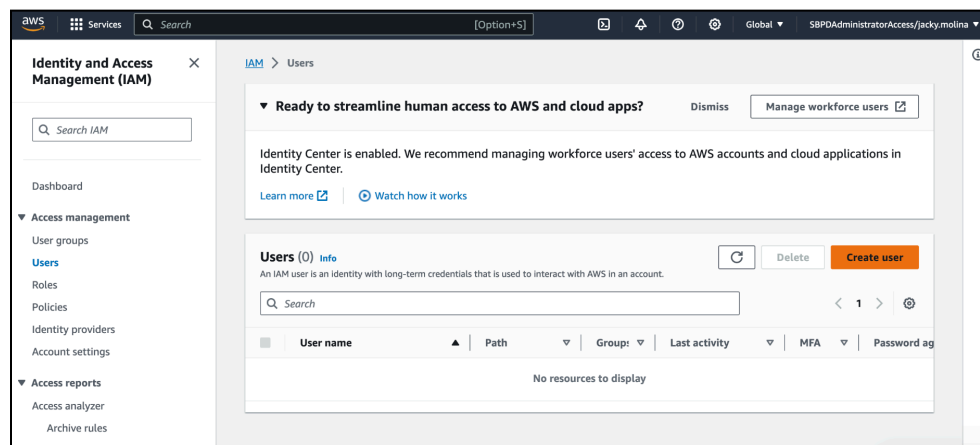
1. Search up the IAM Service



2. This will take you to the IAM dashboard



3. Select the Users tab from the left navigation menu



4. Select 'Create User' (a. fill out user name, b. select 'I want to create an IAM user', c. create custom password, d. select 'Users must create a new password at next sign in')

Step 3
Review and create

Step 4
Retrieve password

User name
TestingSBPD
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , _ . - (hyphen)

☒ Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

Are you providing console access to a person?

User type

☐ Specify a user in Identity Center - Recommended
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

☒ I want to create an IAM user
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keyspaces, or a backup credential for emergency account access.

Console password

☐ Autogenerated password
You can view the password after you create the user.

☒ Custom password
Enter a custom password for the user.
Seventeen19!
• Must be at least 8 characters long
• Must include at least three of the following mix of character types: uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), and symbols ! @ # \$ % ^ & * () _ + - (hyphen) = { } [] ' "

☒ Show password

☒ Users must create a new password at next sign-in - Recommended
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

5. You'll get your username and password

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Step 4
Retrieve password

User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console. [View user](#)

Retrieve password
You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details [Email sign-in instructions](#)

Console sign-in URL
<https://755824957602.signin.aws.amazon.com/console>

User name
TestingSBPD

Console password
Seventeen19! [Hide](#)

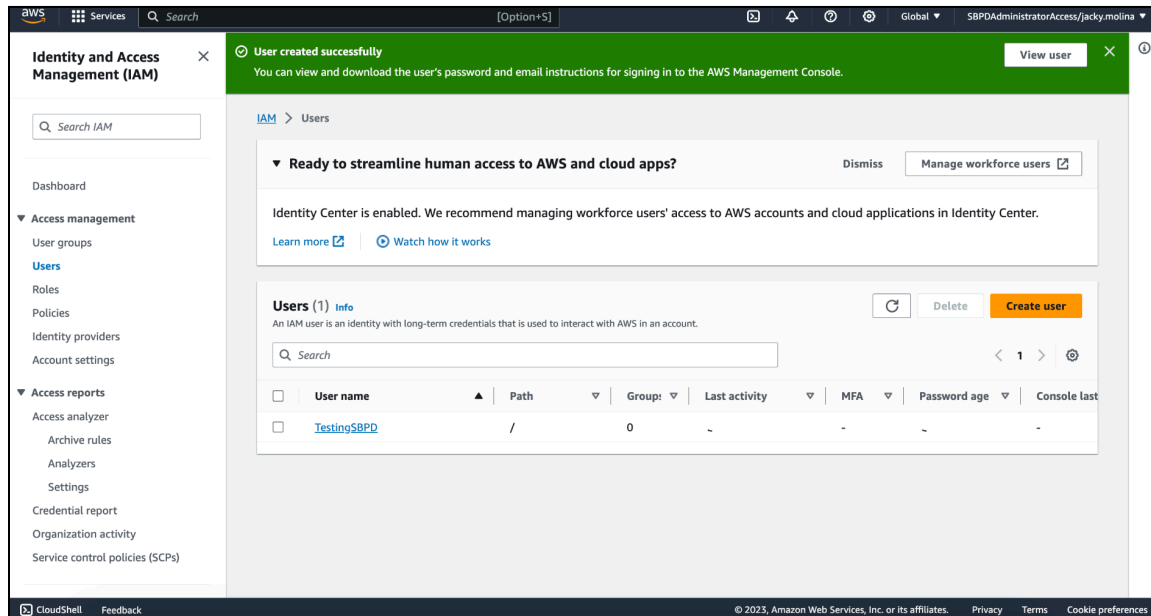
Cancel Download .csv file [Return to users list](#)

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

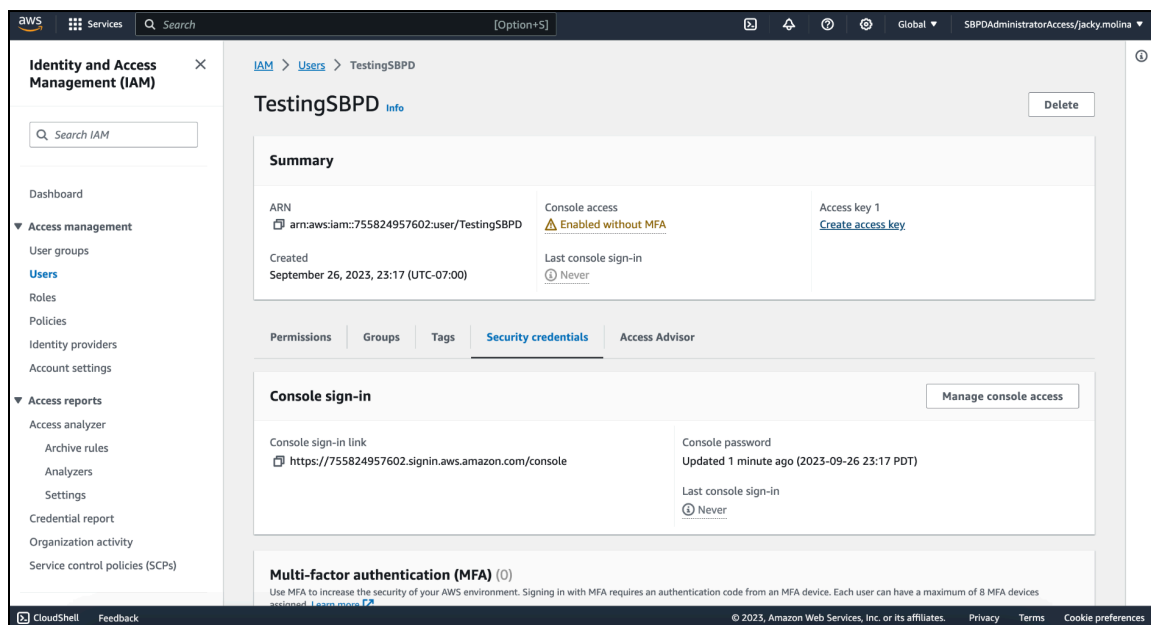
You've created an IAM user!

Creating an Access Key

1. Go to 'Users' in IAM console



2. Select the user and it should take you to a user dashboard, click 'create access key'



3. Select 'Application running on an AWS computer service'

4. Optional: Add a description tag

The screenshot shows the AWS IAM console 'Create access key' page. The left sidebar indicates the current step is 'Set description tag'. The main content area is titled 'Use case' and contains several radio button options: 'Command Line Interface (CLI)', 'Local code', 'Application running on an AWS compute service' (which is selected), 'Third-party service', 'Application running outside AWS', and 'Other'. Below these options is a yellow box with a warning icon and the text: 'Alternative recommended: Assign an IAM role to compute resources like EC2 instances or Lambda functions to automatically supply temporary credentials to enable access. [Learn more](#)'. The bottom of the page shows a 'Confirmation' section.

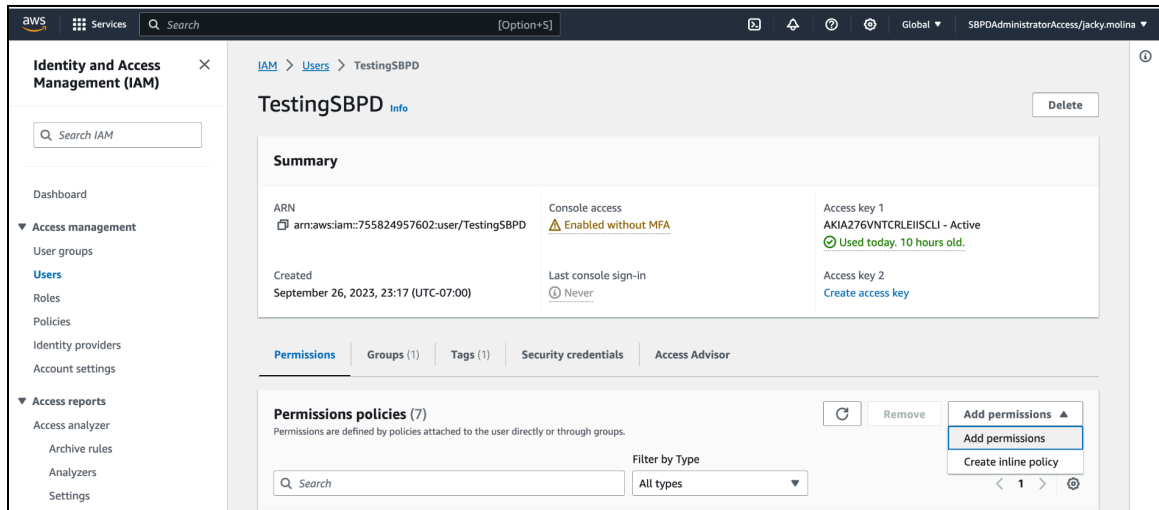
5. You'll get your Access key and Secret Access key

The screenshot shows the AWS IAM console 'Retrieve access keys' page. A green banner at the top states: 'Access key created. This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' The breadcrumb trail is 'IAM > Users > TestingSBPD > Create access key'. The left sidebar shows the current step is 'Retrieve access keys'. The main content area is titled 'Retrieve access keys' and contains a section for the 'Access key' and 'Secret access key'. The Access key is displayed as 'AKIAZ76VNTCRLEIHSCLI'. The Secret access key is masked with asterisks and has a 'Show' button next to it. Below this is a section titled 'Access key best practices' with a list of recommendations: 'Never store your access key in plain text, in a code repository, or in code.', 'Disable or delete access key when no longer needed.', 'Enable least-privilege permissions.', and 'Rotate access keys regularly.' A link at the bottom points to 'best practices for managing AWS access keys'.

Access Key and Secret Access Key created!

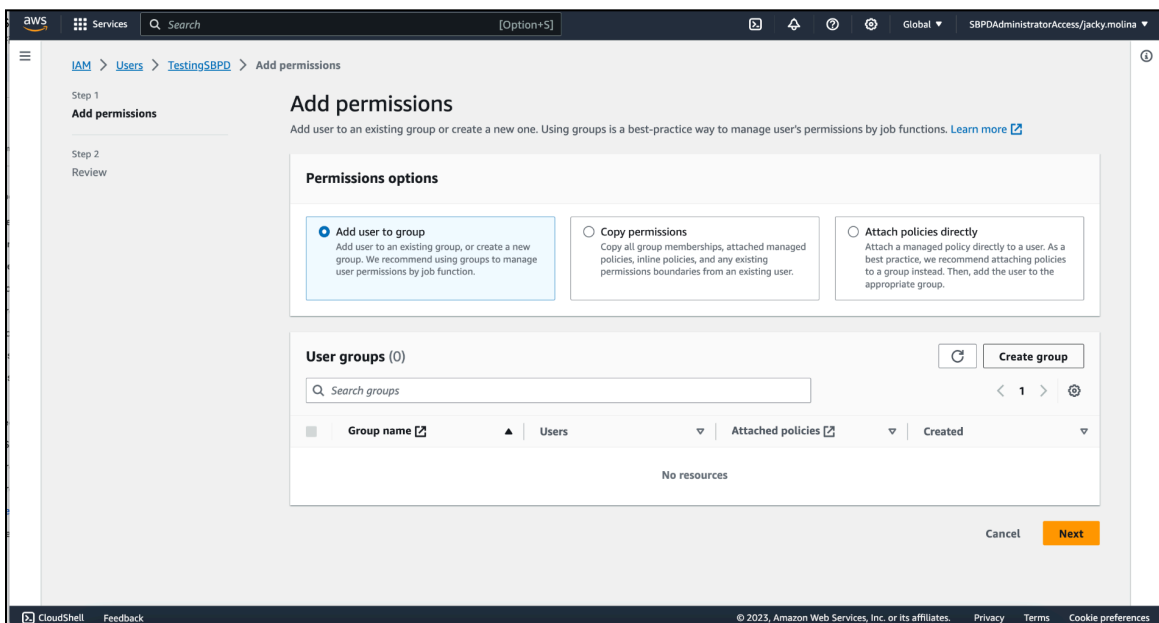
Setting Permissions and Adding User to a Group

1. Go to Users, select your user, and then select 'add permissions'

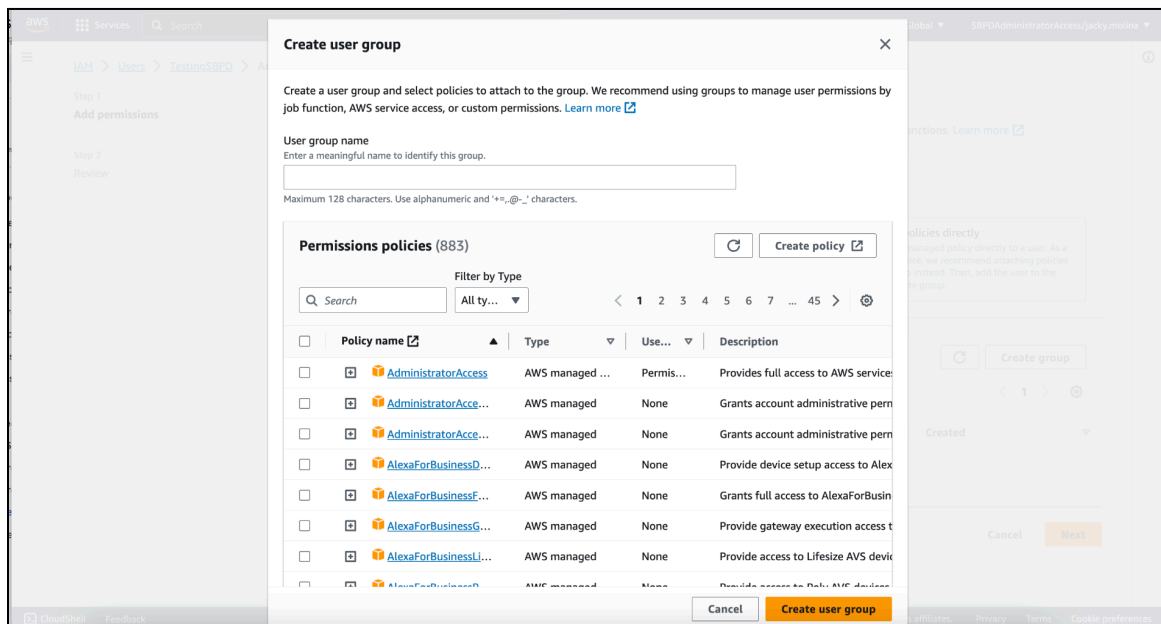


2. Click 'Add user to Group' and then click 'Create Group'

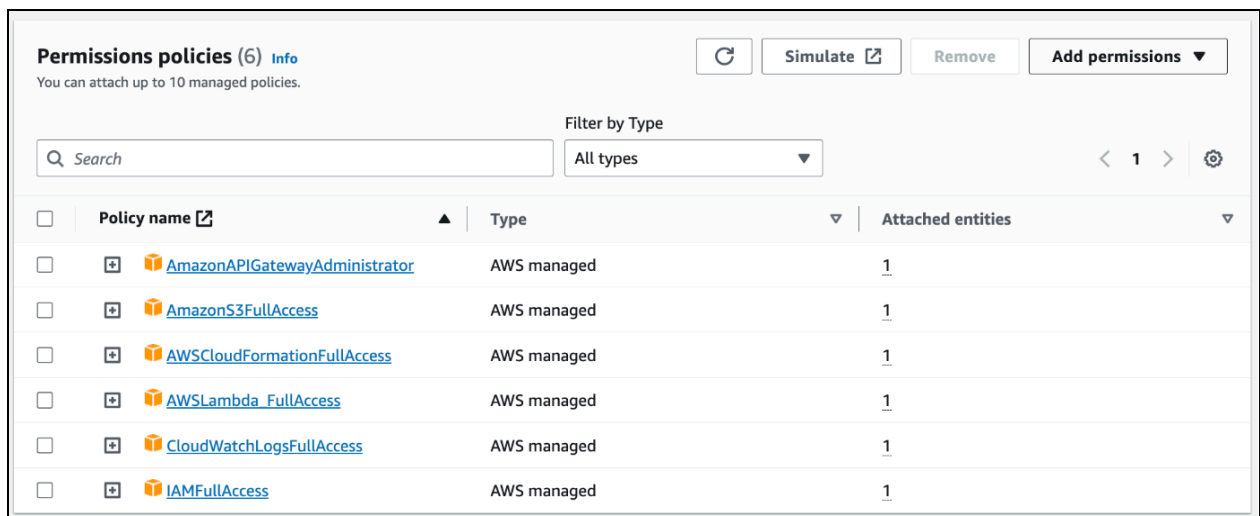
*If someone already completed this process, just add your user to an existing group.



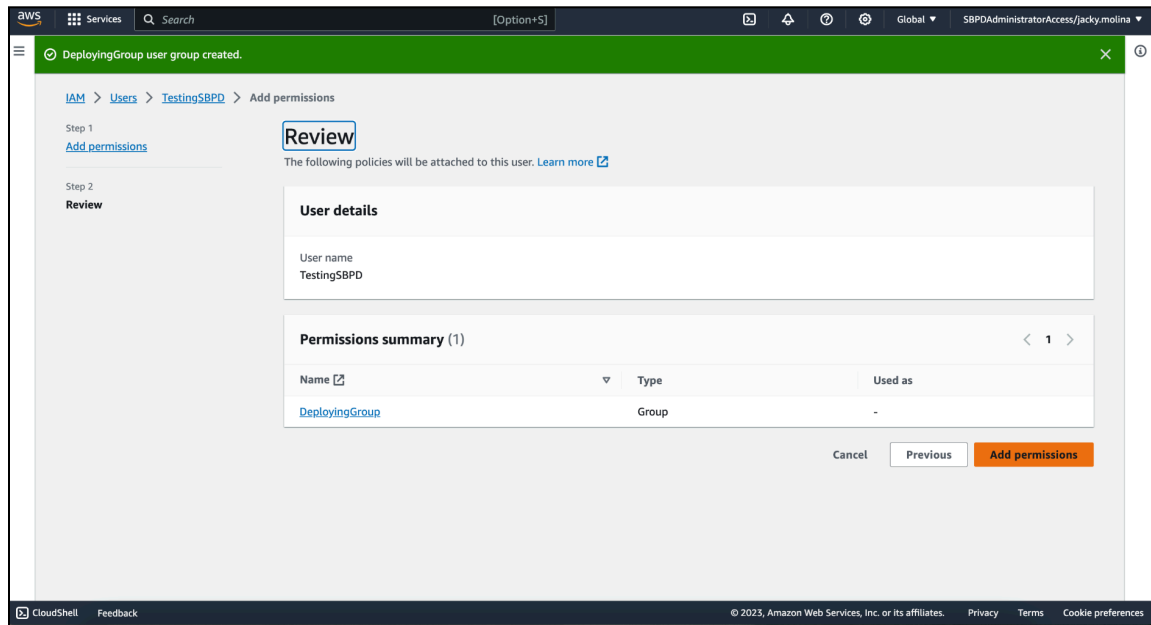
3. Add a name for your group and then select all the permissions you need (see step 4)
After click ‘create user group’



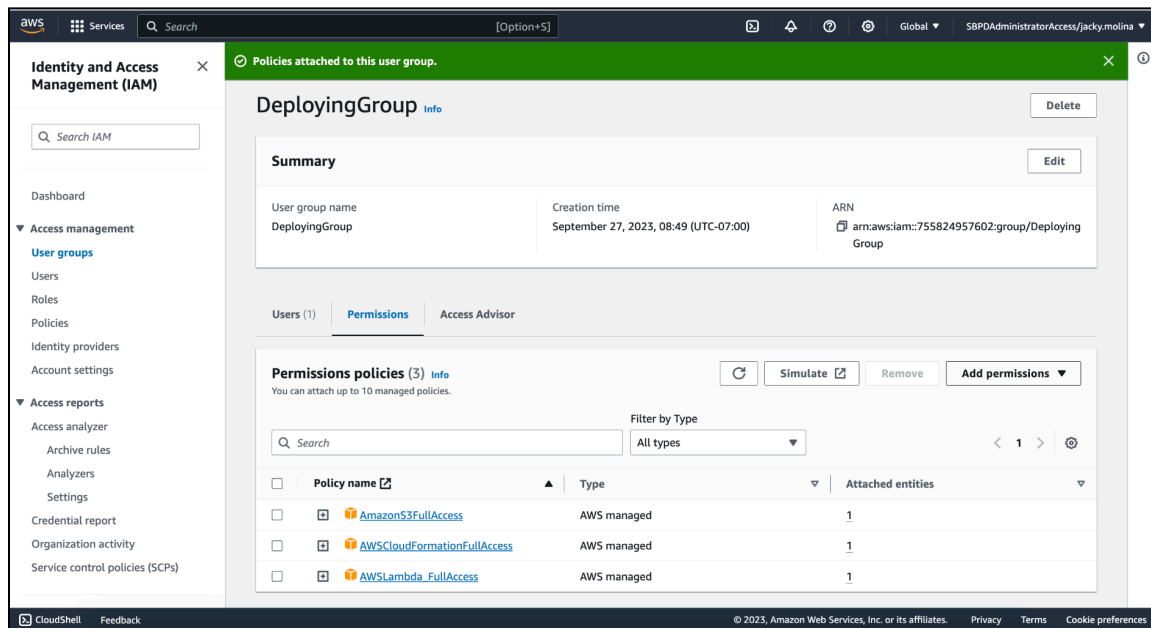
4. Add all of these permissions



5. Click 'Add permissions'



6. You can view all group permissions and add more in the User Groups page



7. You can view all User permissions in the Users page

The screenshot shows the AWS IAM console interface. On the left is a navigation sidebar with sections like 'Identity and Access Management (IAM)', 'Access management', 'Access reports', and 'Credential report'. The main content area is titled 'TestingSBPD' and includes a 'Delete' button. Below the title is a 'Summary' section with three columns: ARN (arn:aws:iam::755824957602:user/TestingSBPD), Console access (Enabled without MFA), and Access key 1 (AKIA276VNTCRLEIISCLI - Active, Used today, 10 hours old). Below the summary are tabs for 'Permissions', 'Groups (1)', 'Tags (1)', 'Security credentials', and 'Access Advisor'. The 'Permissions' tab is selected, showing a table of permissions policies. The table has columns for 'Policy name', 'Type', and 'Attached via'. Two policies are listed: 'AmazonAPIGatewayAdministrator' and 'AmazonS3FullAccess', both of type 'AWS managed' and attached via 'Group DeployingGroup'.

Policy name	Type	Attached via
AmazonAPIGatewayAdministrator	AWS managed	Group DeployingGroup
AmazonS3FullAccess	AWS managed	Group DeployingGroup

You're done setting permissions!

Deploying to AWS with Serverless (for the first time)

1. Open a terminal in your VSCode - make sure you are in the correct folder (cd if you have to)

2. (this is what you're going to be typing)

```
serverless config credentials -o --provider aws --key 'YOUR_AWS_KEY' --secret 'YOUR_AWS_SECRET'
```

Or try: (if top does not work)

```
serverless config credentials -o --provider aws -k 'YOUR_AWS_KEY' --secret 'YOUR_AWS_SECRET'
```

```
kiki@moony box.com_eDefender_integration-main % serverless config credentials -o --provider aws --key 'YOUR_AWS_KEY' --secret 'YOUR_AWS_SECRET'
```

Example: this was mine

```
kiki@moony box.com_eDefender_integration-main % serverless config credentials -o --provider aws --key AKIA276VNTCRLEIISCLI --secret Fb6Y8fyKhW+uRMQFK0h+AeZ5Idqp4crxD4bmRLpq
```

3. You should see this run and a message that says 'Profile "default" has been configured'

```
kiki@moony box.com_eDefender_integration-main % serverless config credentials -o --provider aws --key AKIA276VNTCRLEIISCLI --secret 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

Running "serverless" from node_modules
✓ Profile "default" has been configured
kiki@moony box.com_eDefender_integration-main %
```

4. Go to the .yaml file and change the 'name' to 'box-video-skills-[your-name]'
(This will help identifying which function and bucket are yours on aws)

```
! serverless.yml x {} package.json {} config.json JS index.js JS video-indexer.js

! serverless.yml
1 service: box-video-skill-jackie
2 provider:
3   name: aws
4   runtime: nodejs14.x
```


5. Run 'sls deploy' or 'serverless deploy'

(This is what you should see happening)

```
o kiki@moony box.com_eDefender_integration-main % sls deploy
Running "serverless" from node_modules

Deploying box-video-skill-promise to stage dev (us-east-1)

:: Packaging (6s)
```

(SideNote1: if at any point you get a permissions error, just add that permission to your user/group and then run 'sls deploy' again)

(SideNote2: if at any point you get an error that says '...ROLLBACK_COMPLETE state and cannot be updated', run 'sls remove' and then 'sls deploy')

Finding APIGateway

5.This should be the final result you see (SAVE THE 'endpoint' LINK SOMEWHERE BECAUSE YOU'LL NEED IT AND YOU ONLY GET IT **ONCE PER DEPLOYMENT**)

```
Deploying box-video-skill-promise to stage dev (us-east-1)
✓ Service deployed to stack box-video-skill-promise-dev (110s)

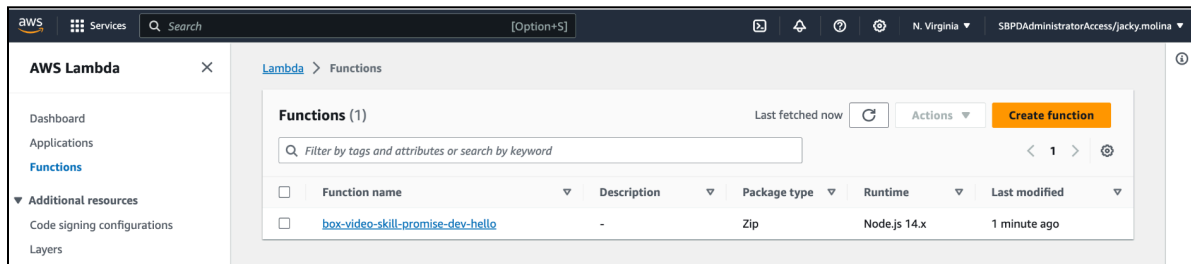
endpoint: ANY - https://tg1pak7d6c.execute-api.us-east-1.amazonaws.com/dev/
functions:
  hello: box-video-skill-promise-dev-hello (13 MB)

1 deprecation found: run 'serverless doctor' for more details

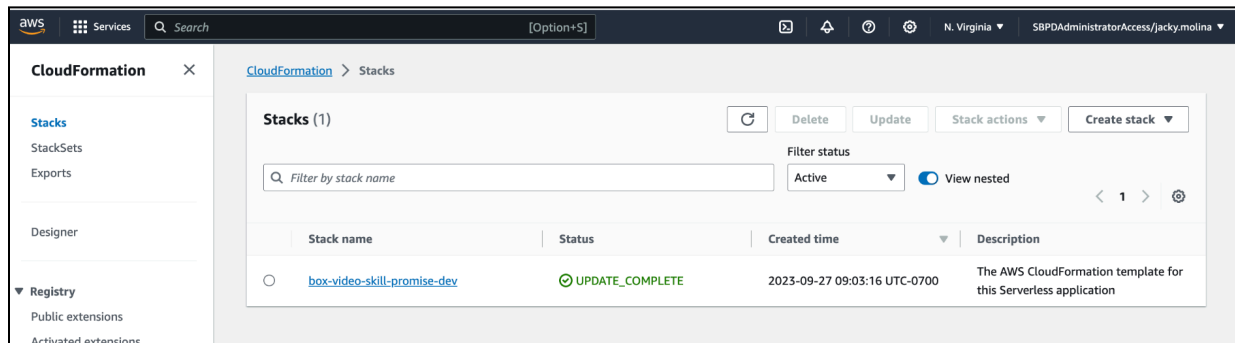
Monitor Express APIs by route with the Serverless Console: run "serverless --console"
o kiki@moony box.com_eDefender_integration-main %
```

This Link is your APIGateway needed as an Environment Variable!

CHECK: If you go to the AWS Lambda service, you should see this:



CHECK: If you go to CloudFormation, you should see this



You've deployed your code to AWS using serverless!

Environment Variables Key

(A good idea would be to create a .json file with all your non-changing variables so you don't have to look for them every single time you deploy)

APIGATEWAY	'sls deploy' link (changes with every deployment)
S3_BUCKET	YOUR S3 bucket name (<u>changes with every deployment</u>)
VI_ACCOUNT_ID	Video Indexer Account ID
VI_AUTH_KEY_1	Video Indexer Developer Primary Key
VI_AUTH_KEY_2	Video Indexer Developer Secondary Key
BOX_PRIMARY_KEY	Box Primary Key
BOX_SECONDARY_KEY	Box Secondary Key

* S3_BUCKET is the **NAME** of your S3 bucket that was generated with your APIGateway link (Found in AWS S3 service)

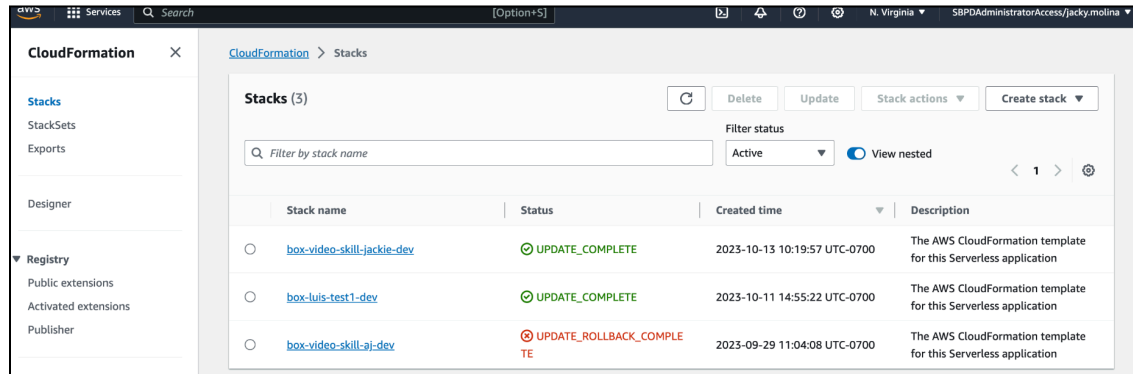
** To find the other environment variables, look under the '**Finding**' sections in this manual

Configuring BOX, S3, and LAMBDA

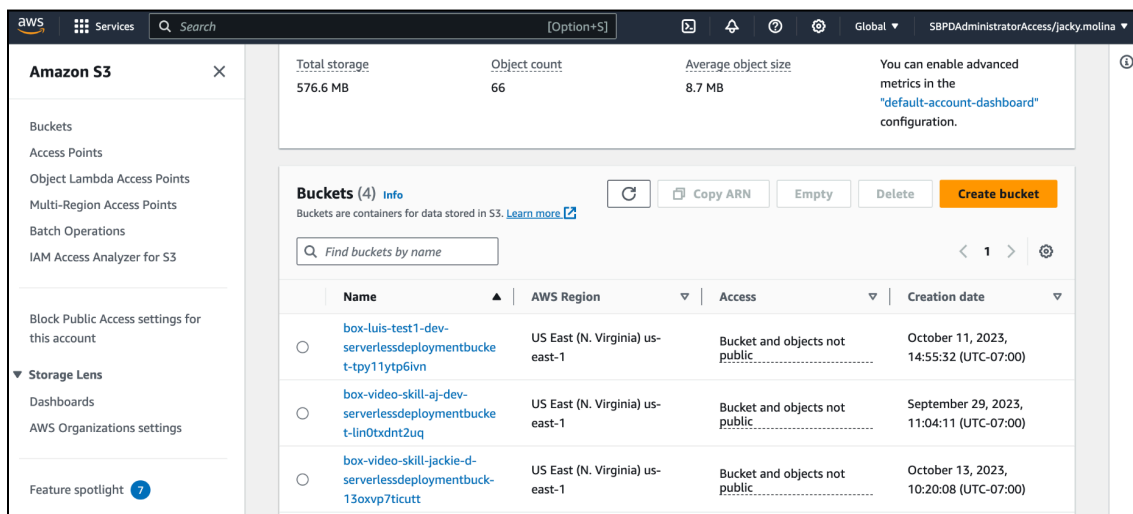
Making sure your lambda function and S3 bucket was created

1. Go to 'CloudFormation' and click on 'Stacks'

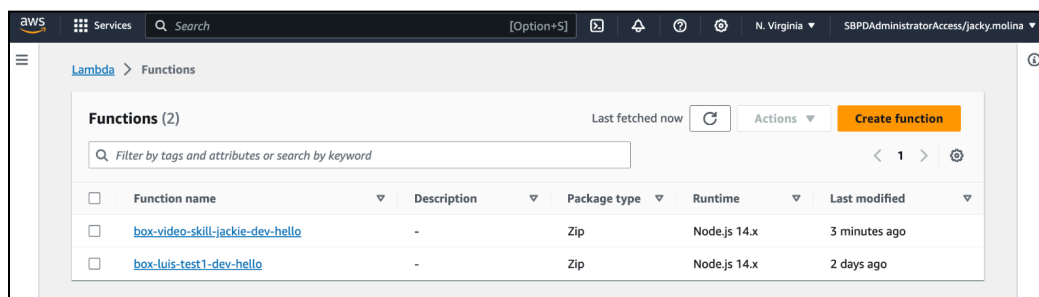
Check to see if your deployment name is there



2. Go to S3 dashboard and check to see if your deployment name is there



3. Go to the Lambda and click Functions and check for your deployment name

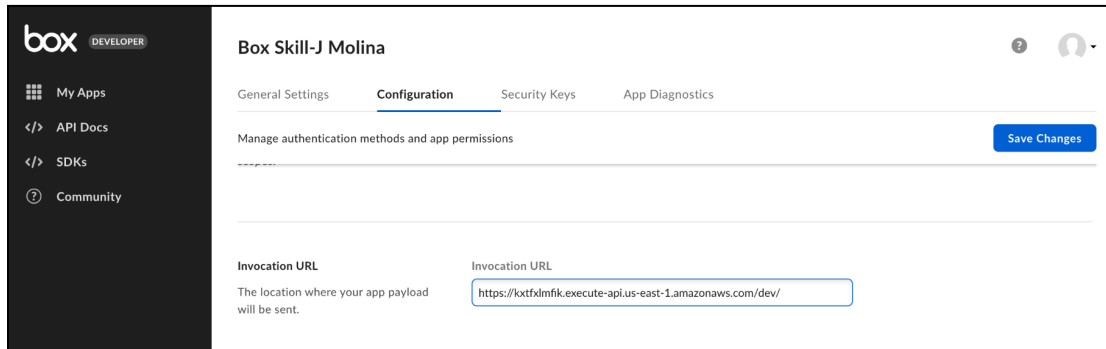


You deployed your function correctly!

Configuring Box

Will need to be done with every deployment because the AWS link changes

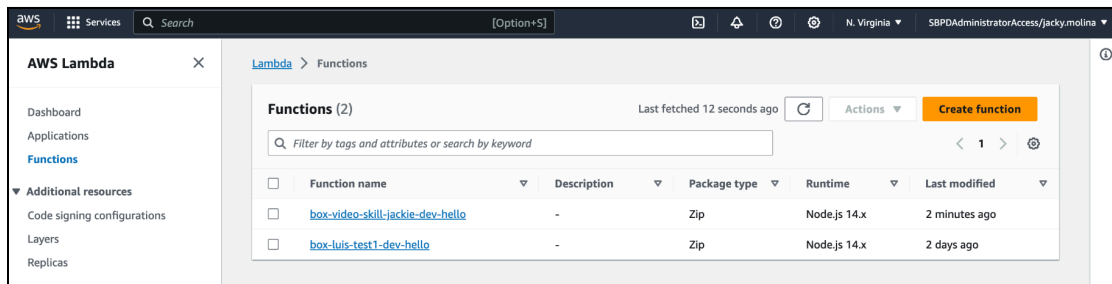
1. Go to your developer Box account, click on your Skill and add the AWS link into the 'Invocation URL'



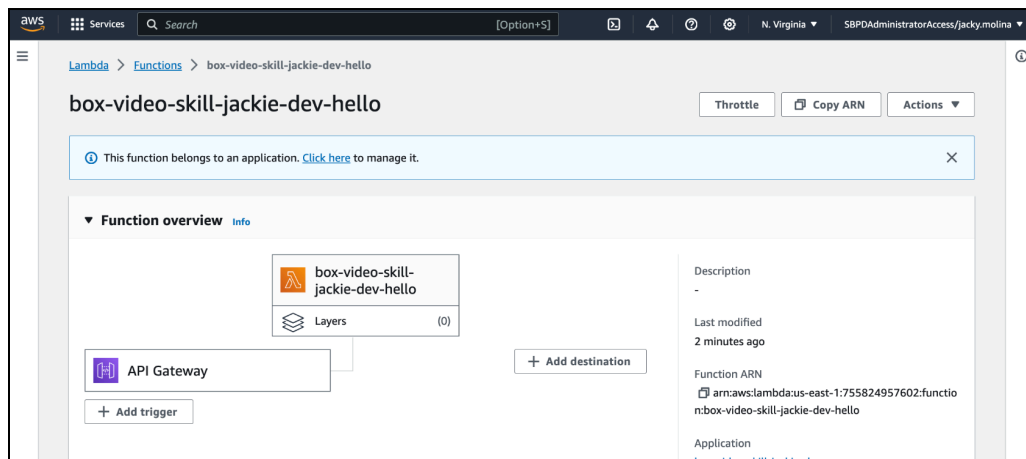
You're done with this step!

Configuring Lambda

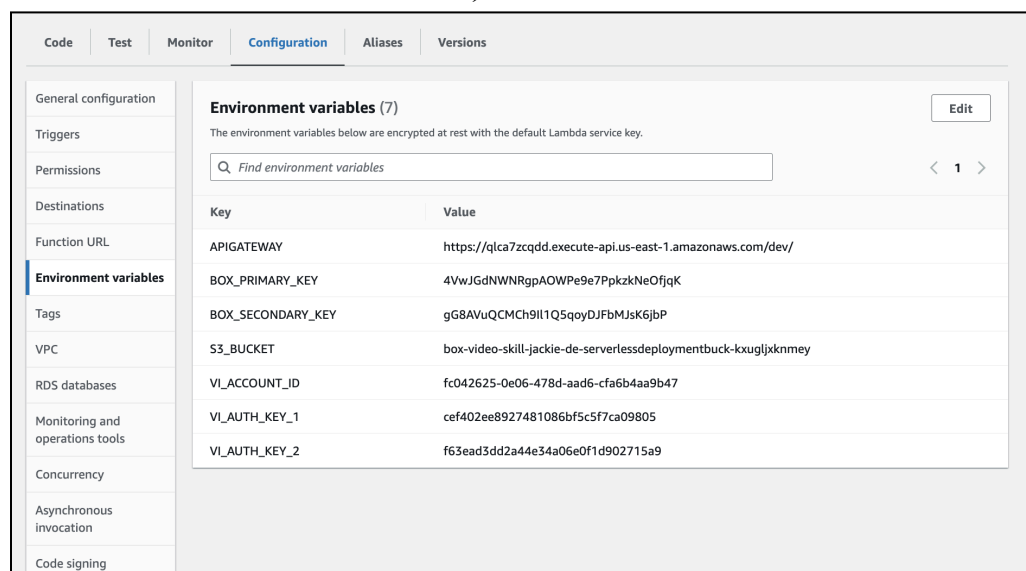
1. Go to your AWS Lambda Functions and click on your function



2. This is what your dashboard should look like, scroll down



3. Click on the Configuration tab, click edit, and add ALL environment variables (see **ENVIRONMENT VARIABLES KEY**)



Post-Deployment Information

How to test your AWS Lambda Function?

1. Go to your Box folder and upload a file (mp4 would be the easiest)

Try to do a short video to cut down on processing time

2. Go to AWS Lambda and click the Monitoring tab

Find the Timestamp that is closest to the time you uploaded the video (it might take a couple minutes for it to appear) and click on the LogStream link.

The screenshot shows the AWS Lambda console interface. The 'Monitor' tab is selected, displaying 'CloudWatch Logs'. Below the header, there are buttons for 'Metrics', 'Logs', and 'Traces'. The 'Logs' button is active. A table titled 'Recent invocations' lists the most recent function executions. Each row includes a number, a timestamp, a request ID, a log stream link, and the duration in milliseconds.

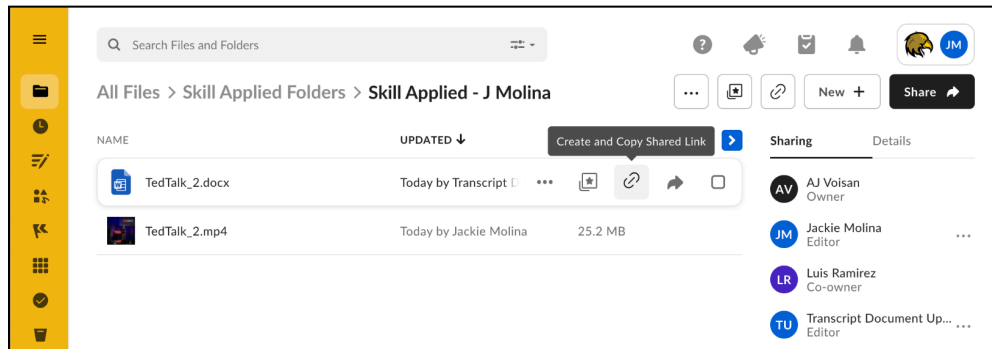
#	Timestamp	RequestID	LogStream	DurationInMS
1	2023-10-14T02:32:06.075Z	ab50c206-4c30-4006-a34b-d9c8f429fc69	2023/10/14/[LATEST]3a32f7084f34448ebe8d8c8d5f6e951e	364.76
2	2023-10-14T02:25:32.367Z	15d0b265-2191-4352-9a55-3d483ad2d01d	2023/10/14/[LATEST]2f12e2e7d8254872b3e474229be79080	340.98
3	2023-10-14T02:25:15.744Z	06eaf995-bb1d-4269-8d80-545067f5271b	2023/10/14/[LATEST]2f12e2e7d8254872b3e474229be79080	4583.08
4	2023-10-14T02:21:48.392Z	64939b5d-4bdc-43fe-b78a-d9ccc1d92dbc	2023/10/14/[LATEST]2f12e2e7d8254872b3e474229be79080	413.79
5	2023-10-14T02:21:41.342Z	74a699f1-93b8-4195-8179-1647a28e8bfc	2023/10/14/[LATEST]2f12e2e7d8254872b3e474229be79080	6287.98
6	2023-10-14T02:18:30.420Z	5e6156e2-e5a9-4b6d-b6bc-666745f5952b	2023/10/14/[LATEST]76da306421e041e091fb2630f61662e1	4273.37
7	2023-10-14T02:15:39.575Z	39c445ee-c919-4ff7-bc86-81066bccc205	2023/10/14/[LATEST]76da306421e041e091fb2630f61662e1	4685.98
8	2023-10-14T02:15:22.284Z	5d34fa0b-2550-408f-86ed-848e6dbdf35	2023/10/14/[LATEST]76da306421e041e091fb2630f61662e1	5539.08
9	2023-10-14T02:12:17.119Z	0934fde4-1b9a-4309-a5f2-6717dc158061	2023/10/14/[LATEST]76da306421e041e091fb2630f61662e1	28034.4

Here you can see if any errors occur.

The screenshot shows the AWS CloudWatch console interface. The 'Log groups' section is selected, displaying a list of log groups. The log group '2023/10/14/[LATEST]3a32f7084f34448ebe8d8c8d5f6e951e' is selected. The 'Log events' section is visible, showing a list of log events. The events are filtered by the selected log group and the time range is set to '3h'. The events are displayed in a table with columns for 'Timestamp' and 'Message'.

Timestamp	Message
2023-10-13T19:32:04.342-07:00	INIT_START Runtime Version: nodejs:14.v37 Runtime Version ARN: arn:aws:lambda:us...
2023-10-13T19:32:05.710-07:00	START RequestId: ab50c206-4c30-4006-a34b-d9c8f429fc69 Version: \$LATEST
2023-10-13T19:32:05.729-07:00	2023-10-14T02:32:05.729Z ab50c206-4c30-4006-a34b-d9c8f429fc69 INFO { type: 'ski...
2023-10-13T19:32:05.729-07:00	2023-10-14T02:32:05.729Z ab50c206-4c30-4006-a34b-d9c8f429fc69 DEBUG Box event re...
2023-10-13T19:32:05.805-07:00	2023-10-14T02:32:05.805Z ab50c206-4c30-4006-a34b-d9c8f429fc69 INFO <ref *2> Inco...
2023-10-13T19:32:05.805-07:00	2023-10-14T02:32:05.805Z ab50c206-4c30-4006-a34b-d9c8f429fc69 INFO statusCode: 2...
2023-10-13T19:32:05.805-07:00	2023-10-14T02:32:05.805Z ab50c206-4c30-4006-a34b-d9c8f429fc69 INFO headers: { 'c...
2023-10-13T19:32:05.806-07:00	2023-10-14T02:32:05.806Z ab50c206-4c30-4006-a34b-d9c8f429fc69 INFO eyJhbGciOiJIU...
2023-10-13T19:32:05.808-07:00	2023-10-14T02:32:05.808Z ab50c206-4c30-4006-a34b-d9c8f429fc69 INFO sourceFolderI...
2023-10-13T19:32:05.808-07:00	2023-10-14T02:32:05.808Z ab50c206-4c30-4006-a34b-d9c8f429fc69 INFO Request ID: 7...
2023-10-13T19:32:05.888-07:00	2023-10-14T02:32:05.888Z ab50c206-4c30-4006-a34b-d9c8f429fc69 INFO { type: 'ski...

3. After about 5-10 minutes, go back to Box and refresh. You should see a Docx file and your video with new insights.



What do I do if I change the code?

1. Delete S3 Bucket (empty it if you have to)
2. Delete CloudFormation stack
3. Redeploy through Serverless
4. Change the Box Dev Invocation link to the new AWS Link
5. On AWS Lambda re-enter all your environment variables

You should be ready to test the environment again!