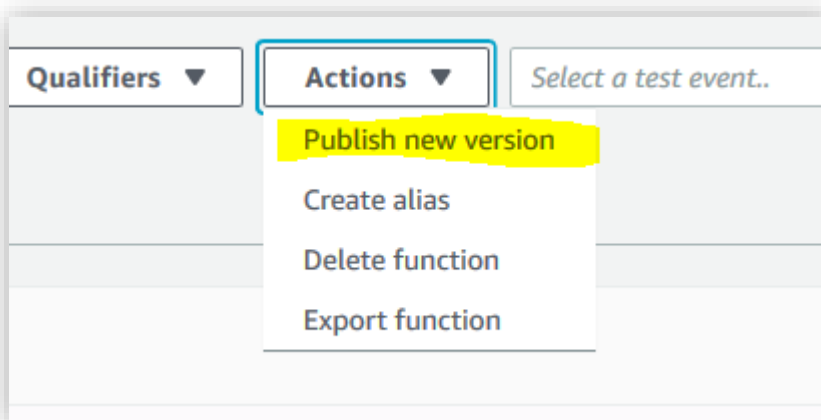


Lab 4: Create a Canary deployment

Using our existing Lambda function, change the code to show a version number:

```
exports.handler = (event, context, callback) => {  
  const response = {  
    statusCode: 200,  
    body: JSON.stringify({ message: 'Version 1' }),  
    headers: {  
      'Access-Control-Allow-Origin': '*',  
      'Content-Type': 'application/json'  
    }  
  };  
  console.log(response);  
  callback(null, response);  
};
```

Save your function, and publish a new version by clicking on “Actions” and selecting Publish new version. Be sure you save your function before publishing a new version:



Enter any description you would like for the version description. Commonly this will be a date time stamp or a git commit hash and then click on publish:

Publish new version from \$LATEST ×

Publishing a new version will save a "snapshot" of the code and configuration of the \$LATEST version. You will be unable to edit the new version's code. Please click to confirm.

Version description

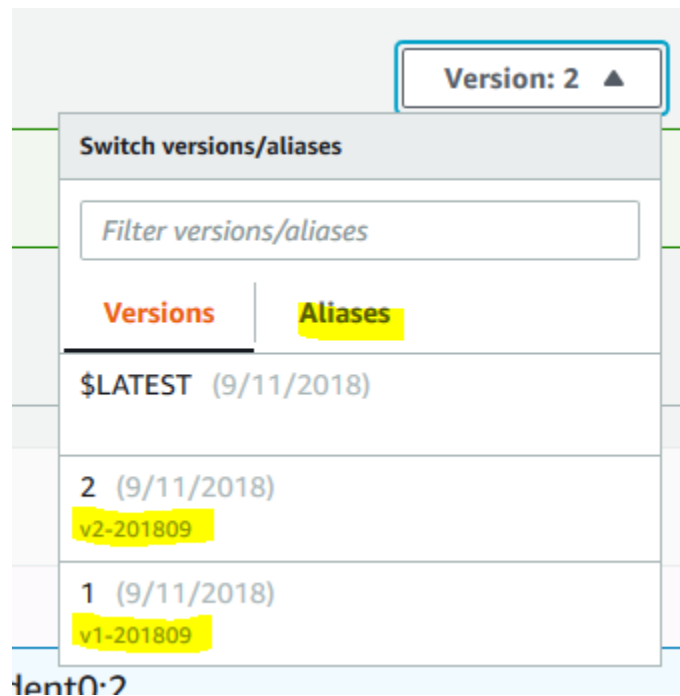
Cancel Publish

After you publish a version, you will no longer be able to edit that version. You will see a warning letting you know it can't be edited with a link to work on the "Latest" branch of the function if you want to make edits.

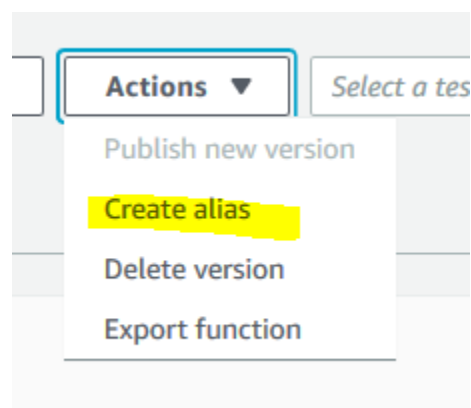
Function code (Preview) [Info](#)

ⓘ Code and handler editing is only available for the \$LATEST version. [Click here to go to \\$LATEST.](#)

Change your function again to show another version or a different message of your choosing, and save another version. So you have 2 versions in the version drop down.



Under actions select “Create Alias”:



Create a new alias, the name is important this will be used to connect it to the API Gateway so pick something simple. Select your versions from the drop downs and assign a weight. For this lab use 50% so each version has an equal chance of being run. In a real life scenario you would start much lower with possibly 5% or less traffic, ramping up over time:

Create a new alias

An alias is a pointer to one or two versions. Select the version(s) you would like the alias to point to.

Name*

CanaryFunction

Description

Canary function 50/50 split

Version*

1

Weight: 50%

You can shift traffic between two versions, based on weights (%) that you assign. Click [here](#) to learn more.

Additional Version

2

Weight

50

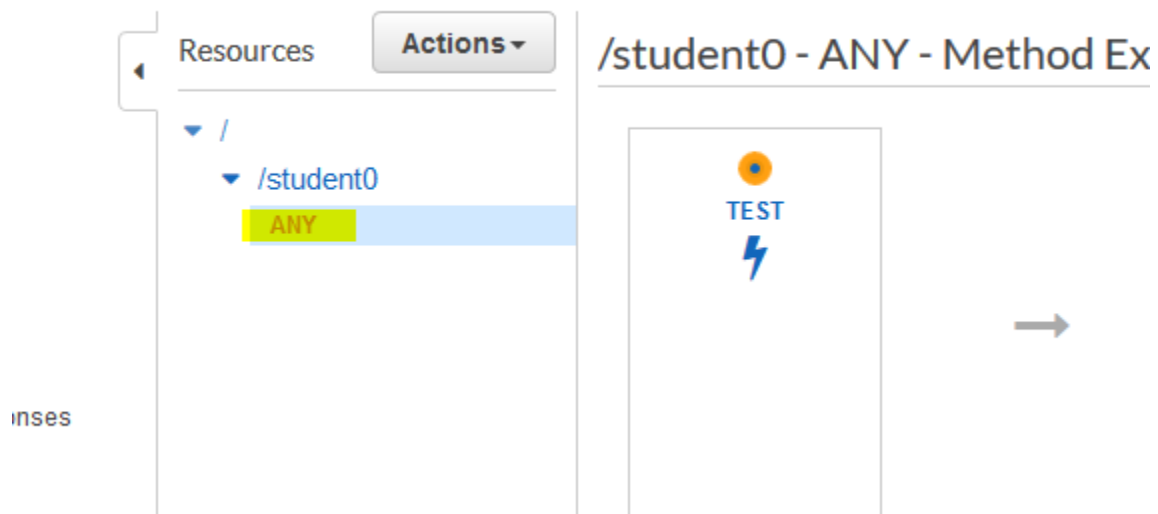
%

Cancel

Create

Now that we have our canary enabled function, go back to the API gateway, select your API endpoint and click on “ANY” to modify the Lambda wiring for all of the methods for that API endpoint:

ateway > APIs > student0-API (w018end1q8) > Resources > /student0 (73TK4x)



On the right hand side you will see a box with the title “Integration Request” click on the title to access the low level function configuration:

Integration Request

Type: LAMBDA_PROXY

Click on the edit icon next to the Lambda Function:

[← Method Execution](#) /student0 - ANY - Integration Request

Provide information about the target backend that this method will call and whether the incoming request data should be modified.

Integration type ☒ Lambda Function ⓘ

☐ HTTP ⓘ

☐ Mock ⓘ

☐ AWS Service ⓘ

☐ VPC Link ⓘ

Use Lambda Proxy integration ☒ ⓘ

Lambda Region us-east-2 ✎

Lambda Function student0 ✎

Invoke with caller credentials ☐ ⓘ


Credentials cache Do not add caller credentials to cache key ✎

Use Default Timeout ☒ ⓘ

Add the name of your canary function to the end of the function name. The “:” is how you point the gateway towards a specific alias or version number. If you enter the incorrect data you will be shown an error message that just says “unknown error occurred”:

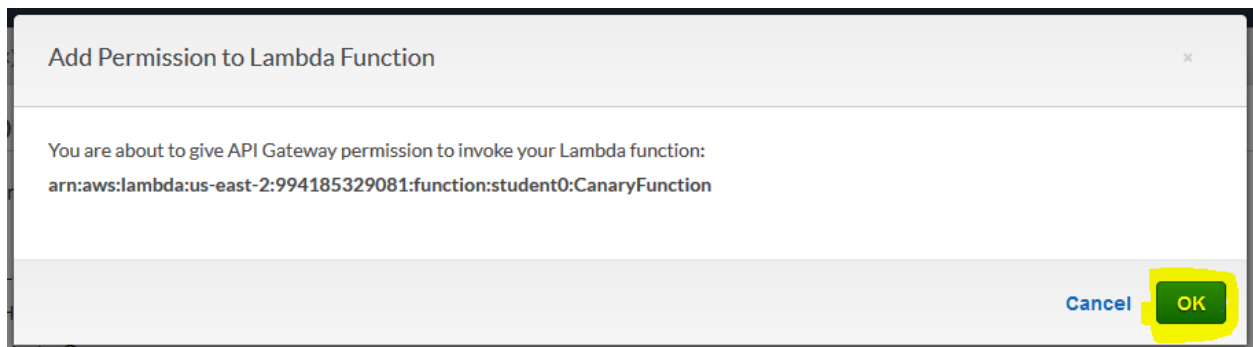
Use Lambda Proxy integration ☒ ⓘ

Lambda Region us-east-2 

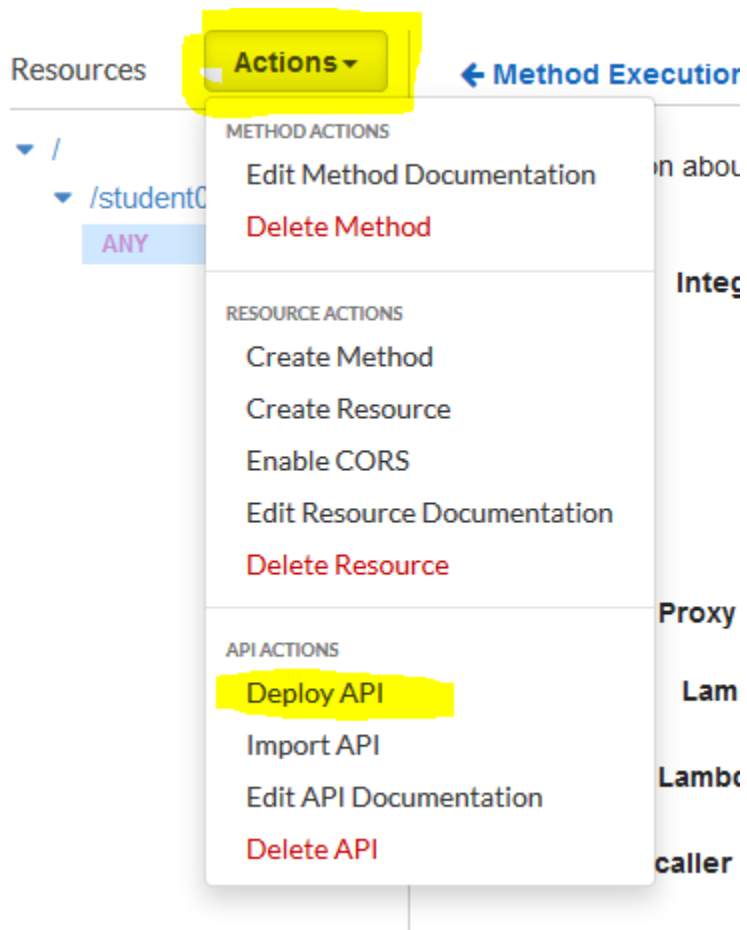
Lambda Function  

Invoke with caller credentials ☐ ⓘ

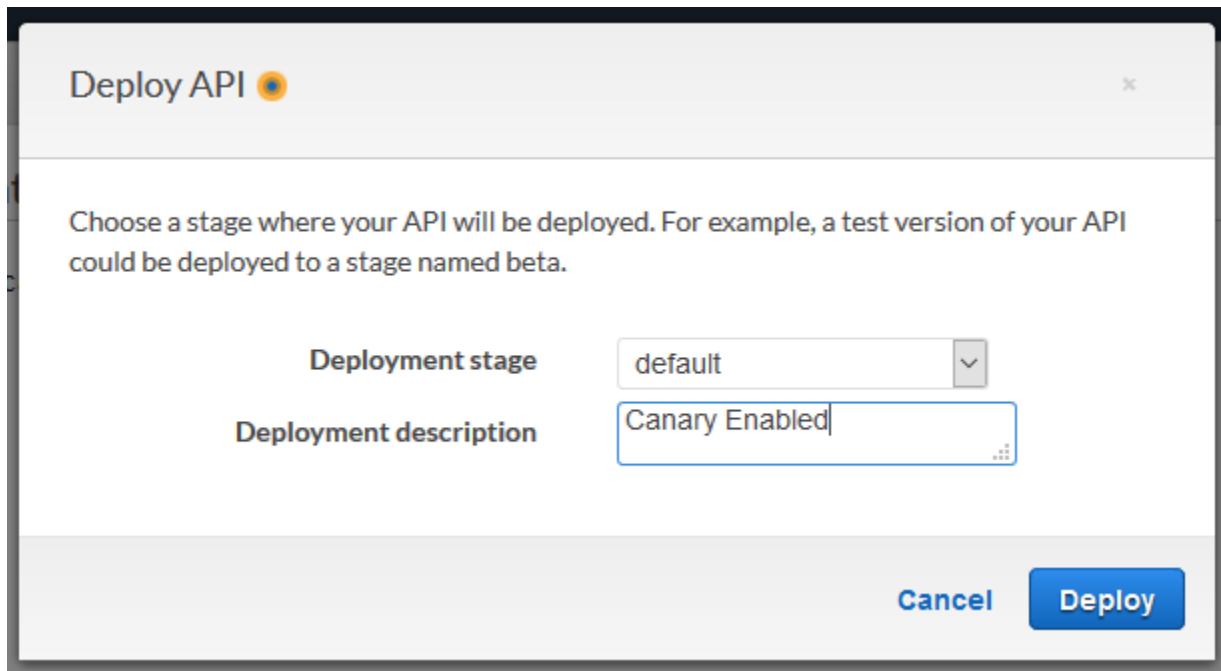
You will be prompted with a warning to add permissions, this allowed the function the be called from the API Gateway. Click “OK”:



We need to deploy the API for the changes to the Lambda pointer to take effect. Click on “Actions” and select “Deploy API”:



Add a meaningful description and click on deploy:



The image shows a 'Deploy API' dialog box with a title bar containing the text 'Deploy API' and a close button. Below the title bar, there is a paragraph of text: 'Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.' Below this text, there are two labels: 'Deployment stage' and 'Deployment description'. The 'Deployment stage' label is positioned to the left of a dropdown menu that currently shows 'default'. The 'Deployment description' label is positioned to the left of a text input field that contains the text 'Canary Enabled'. At the bottom right of the dialog box, there are two buttons: 'Cancel' and 'Deploy'.

Deploy API

Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.

Deployment stage default

Deployment description Canary Enabled

Cancel Deploy

Now try calling your API endpoint multiple times in a row the same way we tested before. You should on average see version 1 and version 2 about half the time. It is random not round robin so you will see the same message multiple times in a row.