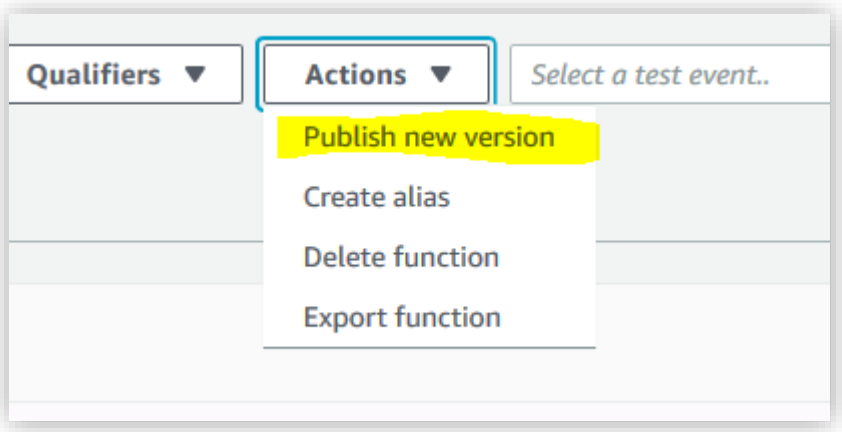
Lab 4: Create a Canary deployment

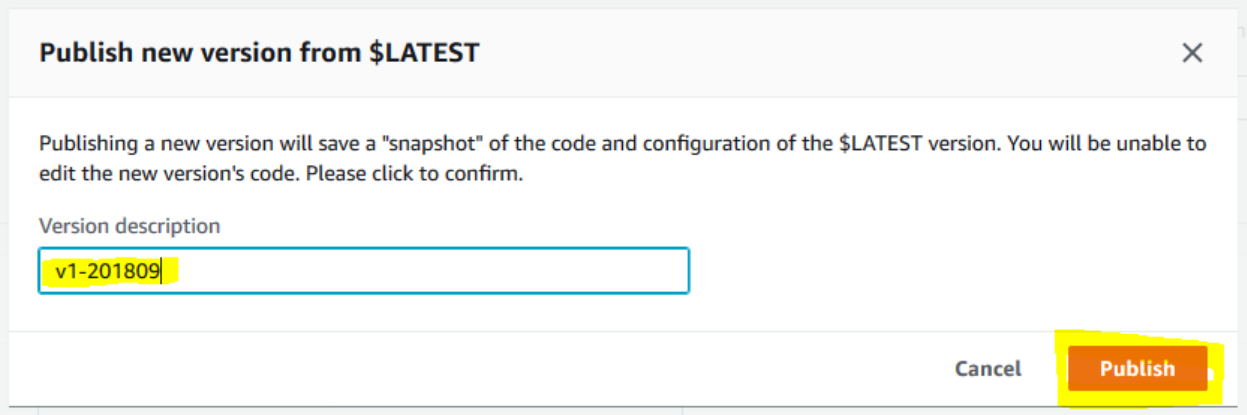
Using our existing Lambda function, change the code to show a version number:

```
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);
};
```

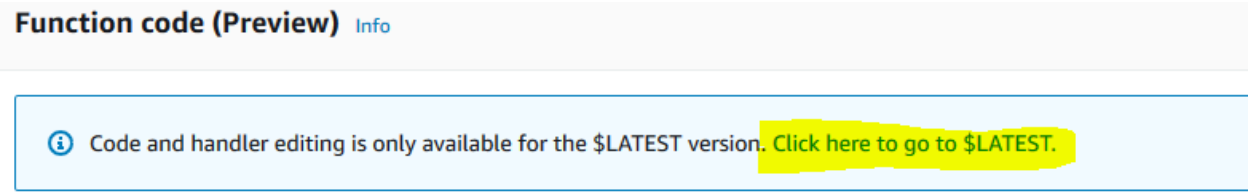
Save your function, and publish a new version by clicking on “Actions” and selecting Publish new version. Be sure you save your function before publishing a new version:



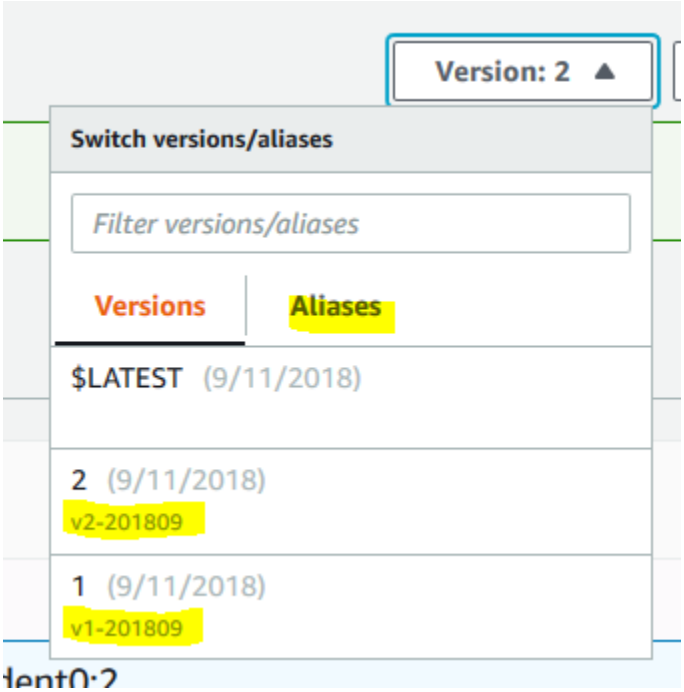
Enter any description you would like for the version description. Commonly this will be a date time stamp or a git commit hash and then click on publish:



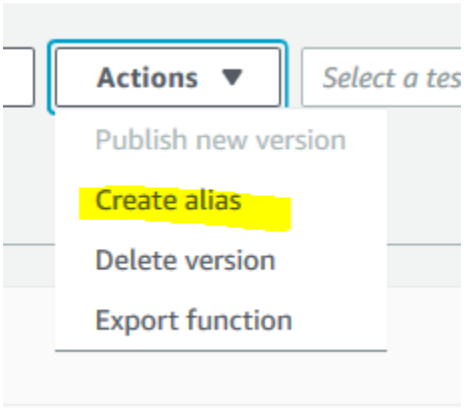
After you publish a version, you will no longer be able to edit that version. You will see a warning letting you know it can't be edited with a link to work on the "Latest" branch of the function if you want to make edits.



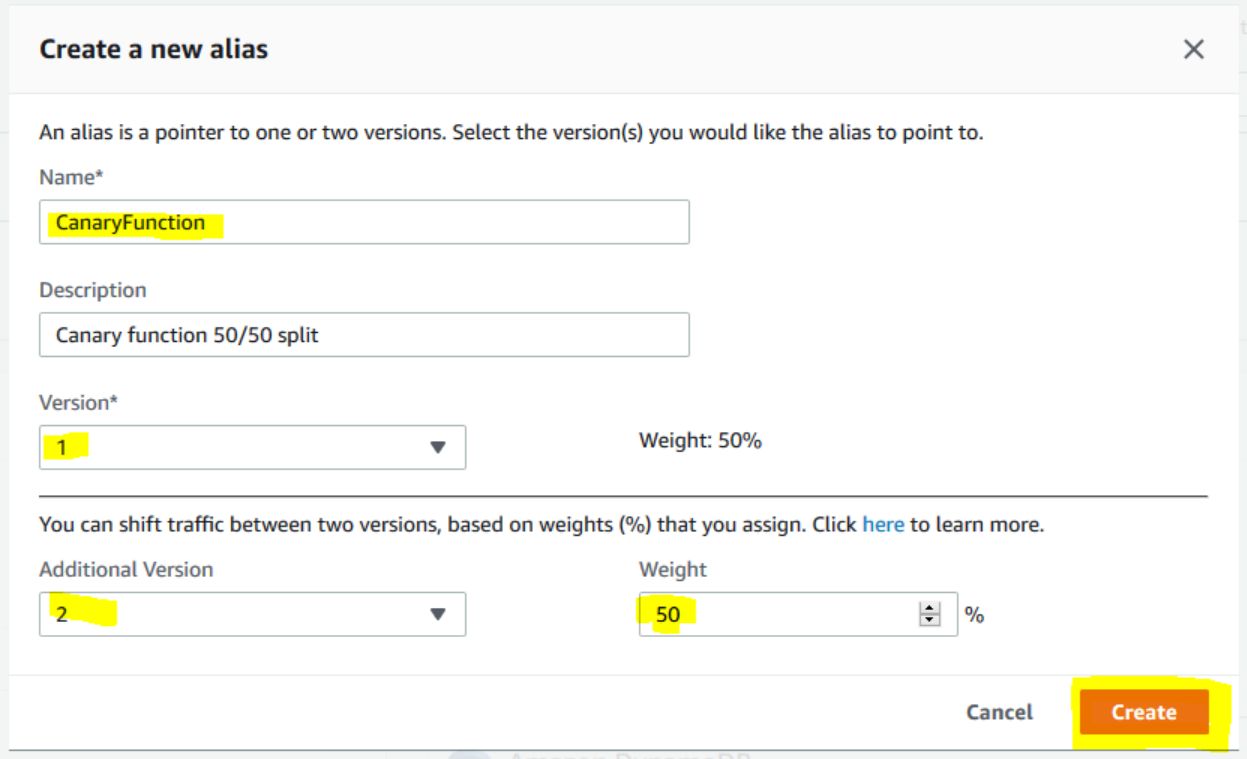
Change your function again to show another version or a different message of your choosing, and save another version. So you have 2 versions in the version drop down.



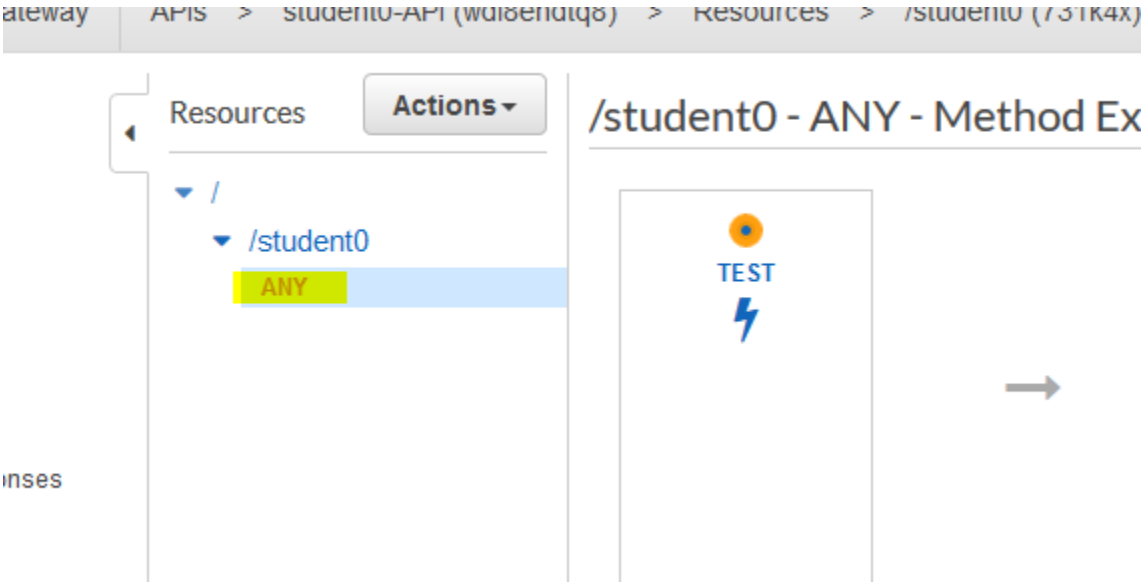
Under actions select “Create Alias”:



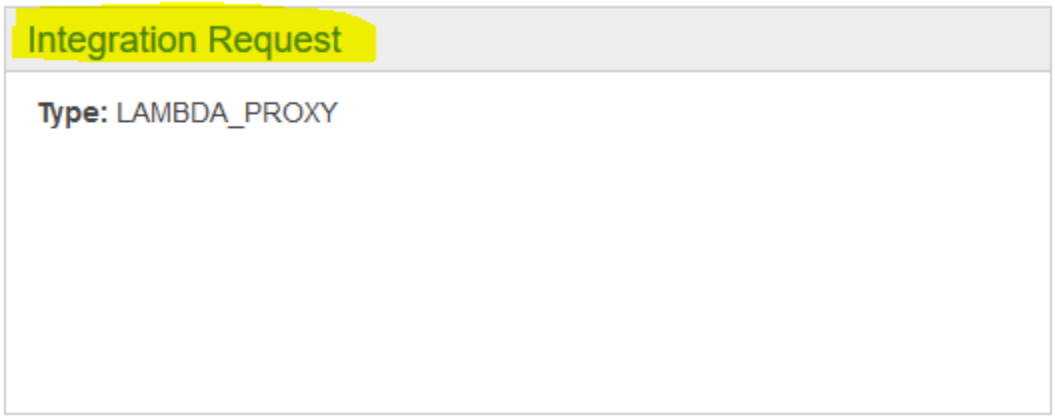
Create a new alias, the name is important this will be used to connect it to the API Gateway so pick something simple. Select your versions from the drop downs and assign a weight. For this lab use 50% so each version has an equal chance of being run. In a real life scenario you would start much lower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



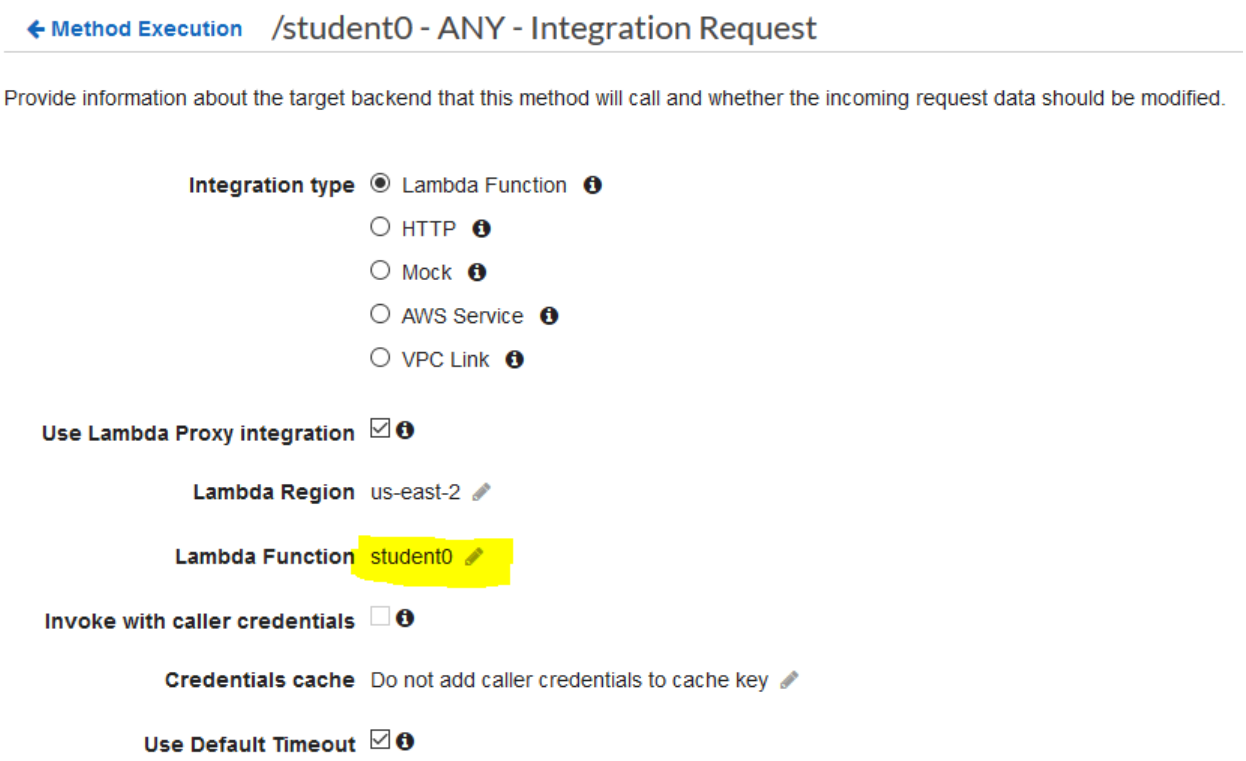
Now that we have our canary enabled function, go back to the API gateway, select your API endpoint and click on “ANY” to modify the Lambda wiring for all of the methods for that API endpoint:



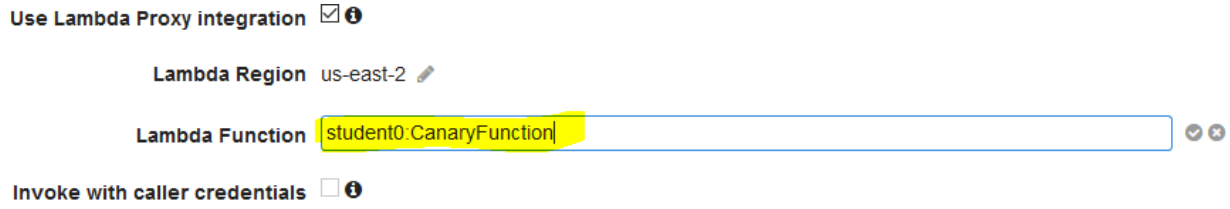
On the right hand side you will see a box with the title “Integration Request” click on the title to access the low level function configuration:



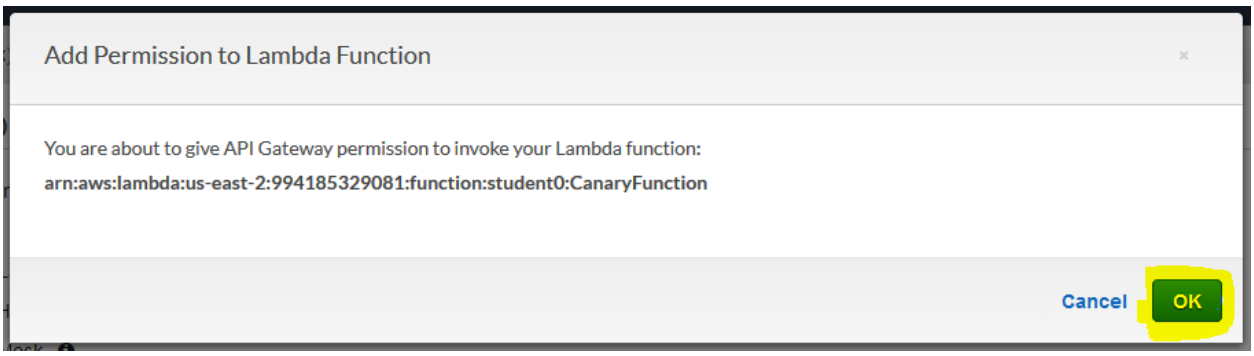
Click on the edit icon next to the Lambda Function:



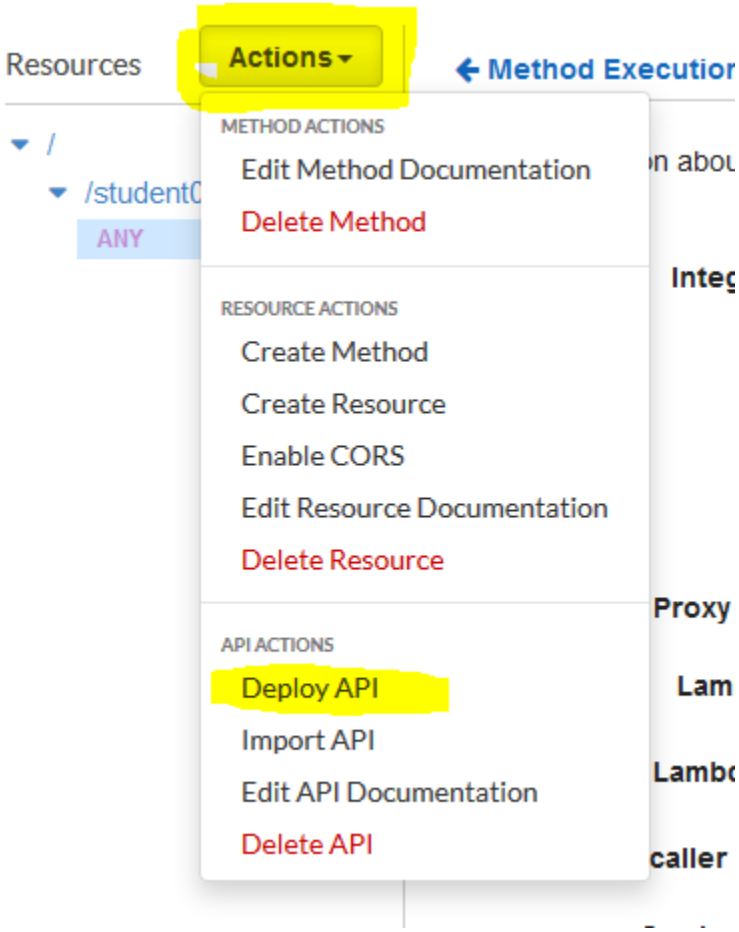
Add the name of your canary function to the end of the function name. The ":" is how you point the gateway towards a specific alias or version number. If you enter the incorrect data you will be shown an error message that just says "unknown error occurred":



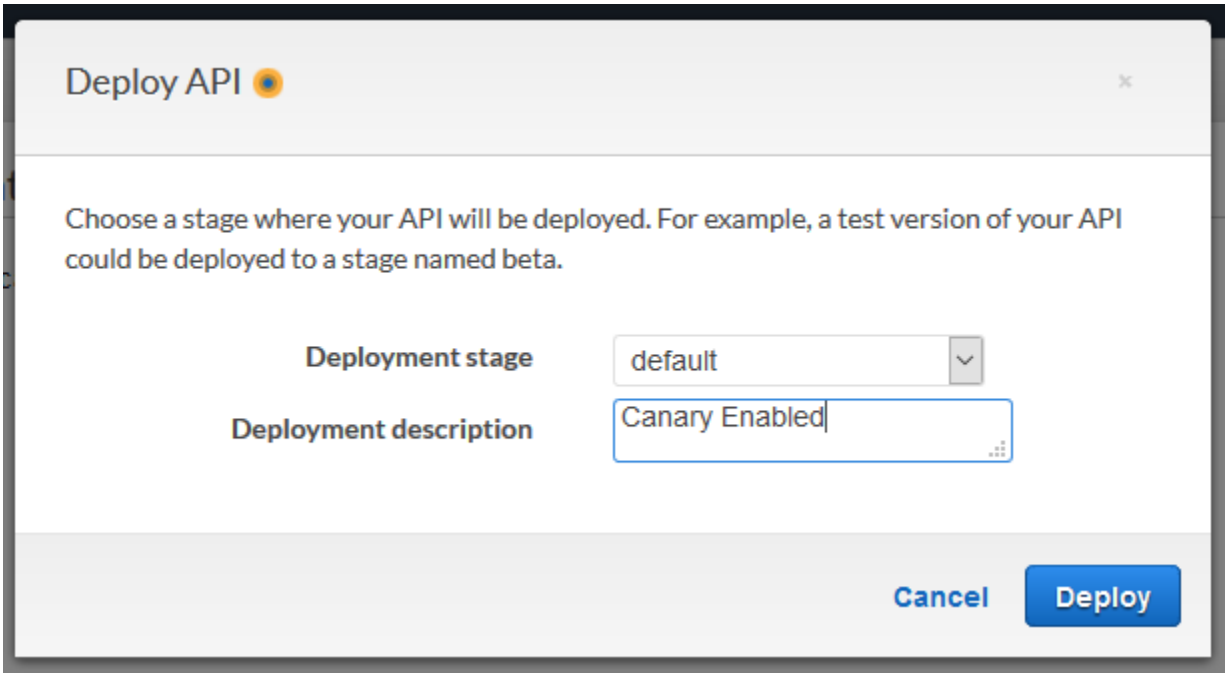
You will be prompted with a warning to add permissions, this allowed the function the be called from the API Gateway. Click "OK":



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:



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.