

Amazon Workflow

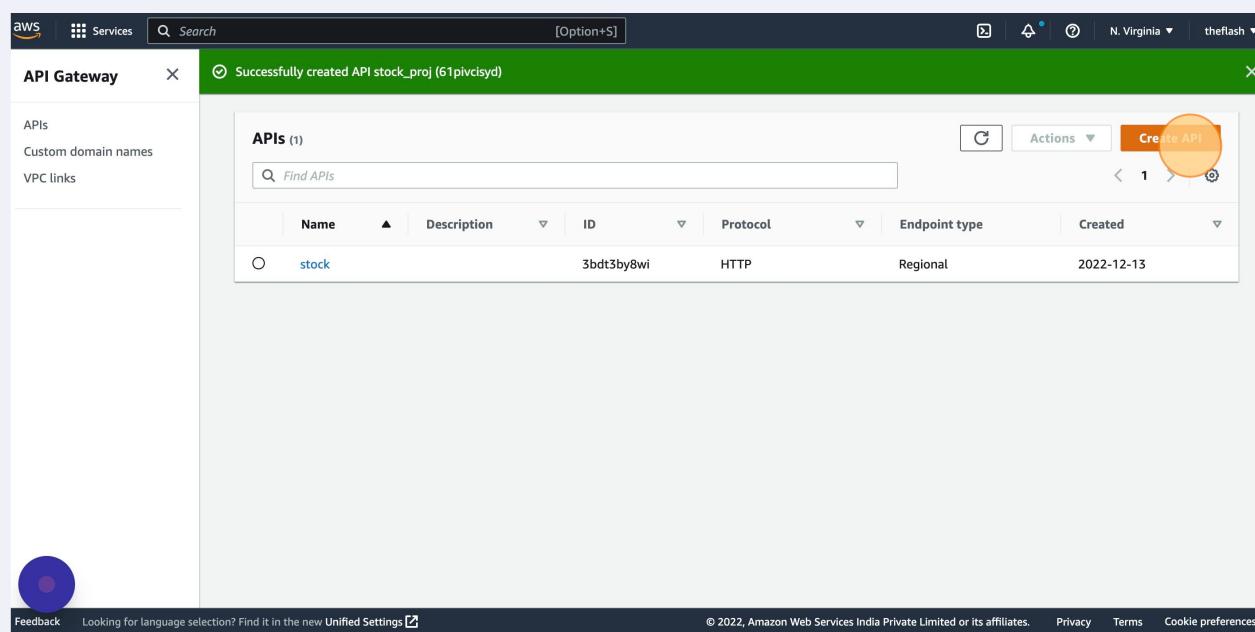
Scribe 

1

Navigate to us-east-1.console.aws.amazon.com/apigateway/main

2

Click "Create API"

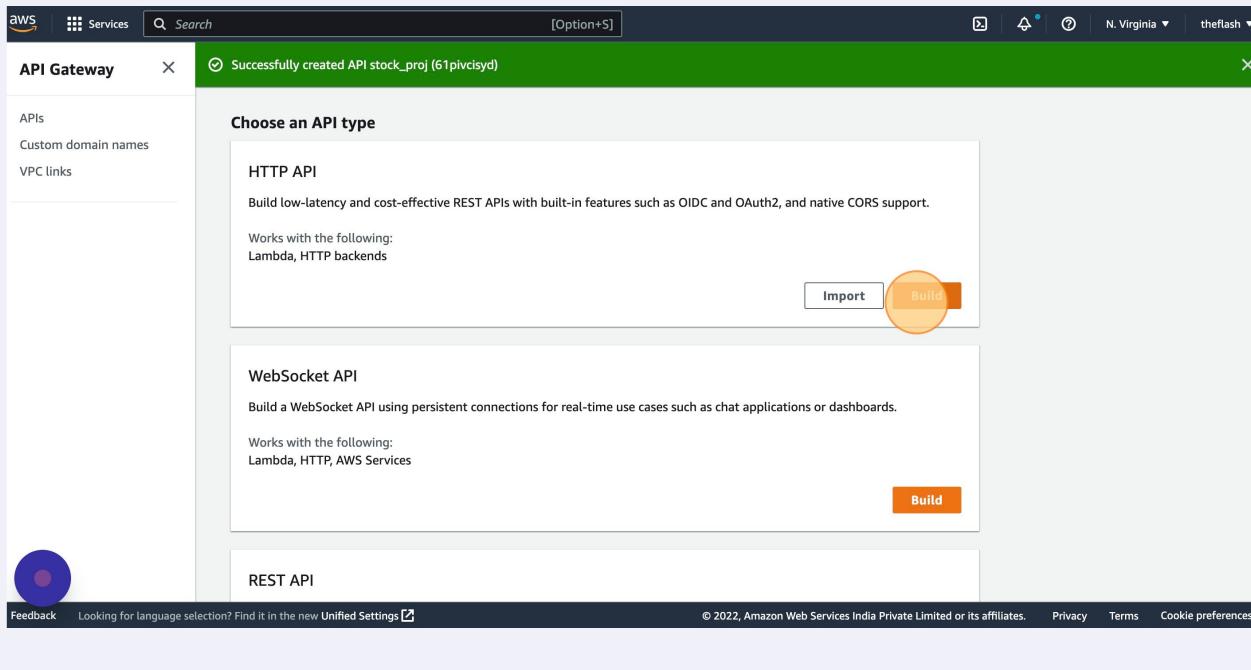


The screenshot shows the AWS API Gateway console. At the top, there's a green success message: "Successfully created API stock_proj (61pivcisyd)". Below it, the "APIs" section displays one entry:

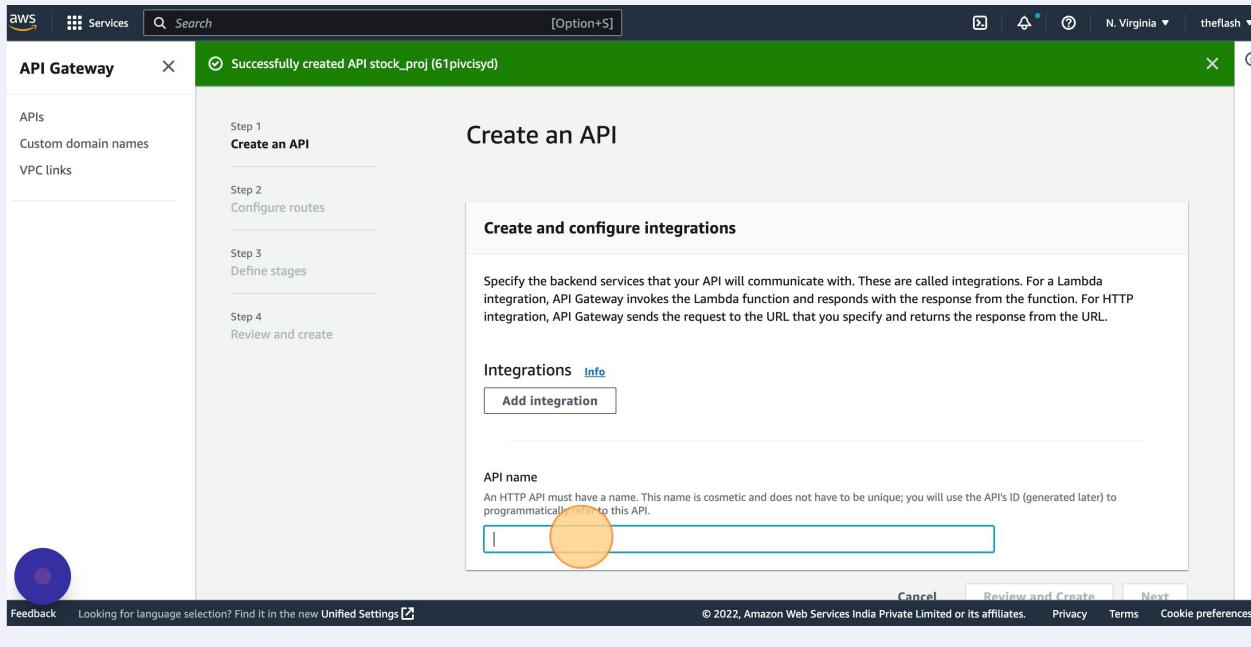
Name	Description	ID	Protocol	Endpoint type	Created
stock		3bdt3by8wi	HTTP	Regional	2022-12-13

A large orange circle highlights the "Create API" button at the top right of the "APIs" table.

3 Click "Build"

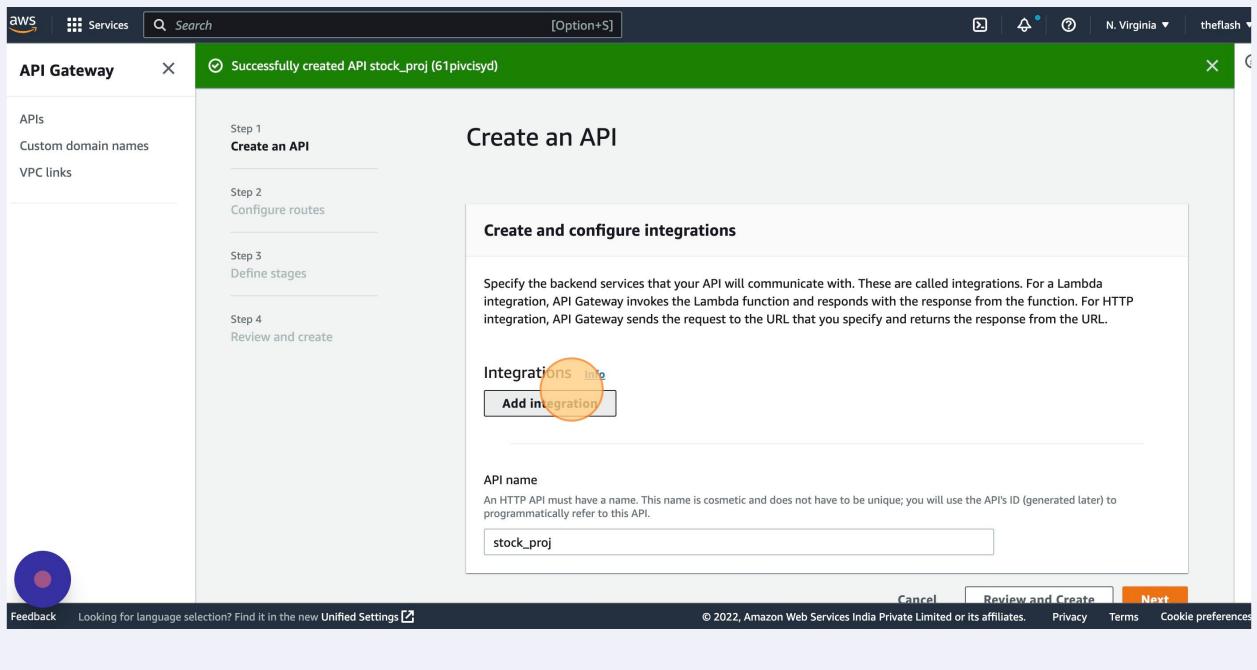


4 Click the "API name" field.

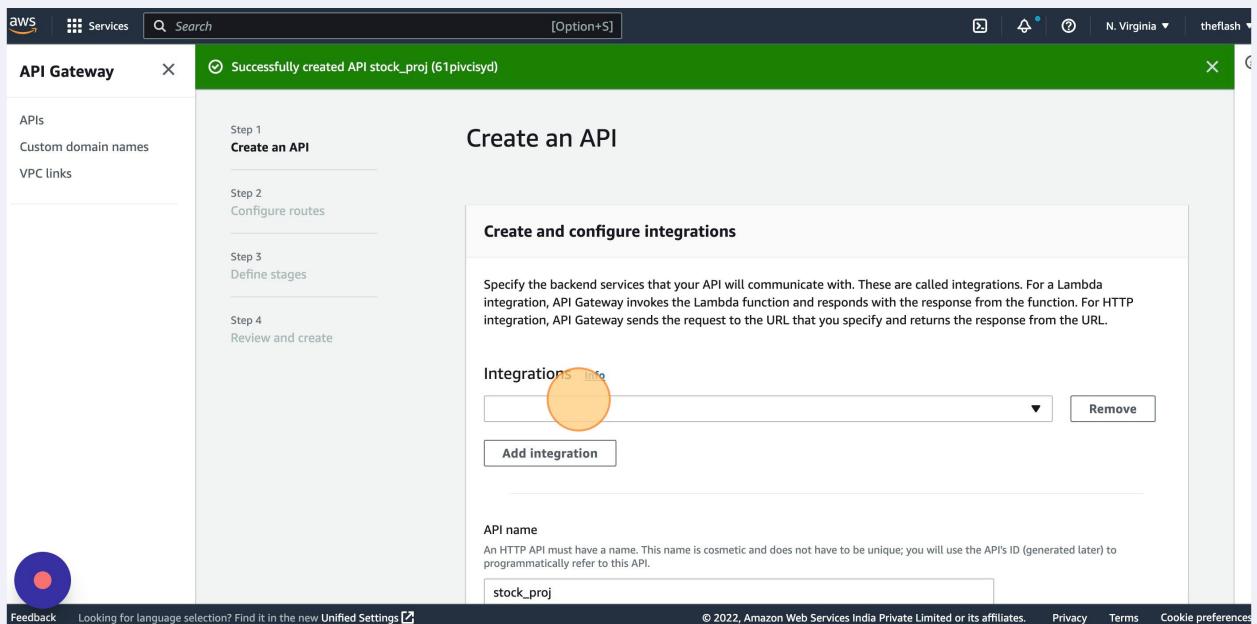


5 Type "stock_proj" or Any other name

6 Click "Add integration"



7 Click here.



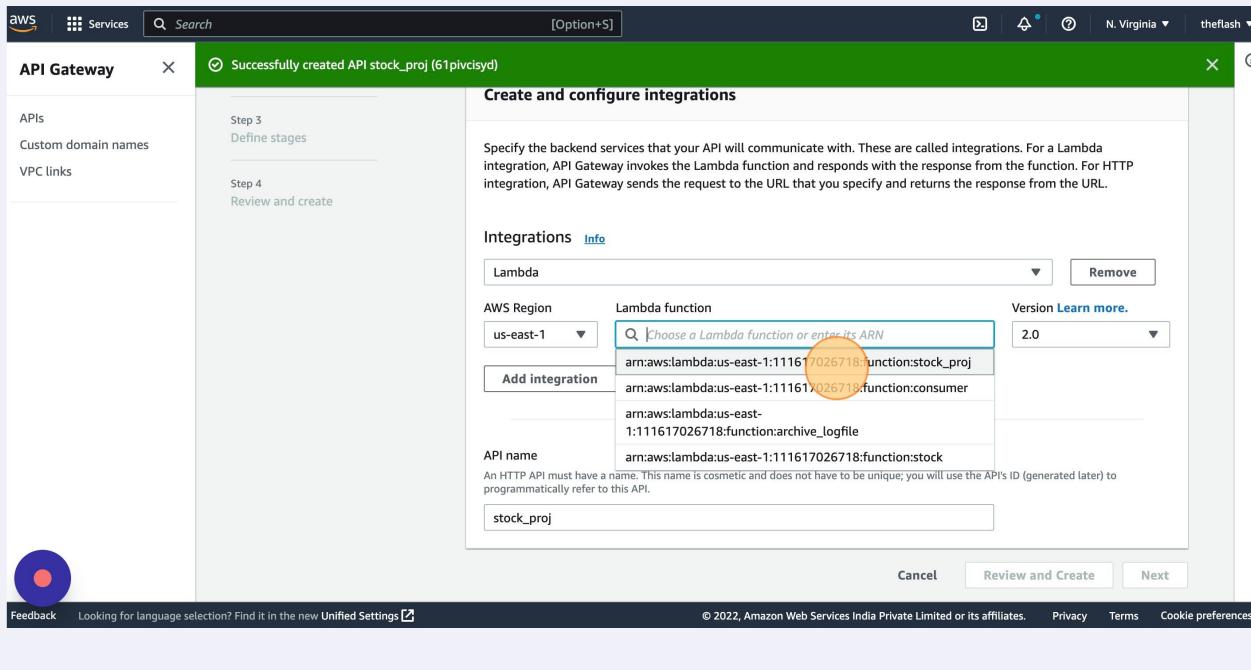
8 Click here.

The screenshot shows the 'Create an API' wizard at Step 1: Create an API. The left sidebar lists 'APIs', 'Custom domain names', and 'VPC links'. The main area has a heading 'Create an API' and a sub-section 'Step 1 Create an API'. It shows a configuration for a Lambda function integration. A callout bubble highlights the 'Lambda' dropdown menu. Below it, there are sections for 'HTTP' and 'Private resource' (ALB, NLB, Cloud Map). A note states: 'The following integrations cannot be set up during API Creation. Finish creating your API and go to Integrations to set this up afterwards.' On the right, there's a note about 'Lambda integrations': 'For a Lambda integration, API Gateway invokes the Lambda function and responds with the response from the function. For HTTP integration, API Gateway sends the request to the URL that you specify and returns the response from the URL.' Buttons for 'Add integration' and 'Remove' are present. At the bottom, the 'API name' is set to 'stock_proj'.

9 Click the "Lambda function" field.

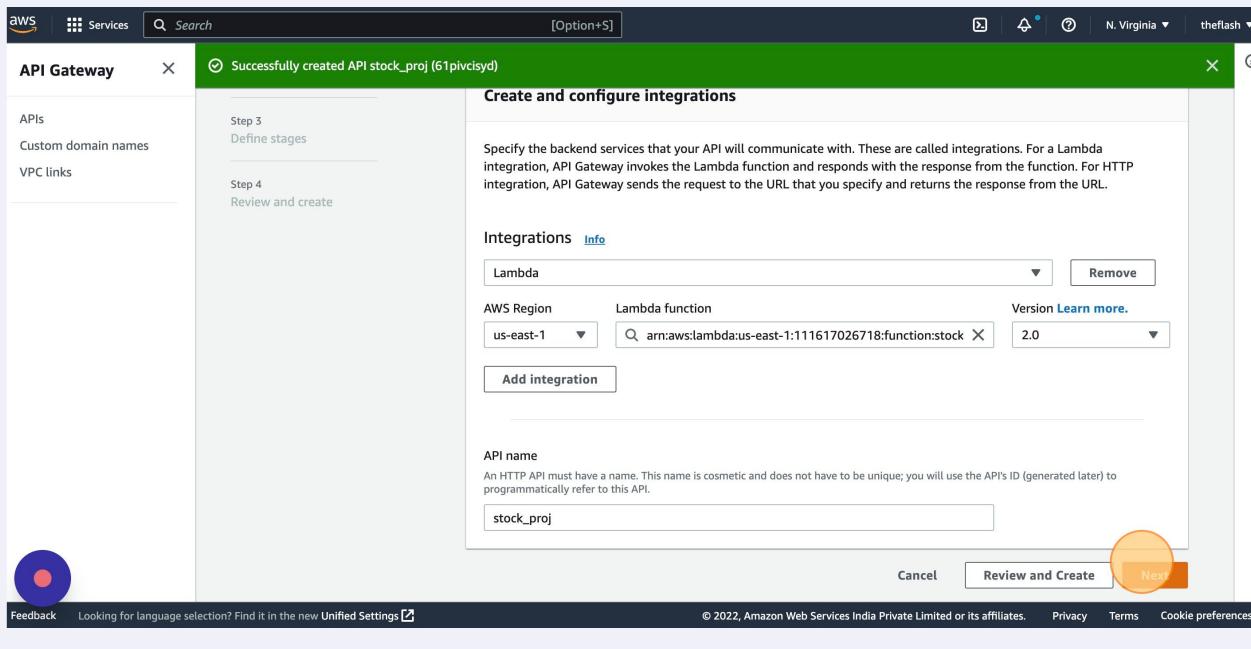
The screenshot shows the 'Create and configure integrations' step 2 of the API creation wizard. The left sidebar lists 'APIs', 'Custom domain names', and 'VPC links'. The main area has a heading 'Create and configure integrations' and a sub-section 'Integrations'. It shows a configuration for a Lambda function integration. A callout bubble highlights the 'Lambda function' dropdown menu. Below it, there are sections for 'AWS Region' (set to 'us-east-1') and 'Version' (set to '2.0'). A note states: 'Specify the backend services that your API will communicate with. These are called integrations. For a Lambda integration, API Gateway invokes the Lambda function and responds with the response from the function. For HTTP integration, API Gateway sends the request to the URL that you specify and returns the response from the URL.' On the right, there's a note about 'Lambda integrations': 'For a Lambda integration, API Gateway invokes the Lambda function and responds with the response from the function. For HTTP integration, API Gateway sends the request to the URL that you specify and returns the response from the URL.' Buttons for 'Add integration', 'Cancel', 'Review and Create', and 'Next' are present. At the bottom, the 'API name' is set to 'stock_proj'.

10 Click "arn:aws:lambda:us-east-1:111617026718:function:stock_proj"



The screenshot shows the AWS API Gateway interface. The top navigation bar includes the AWS logo, Services, Search, and Option+S. The main title is 'Successfully created API stock_proj (61picvcsyd)'. On the left sidebar, there are links for APIs, Custom domain names, and VPC links. The main content area is titled 'Create and configure integrations'. It explains that integrations allow your API to communicate with backend services like Lambda functions. Below this, the 'Integrations' section shows a dropdown set to 'Lambda'. The 'Lambda function' dropdown contains several options, with 'arn:aws:lambda:us-east-1:111617026718:function:stock_proj' selected and highlighted with a yellow circle. Other options listed include 'arn:aws:lambda:us-east-1:111617026718:function:consumer' and 'arn:aws:lambda:us-east-1:111617026718:function:archive_logfile'. The 'API name' field is set to 'stock_proj'. At the bottom right of the integration form are 'Cancel', 'Review and Create', and 'Next' buttons.

11 Click "Next"



This screenshot shows the same 'Create and configure integrations' step as the previous one, but the Lambda function dropdown now shows 'arn:aws:lambda:us-east-1:111617026718:function:stock' instead of the previously selected option. The rest of the interface remains the same, including the integration details and the 'Next' button highlighted with a yellow circle.

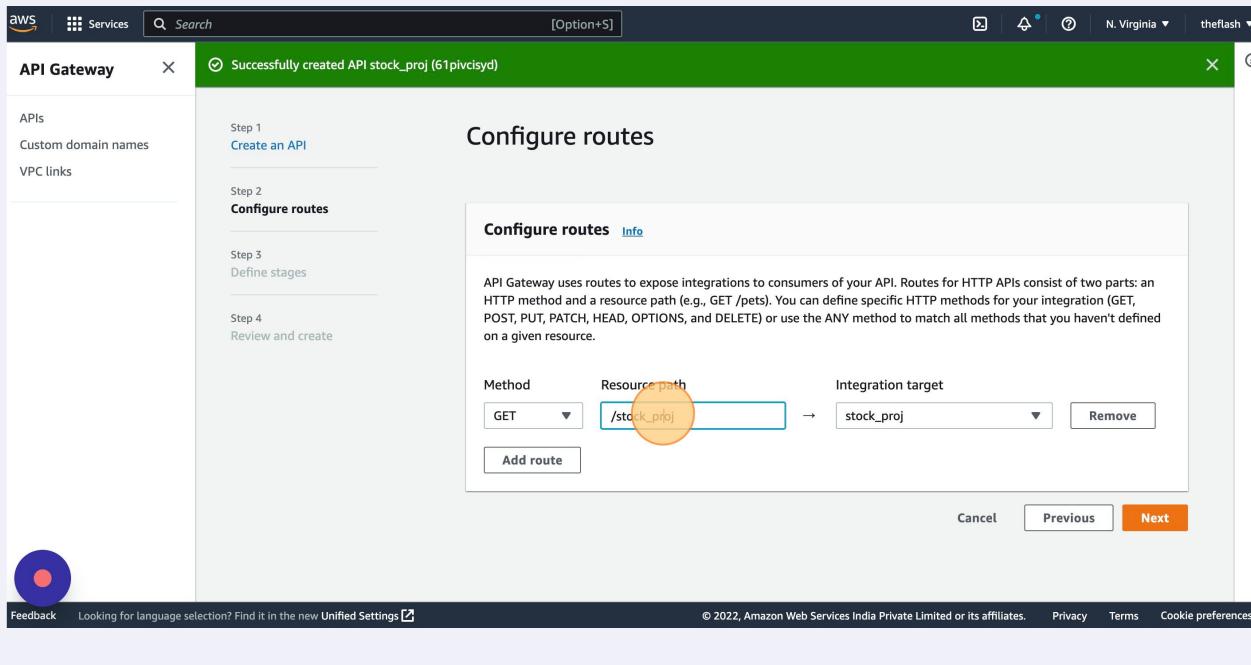
12 Click here.

The screenshot shows the AWS API Gateway configuration interface. The top navigation bar includes the AWS logo, services dropdown, search bar, and region selection (N. Virginia). The main title is 'Successfully created API stock_proj (61pivcisyd)'. On the left sidebar, there are links for APIs, Custom domain names, and VPC links. The main content area is titled 'Configure routes' and shows Step 2: 'Configure routes'. A sidebar on the right lists HTTP methods: GET, POST, PUT, PATCH, HEAD, OPTIONS, DELETE, and ANY. The 'ANY' method is currently selected and highlighted with a blue background. Below the method list, there are fields for 'Resource path' (set to '/stock_proj') and 'Integration target' (set to 'stock_proj'). A large orange circle highlights the 'ANY' method button. At the bottom of the page, there are 'Cancel', 'Previous', and 'Next' buttons, along with copyright information and links for feedback, language selection, privacy, terms, and cookie preferences.

13 Click here.

This screenshot is identical to the one above, showing the 'Configure routes' step in the AWS API Gateway. The 'ANY' method is still selected and highlighted with a blue background. However, in this version, the 'GET' method is also highlighted with an orange circle, indicating it has been selected or is the active method. The rest of the interface, including the sidebar with other methods, the resource path, integration target, and footer links, remains the same.

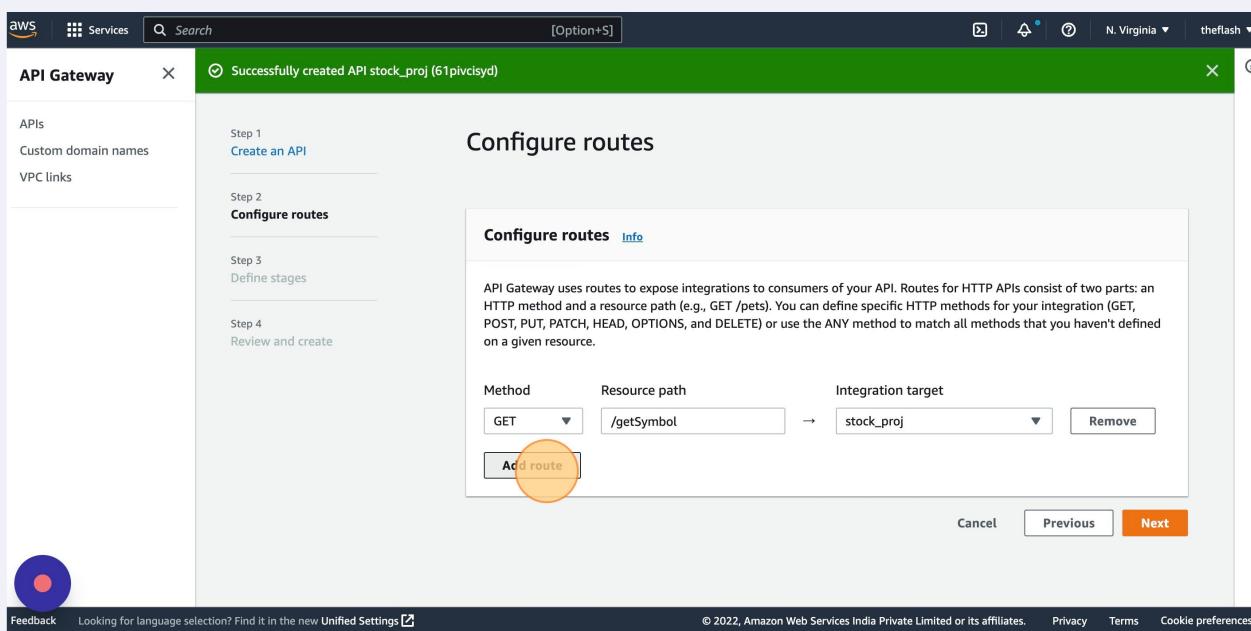
14 Double-click the "/some/path/parts" field.



The screenshot shows the AWS API Gateway configuration interface. On the left sidebar, there are links for APIs, Custom domain names, and VPC links. The main panel is titled "Configure routes" (Step 2). It contains a sub-section titled "Configure routes" with an "Info" link. Below this, a detailed description explains that API Gateway uses routes to expose integrations to consumers of your API. Routes for HTTP APIs consist of two parts: an HTTP method and a resource path (e.g., GET /pets). You can define specific HTTP methods for your integration (GET, POST, PUT, PATCH, HEAD, OPTIONS, and DELETE) or use the ANY method to match all methods that you haven't defined on a given resource. The configuration area shows a route being defined: Method is set to GET, Resource path is set to /stock_proj (which is highlighted with a yellow circle), and the Integration target is set to stock_proj. There are "Add route" and "Remove" buttons below the route definition. At the bottom right of the main panel are "Cancel", "Previous", and "Next" buttons. The top navigation bar includes the AWS logo, services menu, search bar, and account information for N. Virginia.

15 Type "getSymbol"

16 Click "Add route"



This screenshot is identical to the one above it, showing the "Configure routes" step 2 of the AWS API Gateway configuration wizard. The route definition is identical: Method is GET, Resource path is /getSymbol (highlighted with a yellow circle), and Integration target is stock_proj. The "Add route" button at the bottom of the configuration panel is also highlighted with a yellow circle. The rest of the interface, including the sidebar, main description, and navigation buttons, remains the same.

17

Click "-"

The screenshot shows the AWS API Gateway configuration interface. The top bar indicates a successful creation of an API named 'stock_proj'. The left sidebar lists 'APIs', 'Custom domain names', and 'VPC links'. The main area is titled 'Configure routes' and shows Step 2: 'Configure routes'. On the left, a dropdown menu for HTTP methods includes 'GET', 'POST', 'PUT', 'PATCH', 'HEAD', 'OPTIONS', 'DELETE', and 'ANY'. The 'GET' option is currently selected. To its right, there's a table for defining routes. The first route is defined as 'Resource path /getSymbol' pointing to 'Integration target stock_proj'. Below this table is an 'Add route' button. At the bottom of the page are 'Cancel', 'Previous', and 'Next' buttons.

18

Click here.

This screenshot is identical to the previous one, showing the 'Configure routes' step in the AWS API Gateway. The 'POST' method is now highlighted with a circle. The rest of the interface, including the table for routes and the navigation buttons at the bottom, remains the same.

19 Click the "/some/path/parts" field.

The screenshot shows the AWS API Gateway configuration interface. On the left sidebar, under 'Step 2 Configure routes', there are four steps: Step 1 Create an API, Step 2 Configure routes (which is active), Step 3 Define stages, and Step 4 Review and create. The main content area is titled 'Configure routes' and contains a detailed description of routes. Below the description, there are two route entries. The first route is for 'GET' method with resource path '/getSymbol' and integration target 'stock_proj'. The second route is for 'POST' method with resource path '/some/path/parts' and integration target 'stock_proj'. The 'parts' part of the POST route's path is highlighted with a blue selection bar and has an orange circle around it, indicating it is the target for step 19.

20 Type "/createSymbol"

21 Click here.

This screenshot is identical to the one above, showing the 'Configure routes' step 2 page. It displays the same two routes: a GET route to /getSymbol and a POST route to /some/path/parts. The focus is on the second POST route. The integration target dropdown for the POST route is open, showing the options 'stock_proj' and 'Remove'. An orange circle is drawn around the 'stock_proj' option, indicating it is the target for step 21.

22 Click here.

The screenshot shows the AWS API Gateway configuration interface. The top navigation bar includes the AWS logo, Services, a search bar, and a user icon. The main title is "Successfully created API stock_proj (61pivcisyd)". On the left sidebar, there are links for APIs, Custom domain names, and VPC links. The main content area is titled "Configure routes". It displays two route configurations:

Method	Resource path	Integration target
GET	/getSymbol	stock_proj
POST	/createSymbol	stock_proj

Below the routes, there are buttons for "Add route", "Cancel", "Previous", and "Next". The "Next" button is highlighted with a yellow circle.

23 Click "Next"

This screenshot is identical to the previous one, showing the "Configure routes" step of the API creation wizard. The routes and interface elements are the same, with the "Next" button highlighted by a yellow circle.

24

Click "Next"

The screenshot shows the 'Define stages' step of creating an API. On the left sidebar, there are links for APIs, Custom domain names, and VPC links. The main area has four steps: Step 1 (Create an API), Step 2 (Configure routes), Step 3 (Define stages), and Step 4 (Review and create). Step 3 is currently active. A 'Configure stages' section contains a note about stages being independently configurable environments. It shows a stage named '\$default' with an 'Auto-deploy' toggle switch turned on. Below the stage name is a 'Remove' button and an 'Add stage' button. At the bottom right of the stage configuration box are 'Cancel', 'Previous', and 'Next' buttons, with 'Next' being highlighted with a yellow circle.

25

Click "Create"

The screenshot shows the 'Review and create' step of creating an API. The left sidebar shows APIs, Custom domain names, and VPC links. The main area displays the summary of the created API: API name 'stock_proj', Integration 'stock_proj (Lambda)', and Routes (GET /getSymbol and POST /createSymbol both mapped to stock_proj Lambda function). Below the summary are sections for 'Routes' (with an 'Edit' button) and 'Stages' (with an 'Edit' button). The 'Stages' section shows '\$default (Auto-deploy: enabled)'. At the bottom right are 'Cancel', 'Previous', and 'Create' buttons, with 'Create' being highlighted with a yellow circle.

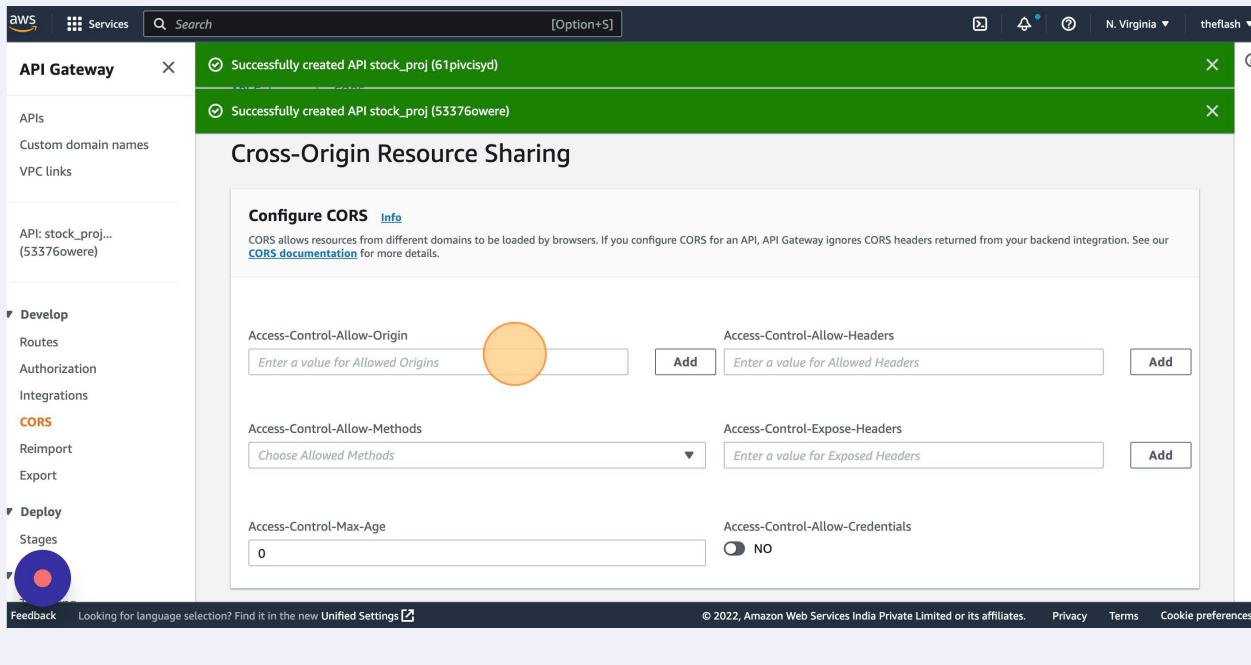
26 Click "CORS"

The screenshot shows the AWS API Gateway console. On the left, there's a sidebar with options like APIs, Custom domain names, VPC links, and a section for the selected API: stock_proj (53376owere). Under 'Develop', the 'CORS' option is highlighted with a red circle. The main content area shows the 'stock_proj' API details, including its ID (53376owere), protocol (HTTP), and creation date (2022-12-21). Below this is a table for stages, with one stage listed: 'dev' (version 1.0.0, created 2022-12-21). At the bottom right of the main area is a 'Deploy' button.

27 Click "Configure"

The screenshot shows the 'Cross-Origin Resource Sharing' configuration page for the 'stock_proj' API. The 'Configure CORS' button is highlighted with a red circle. The page contains sections for Access-Control-Allow-Origin (No Origins are allowed), Access-Control-Allow-Methods (No Methods are allowed), Access-Control-Max-Age (0 Seconds), Access-Control-Allow-Headers (No Headers are allowed), Access-Control-Expose-Headers (No Expose Headers are allowed), and Access-Control-Allow-Credentials (NO).

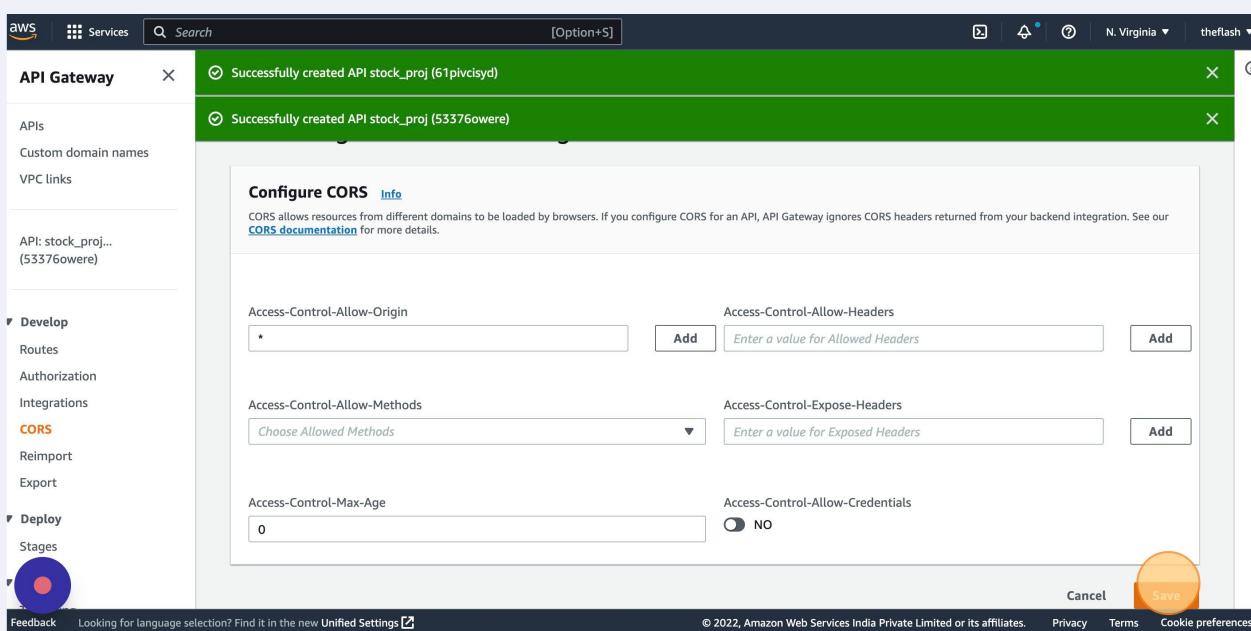
28 Click the "Access-Control-Allow-OriginAdd" field.



The screenshot shows the AWS API Gateway CORS configuration page. The left sidebar has sections for APIs, Develop (Routes, Authorization, Integrations, CORS), and Deploy (Stages). The main area is titled 'Cross-Origin Resource Sharing' and contains a 'Configure CORS' section. The 'Access-Control-Allow-Origin' input field is highlighted with an orange circle. Other fields include 'Access-Control-Allow-Headers', 'Access-Control-Allow-Methods', 'Access-Control-Expose-Headers', 'Access-Control-Max-Age', and 'Access-Control-Allow-Credentials'. A note at the bottom says 'CORS allows resources from different domains to be loaded by browsers. If you configure CORS for an API, API Gateway ignores CORS headers returned from your backend integration. See our CORS documentation for more details.'

29 Type "*"

30 Click "Save"



The screenshot shows the same AWS API Gateway CORS configuration page after saving. The 'Access-Control-Allow-Origin' input field now contains the value '*' (asterisk). The 'Save' button at the bottom right is highlighted with an orange circle. The rest of the configuration fields remain the same as in the previous step.

31 Click the "Search" field.

The screenshot shows the AWS API Gateway CORS configuration page. At the top, there are two green notifications: "Successfully created API stock_proj (61pivcisyd)" and "Successfully created API stock_proj (53376owere)". Below the notifications, the title "Cross-Origin Resource Sharing" is displayed. Under the "Configure CORS" section, there are four pairs of settings:

Header	Value
Access-Control-Allow-Origin	No Origins are allowed
Access-Control-Allow-Methods	No Methods are allowed
Access-Control-Max-Age	0 Seconds
Access-Control-Expose-Headers	No Expose Headers are allowed
Access-Control-Allow-Credentials	NO

At the bottom right of the configuration area are "Configure" and "Clear" buttons. The left sidebar shows navigation options like APIs, Custom domain names, VPC links, and sections for Develop (Routes, Authorization, Integrations, CORS) and Deploy (Stages). A feedback link and language selection information are at the bottom left, and copyright and privacy information are at the bottom right.

32 Search "lambda" service

33 Click "Lambda"

The screenshot shows the AWS Services search interface. The search bar at the top contains the text 'lambda'. Below the search bar, the results are categorized under 'Services'. The first result is 'Lambda', which is highlighted with a yellow circle. The Lambda entry includes the description 'Run Code without Thinking about Servers'. Other services listed include 'CodeBuild', 'AWS Signer', and 'Amazon Inspector'. To the right of the search results, there is a green sidebar with a 'Configure' button and a 'Clear' button.

34 Click "stock_proj"

The screenshot shows the AWS Lambda service interface. The left sidebar has a 'Functions' section selected. The main area displays a table titled 'Functions (4)'. The table lists four functions: 'stock_proj', 'consumer', 'archive_logfile', and 'stock'. The 'stock_proj' function is highlighted with a yellow circle. The table columns are 'Function name', 'Description', 'Package type', and 'Runtime'. The 'stock_proj' function is listed as 'Zip' package type and 'Python 3.8' runtime.

	Function name	Description	Package type	Runtime
<input type="checkbox"/>	stock_proj	-	Zip	Python 3.8
<input type="checkbox"/>	consumer	-	Zip	Python 3.8
<input type="checkbox"/>	archive_logfile	-	Zip	Python 3.9
<input type="checkbox"/>	stock	-	Zip	Python 3.8

35

Click "API Gateway"

The screenshot shows the AWS Lambda Functions interface. In the center, there's a card for a function named "stock_proj". The "Function overview" section is expanded, showing a diagram where "API Gateway" is highlighted with a large orange circle. Below the diagram, there are buttons for "+ Add trigger" and "+ Add destination". To the right of the diagram, there are sections for "Description", "Last modified", "Function ARN", and "Function URL". At the bottom of the card, there are tabs for "Code", "Test", "Monitor", "Configuration", "Aliases", and "Versions". The "Code" tab is currently selected. At the very bottom of the page, there are links for "Feedback", "Upload from", and copyright information.