**API Gateway Canary Deployments**

API Canary Deployments are a technique for gradually rolling out new versions of your API to a limited set of users before making it available to everyone. This allows you to test the new version in a controlled environment and identify any issues before impacting your entire user base. Here's what an AWS Developer Associate should know about them:

* **Concept:**
  + Canary deployments involve releasing a new version of your API (often backed by AWS Lambda functions) to a small subset of users while keeping the existing version running for the majority.
  + This "canary" version acts as an early detection system for potential problems.
* **Benefits:**
  + **Reduced Risk:** Issues in the new version are exposed to a limited audience, minimizing potential impact.
  + **Improved Quality:** Real-world testing with actual users helps identify bugs and performance issues before full rollout.
  + **Faster Feedback:** Canary deployments provide quicker insights into user experience with the new version.
* **Implementation with API Gateway:**
  + API Gateway allows canary deployments on any stage, including production.
  + You can control the percentage of traffic routed to the canary version.
  + Separate metrics and logs for canary and production versions enable performance comparison and issue identification.
  + Stage variables can be overridden for the canary deployment to facilitate specific testing configurations.
* **Blue/Green Deployment Analogy:**
  + Canary deployments essentially mimic a Blue/Green deployment strategy.
    - Blue represents the production version.
    - Green represents the canary version with the new code.
  + Traffic is split between Blue and Green based on the defined percentage.

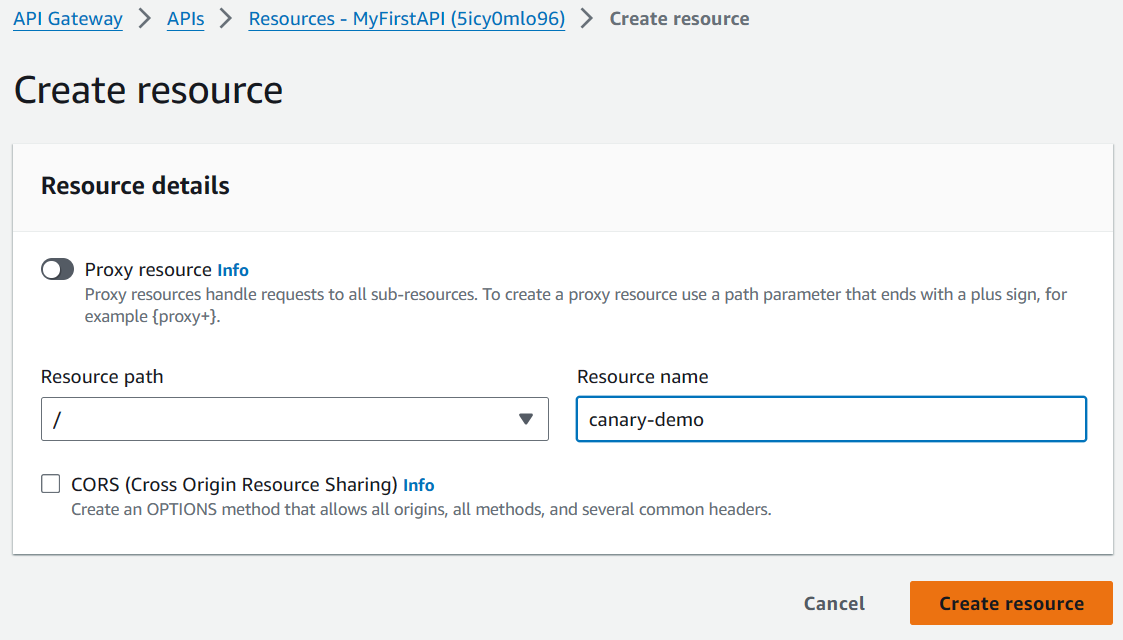
**API Gateway Canary Deployment Hands-On Lab**

**Prerequisites:**

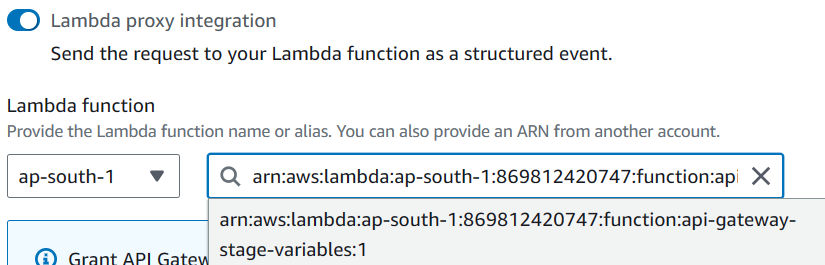
* An existing API in API Gateway with a Lambda function (api-gateway-stage-variables) with two versions (v1 and v2).

**Steps:**

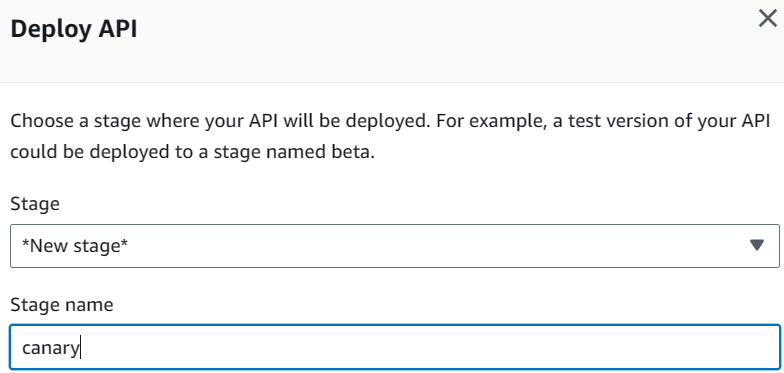
1. **Create a Canary Resource:**
   * In your API Gateway console, navigate to **Resources**.
   * Create a new resource named canary-demo.



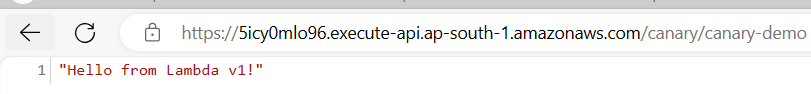
1. **Create a Canary Method:**
   * Under the canary-demo resource, create a new **Method** of type GET.
   * Choose **Proxy Integration** as the integration type.
   * For the integration target, select the api-gateway-stage-variables Lambda function.
2. **Specify Lambda Function Version (v1):**
   * In the integration request, add a path parameter named {version}.
   * Set the value of the path parameter to :1. This specifies version 1 of the Lambda function.



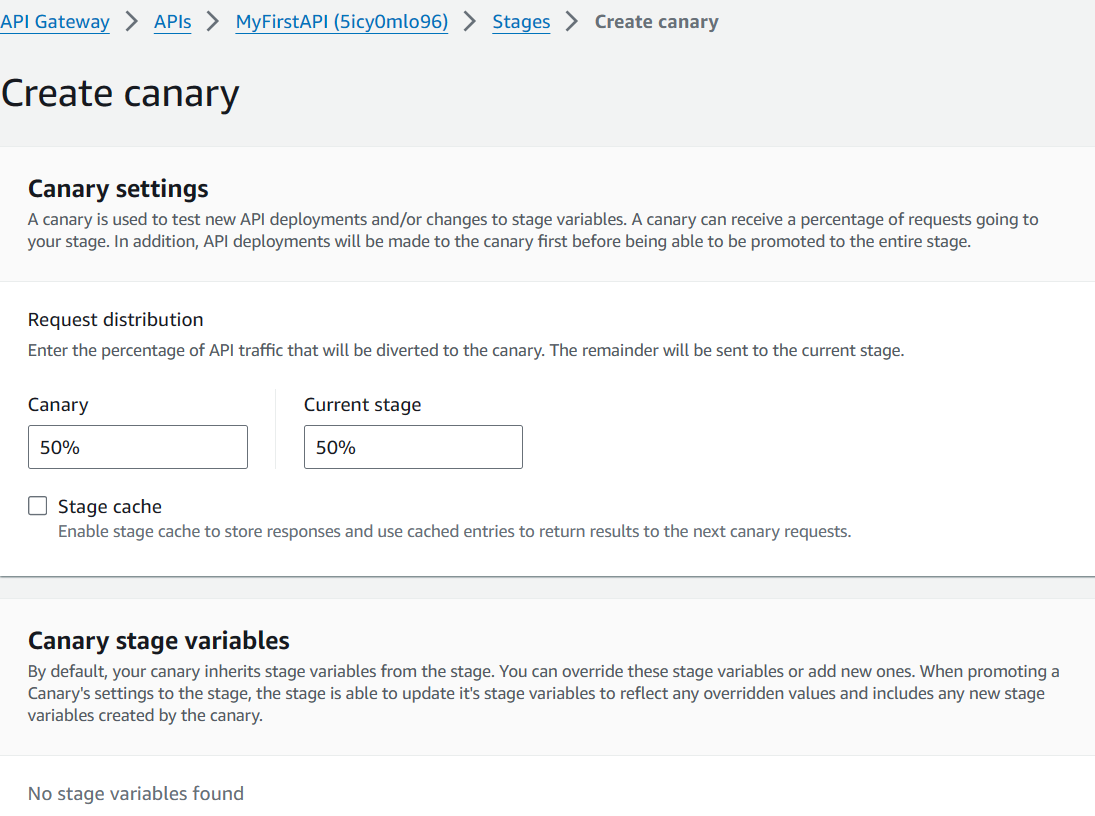
1. **Test the Canary Method:**
   * Deploy the API to a new stage called canary.



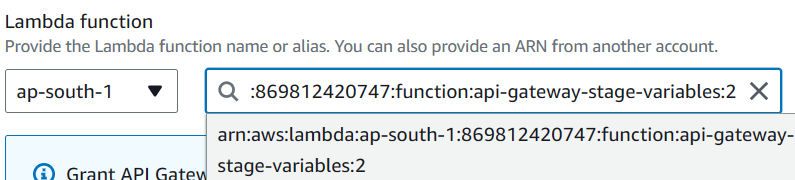
* + Invoke the canary URL (usually in the format <API Gateway endpoint>/canary/canary-demo). You should receive the response "Hello from Lambda v1".



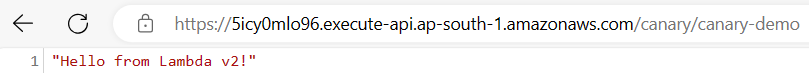
1. **Configure Canary Deployment:**
   * In the API Gateway console, navigate to **Stages**.
   * Select the existing stage (where your main API resides).
   * Click on **Canaries**.
   * Define a new canary deployment with a traffic distribution of 50% for the canary (canary) stage and 50% for the current stage. (In production, you would typically use a smaller percentage for the canary initially.)



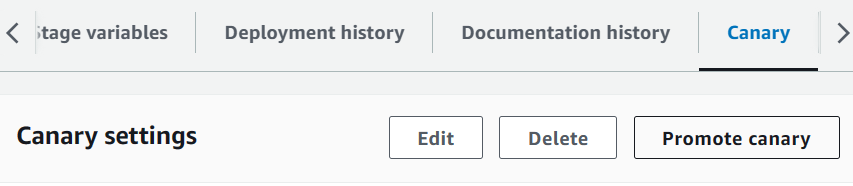
1. **Deploy v2 to Canary Stage:**
   * Go back to **Resources**.
   * Under the canary-demo method, edit the integration request.
   * Change the path parameter value to :2 to target version 2 of the Lambda function.
   * Save the changes.

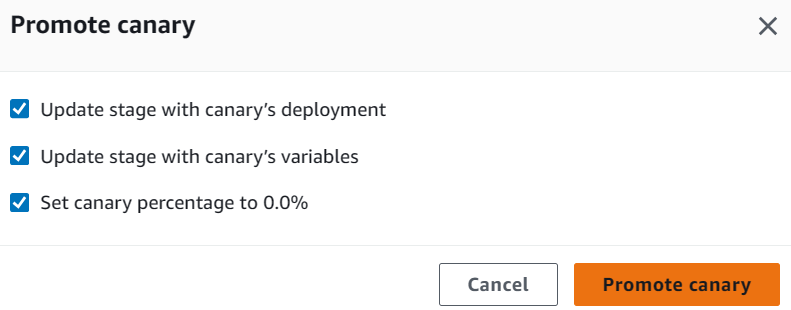


1. **Test v2 in Canary Stage:**
   * Deploy the API to the canary stage again.
   * Access the canary URL. You should now see "Hello from Lambda v2" intermittently, indicating the canary is switching between versions.



1. **Promote Canary (Optional):**
   * Once you're satisfied with the new version (v2) in the canary stage, you can promote it to become the main version.
   * In the **Stages** section, select the canary stage.
   * Click on **Actions** and choose **Promote**. This will update the main stage with the canary deployment, directing all traffic to version 2.





**Verifying Promotion:**

* After promotion, accessing the main API endpoint should consistently return "Hello from Lambda v2".