Publish and verify an API Gateway route

Add the method to the public contract, preserve backend authentication, and validate routing safely.

https://docs.qintoravacloud.com/guides/publish-gateway-route/

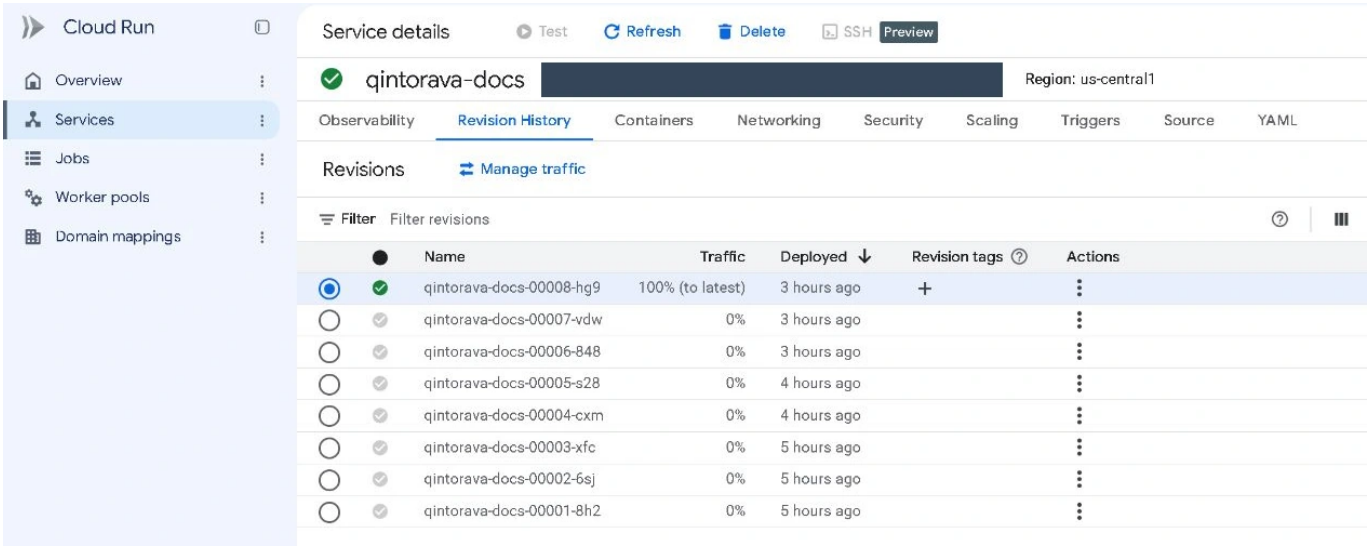
Register the operation

1. Confirm the backend handler exists and enforce its application-level role checks. In `openapi.yaml`, place the route beside related operations. Require path parameters and give the operation a unique `operationId`.
2. Describe the successful result and the most important client errors. Match the method and path to the backend. Keep the configured Cloud Run service as the backend and use the existing gateway service account.

```
/api/purchase-orders/{id}/reconcile:
  parameters:
    - { in: path, name: id, required: true, type: string, format: uuid }
  post:
    operationId: reconcileAssetPurchaseOrder
    responses:
      "200": { description: Original ERP dispatch reconciled }
      "401": { description: Application session required }
      "403": { description: Platform administrator required }
      "409": { description: Order or provider evidence is ineligible }
```

Release the config and probe routing

1. Render the backend URL into the spec with the repository’s release process. Compare the new routes with the active config and retain existing routes required by clients.
2. Create and wait for the new API config to become active. Update the intended gateway to use it. Record the previous and new config names so the operator can restore the previous config if validation fails.
3. Probe the route without credentials and with a synthetic UUID. A `401` from Qintorava shows the route reached the application. Test authenticated role denial, an invalid order, and valid provider proof only in the disposable environment.



Google Cloud Console screenshot. Use Revision History to confirm the backend revision and its traffic before probing the new route.

```
curl -sS -i -X POST -H 'Content-Type: application/json' --data '{}' \
  https://demo.qintoravacloud.com/api/purchase-orders/00000000-0000-4000-8000-000000000000/
reconcile
```

For the Google Cloud commands, see Create an API config. For application results, see ERP reconciliation.