digital Fingers

# SmartInn

# Design and Process on

Google Cloud

## Brief description

SmartInn got more than 200 locations that spread across North America and Asia. They have a website that contains a daily blog and social integration service initiative which shows customer reviews on the quality of SmartInn’s services

## Main features

- Responsive website with blog (multiple language support)
- Revision management enabled for service deployments
- Multiple language support for website
- Translation of customer reviews
- Increased velocity of service deployments
- New revision retains existing level of operational stability by deployment to a reduced user base

## Roles of typical user

- Customer
- Ops lead
- Developer lead
- Product lead
- Marketing lead

SmartInn

# User personas

## Christy

Christy is a busy Site Engineer who likes to fully utilize the time at her disposal, she prefer to order food via online for herself and workers, often booked at specific times in a day. Customer feedback play a major role in the choice Christy makes. Christy likes to perform all operations from her phone

# User personas

## Sam

Sam is a student who likes to travel home to visit parents and also takes vacations twice yearly. His primary concern is cost, and he will always book the lowest price travel regardless of convenience. Sam has no loyalty and will use whichever retailer can provide the best deal.

# User stories

## Website remains responsive

**As a** product lead**, I want** to ensure the website remains responsive, **so that** customers face minimal wait times

# User stories

## Velocity of deployments

**As a** developer lead, **I want** to increase the velocity of service deployments.

# User stories

## System performance

**As an** ops lead, **I want** to ensure system stability is observed, **so the** system performance is not degraded through the deployment of new revisions.

# User stories

## Translation of reviews

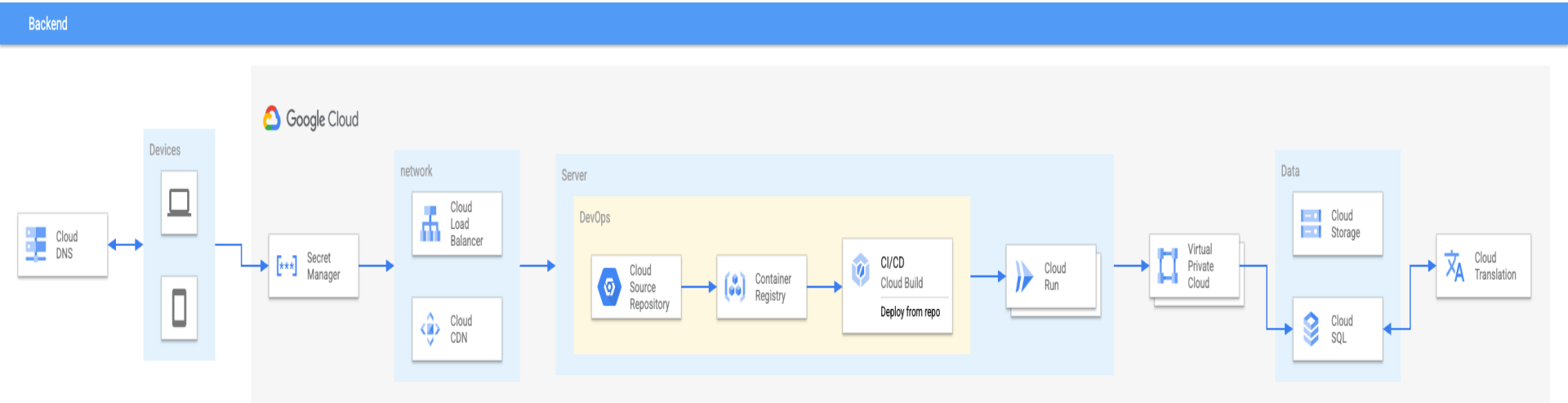
**As a** manager, **I want** to ensure customers' reviews are translated, **so that** I can understand their pain point, server them better and build loyalty

# User stories

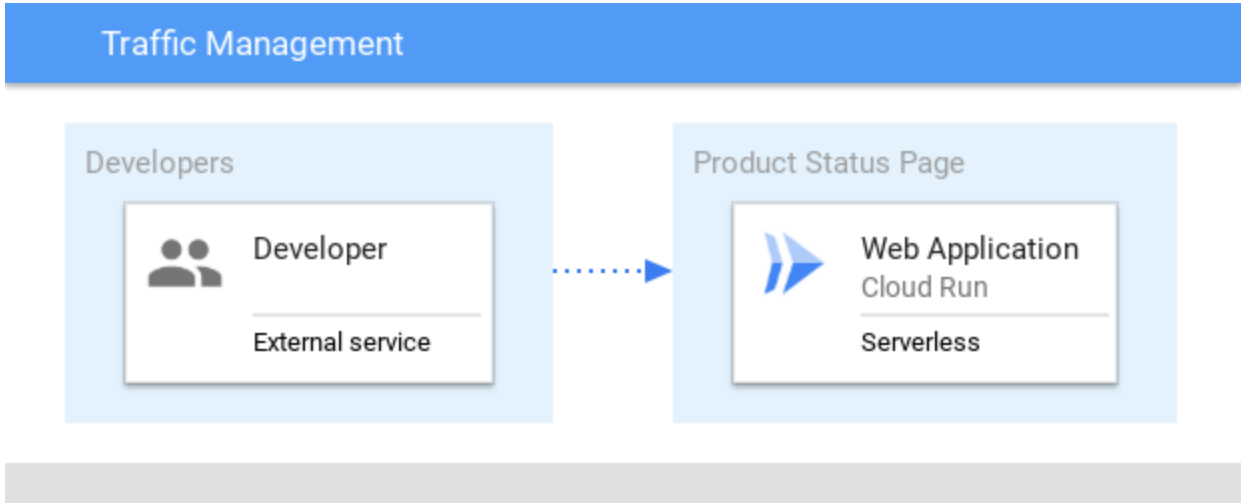
## Website support multi-languages

**As a** marketing lead, **I want** to ensure the website supports at least three languages, **so that** we can gain more visitors and blog readers and enhance our publicity

# Architecture Overview



# MVP Architecture




In the proposed solution, the product service is used to demonstrate how traffic migration and revision tagging can be used with Cloud Run.

# Definition of Done

| Ref | Definition of Done |
|---|---|
| 1 | User base are not impacted by the rollout of new features |
| 2 | Revision management enabled for service deployments |
| 3 | New revision retains existing level of operational stability by deployment to a reduced user base |

# Traffic migration versus revision tags

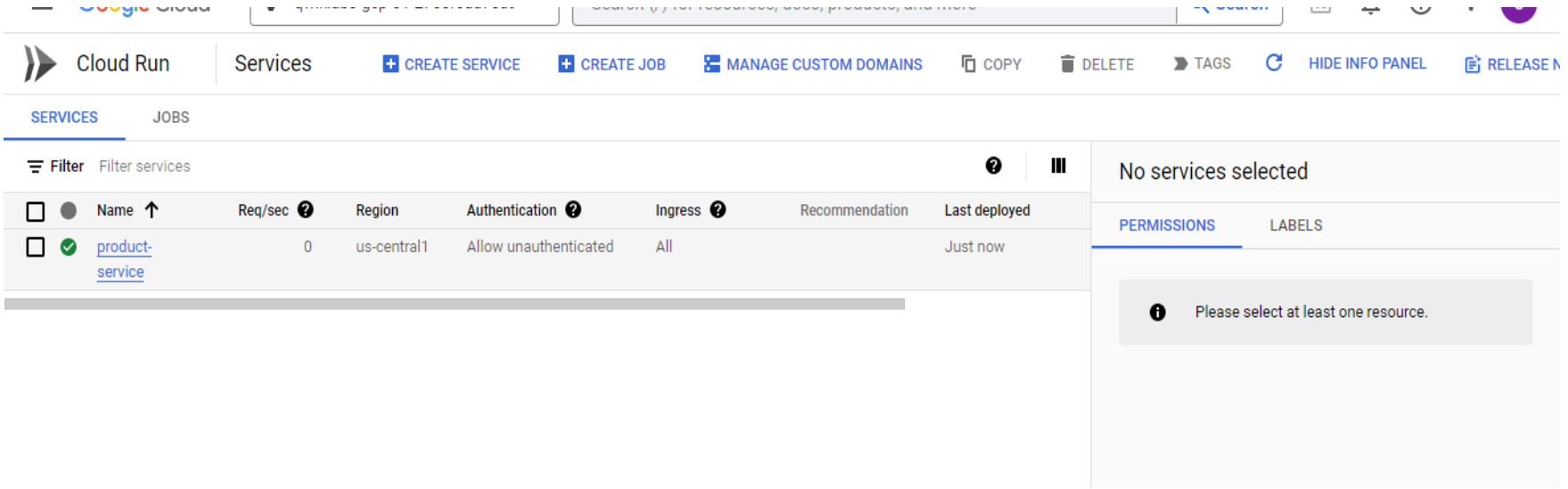
| Feature | Description | |
|---|---|---|
| *Revision Tags* | *"Appropriate for use cases where a task producer needs to defer or control the execution timing of a specific webhook or remote procedure call."* |  |
| Traffic Migration | "Cloud Run allows you to specify which revisions should receive traffic and to specify traffic percentages that are received by a revision.." |  |

# Developing a minimal viable product (MVP)

To build an MVP the following activities are required:

- Configure the environment
- Test Revision Tags
- Test Traffic Migration
- Deploy a public service

# Configure the environment



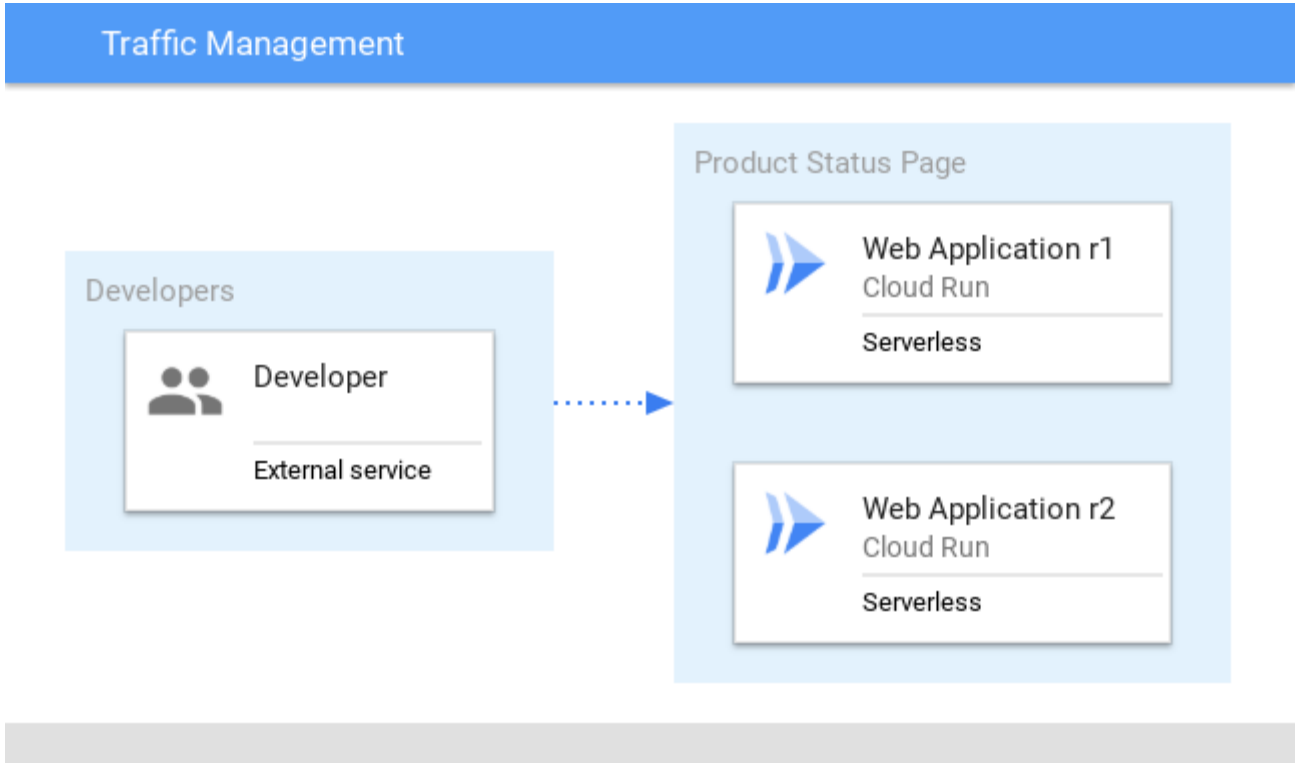
# Revision tags

Each new Cloud Run revision can be assigned a tag. Doing this allows access to a URL without serving traffic. An approach like this can be useful to handle the traffic profile across multiple revisions

The main uses cases for revision tags are shown in the following table:

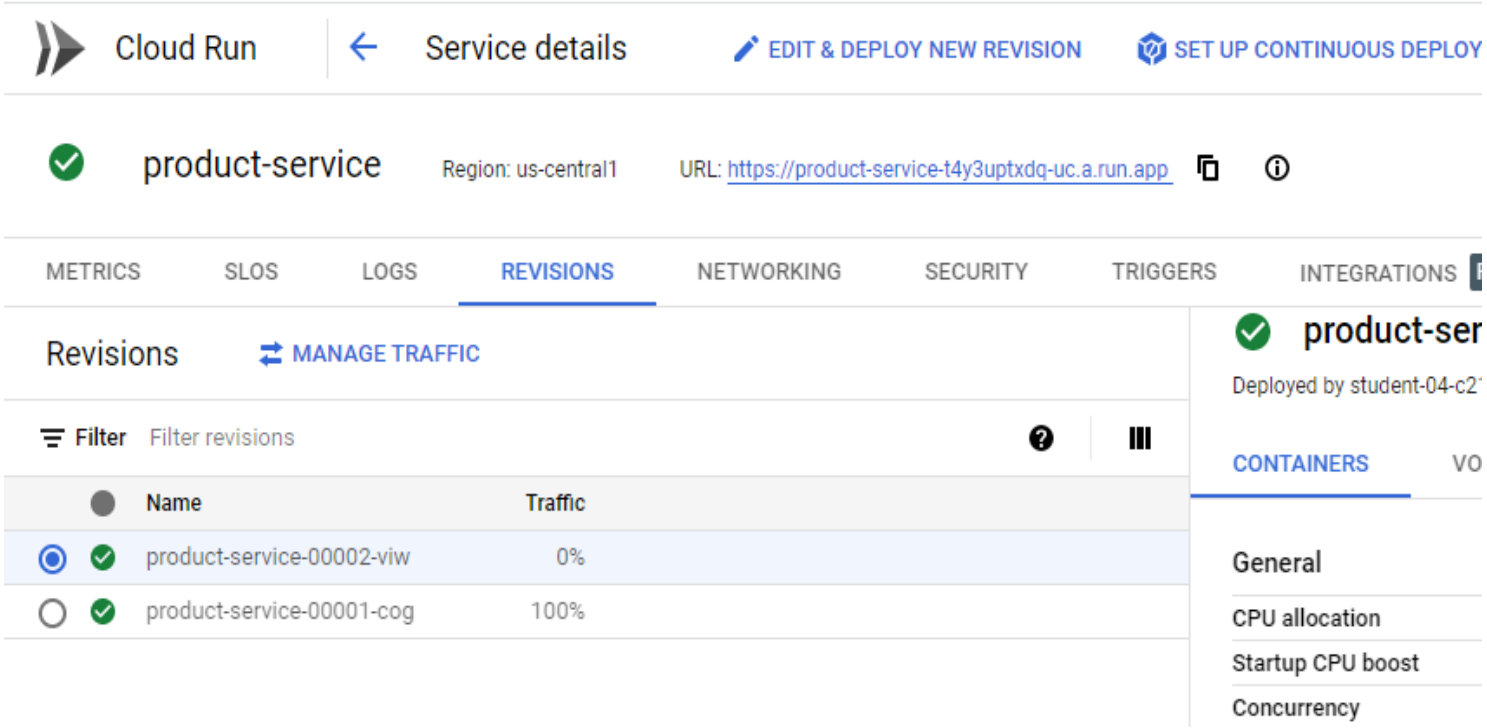
| Use Case | Description |
| --- | --- |
| Integration testing | Run containers revisions during the development |
| Tagged revision migration | Migrate traffic to a tagged revision |
| Tagged revision rollback | Rollback to prior version based on tagged revision |

# Revised architecture



# Integration testing

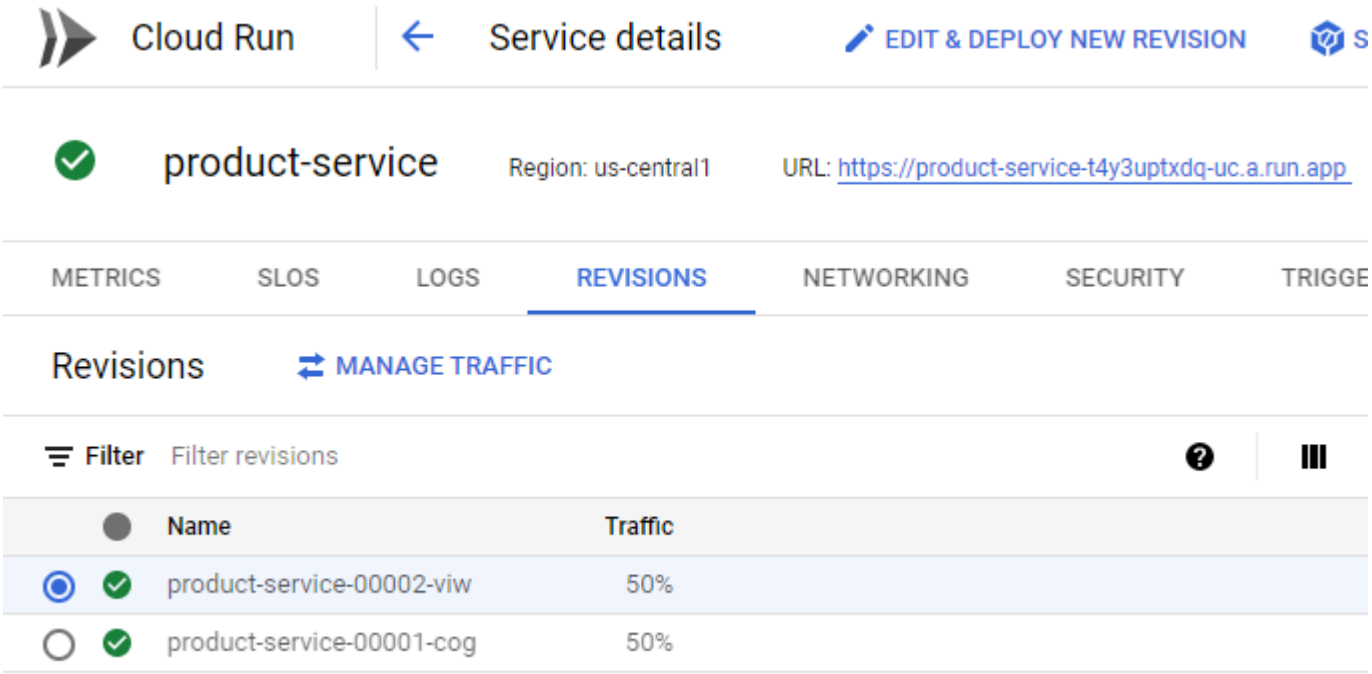
Cloud Run provides the ability to deploy a new revision with redirecting traffic. A deployment of this kind is useful for integration testing of components.




SmartInn

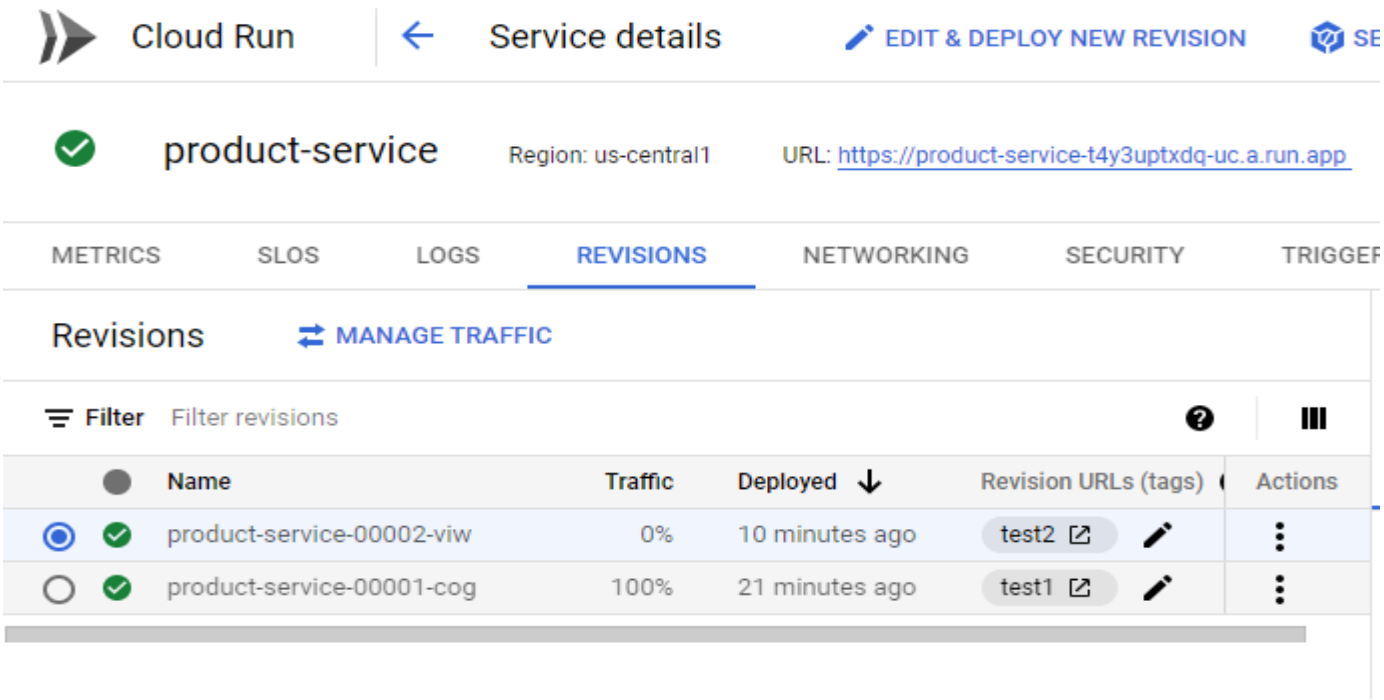
# Revision migration

Deployed revisions to share the traffic profile. Migrated 50% of the traffic to the revision tag



# Tagged revision rollback

In the event an issue is found, the traffic migration can be rolled back by resetting the percentage.



# Traffic migration

Migration of traffic provides a simple mechanism on which to direct communication to a deployed service.

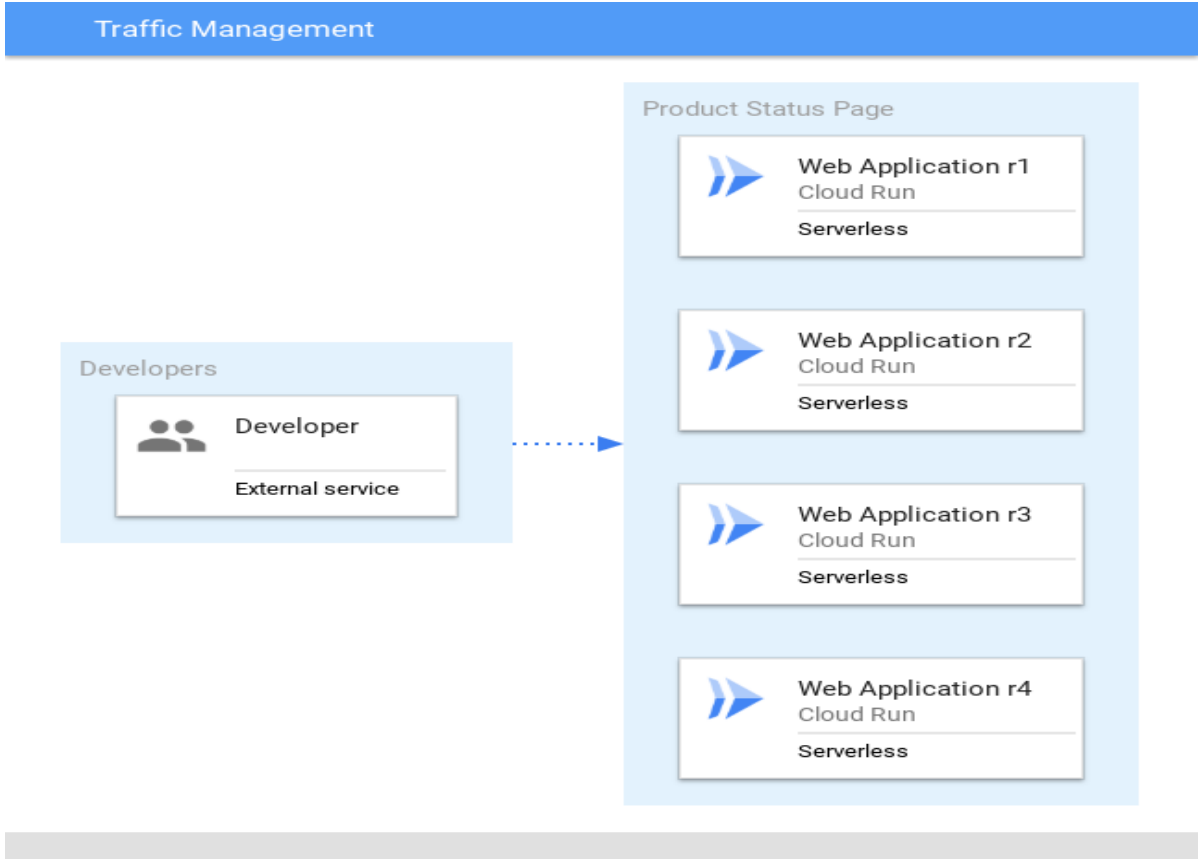
Cloud Run provides the ability to have multiple revisions to be deployed without a cost penalty.

Cloud Run only charges where traffic is handled by the service.

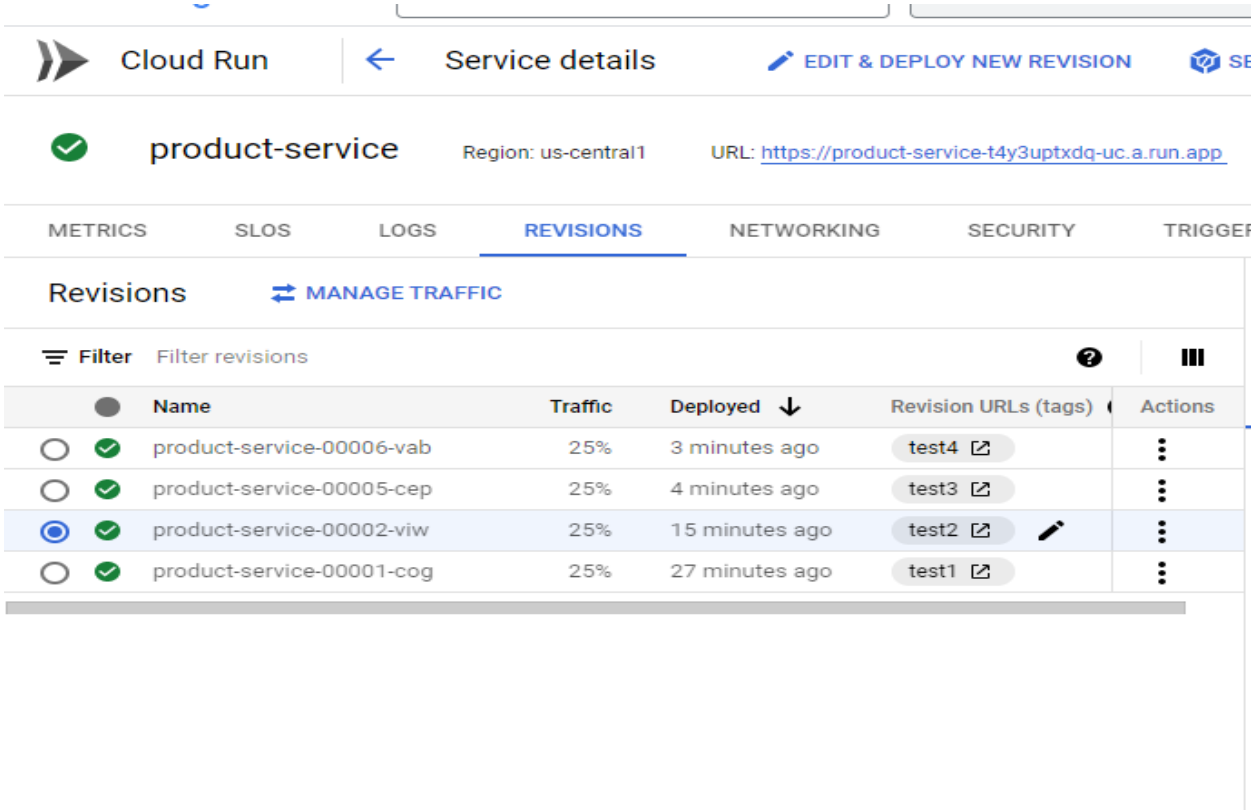
The main use cases for traffic migration are shown in the following table:

| Use Case | Description |
| --- | --- |
| Traffic migration | Enable traffic to be sent to the latest version of the deployed service |
| Traffic splitting | Perform a ratio traffic split between defined deployed services |
| Rollout migration | Deploy a service and gradually enable traffic at a predetermined time |

# Revised architecture

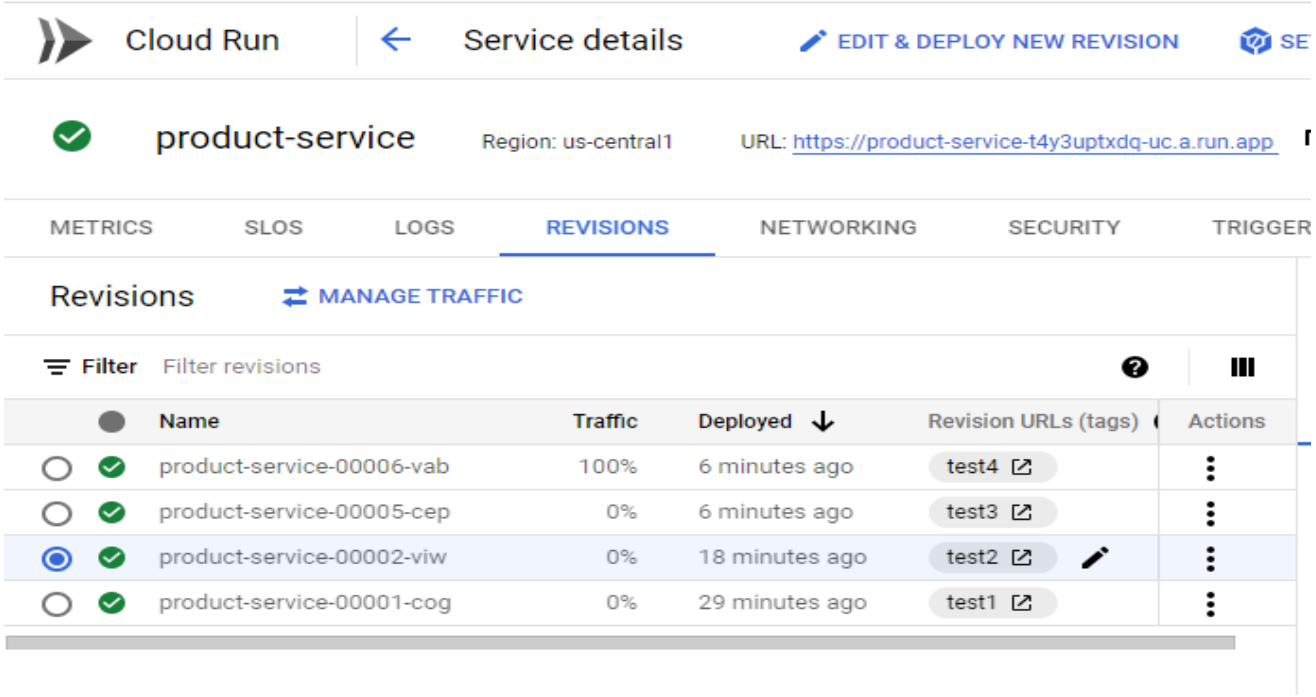


# Traffic migration - deploy a new version



# Traffic splitting - updated traffic between revisions

After reset, the service traffic profile now use the latest deployment as seen below:




The solution now take advantage of Cloud Run traffic management.

SmartInn

# Defining SLIs and SLOs

| SLO | SLI |
| --- | --- |
| Available 99.95% | Fraction of 200 vs 500 HTTP responses from API endpoint measured per month |
| 95% of requests will complete in under 200 ms | Time to last byte GET requests measured every 15 seconds aggregated per 5 minutes |
| Error rate of < 0.00001% | Upload errors measured as a percentage of bulk uploads per day by custom metric |
| Available 99.9% | Fraction of 200 vs 500 HTTP responses from API endpoint measured per month |
| 95% of queries will complete in under 10s | Time to last byte GET requests measured every 60 seconds aggregated per 10 minutes |

- The SLI describes what we are going to measure and how:

for example, the “Fraction of 200 vs 500 HTTP responses from API endpoint measured per month.” This example is a way of measuring availability.

- The SLO represents the goal we are trying to achieve for a given SLI.

For example, “Available 99.95%” of the time.”

digital Fingers

“Customer centric and reliable infrastructure design”.

**THANK YOU**