**Situational overview**

Here, I helped the development team at SmartInn investigate Cloud Run revisions. The team would like to explore how revision management can be incorporated into their existing development workflow.

**Requirements gathering**

The operations team at SmartInn would like to create a status page for their services without introducing additional complexity to their existing systems. Experimentation with Serverless products has led them to select Cloud Run due to the wealth of features supported.

Defining the service priorities

The operations team at SmartInn are keen to define a solution that can be implemented quickly. After discussions with the engineering team, they discovered that serverless may be a good option. To move forward, a series of meetings are held with stakeholders to ascertain the key priorities. The results of which are shown below:

|  |  |
| --- | --- |
| **Ref** | **User Story** |
| 1 | As a product lead, I want to ensure the website remains responsive, so customers face minimal wait times. |
| 2 | As a developer lead, I want to increase the velocity of service deployments. |
| 3 | 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. |

The team leads follow up the meetings and agree the following high level tasks would indicate the project requirements have been met:

|  |  |
| --- | --- |
| **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 |

In consideration of the requirements, the development team decide to look in to the following:

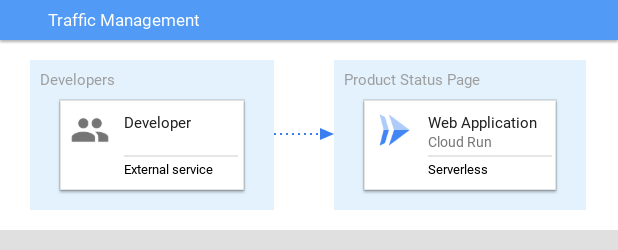
* Traffic Migration
* Revision Tags

**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."* | check mark |
| Traffic Migration | "Cloud Run allows you to specify which revisions should receive traffic and to specify traffic percentages that are received by a revision.." | check mark |

The following high level architecture diagram summaries the minimal viable product they wish to investigate.

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.

**Developing a minimal viable product (MVP)**

SmartInn has a backend product status service that they would like integrated with Cloud Run. To build an MVP the following activities are required:

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

## Revision tags

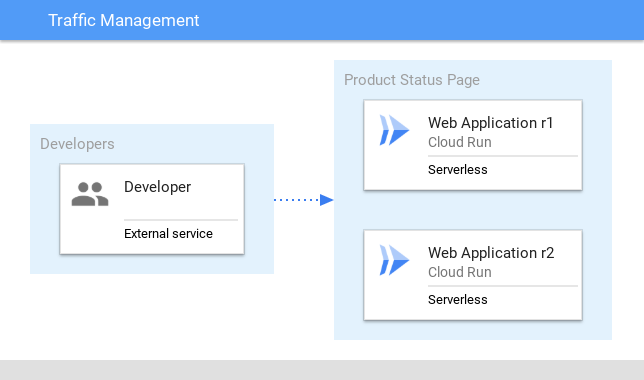
Each new Cloud Run revision can be assigned a tag. Doing this allows access to a URL without serving traffic. An approach like this can be useful to handle the traffic profile across multiple revisions.

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

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

In this section deploy a new revision and learn how to control the traffic to direct to the correct destination.

### 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.

1. Deploy a new tagged revision (test2) with redirection of traffic:

### Revision migration

There are now two versions available. However only one revision is serving traffic. Alter the deployed revisions to share the traffic profile.

1. Migrate 50% of the traffic to the revision tag test2:

### Tagged revision rollback

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

1. Migrate the distributed traffic back to the test1 service:

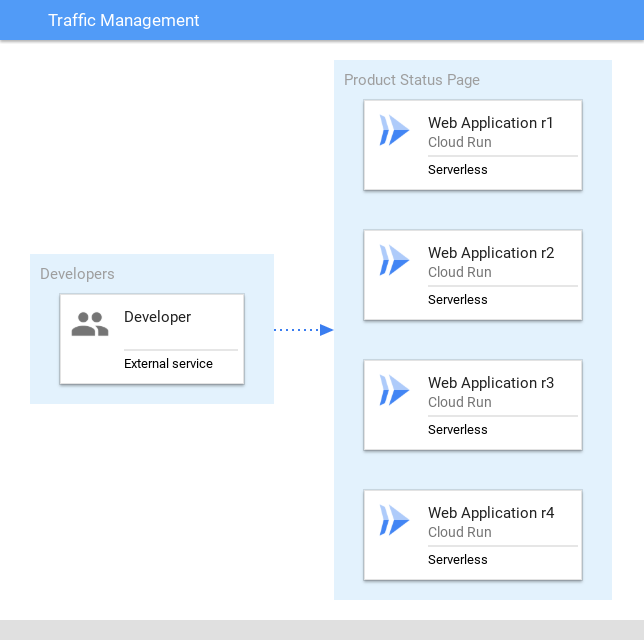
**Traffic migration**

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

The main uses 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



In this section multiple revisions will be deployed. Learn how to split traffic to a specific host.

Traffic migration - deploy a new version

When splitting traffic between two or more revisions, a comma separated list can be used. The list represents the revisions deployed.

1. Deploy a new tagged revision (test3) with redirection of traffic:
2. Deploy a new tagged revision (test4) with redirection of traffic:

### Traffic splitting - update traffic between revisions

1. Reset the service traffic profile to use the latest deployment: