

Execute your migration

After [planning your migration](/migration-center/docs/migration-planning-overview) (/migration-center/docs/migration-planning-overview), you can continue to the execution phase. This and the following documents provide you with a synthesis of the methods and tools you can use to execute your migration.

Before you start

- Complete the first wave of [discovery](/migration-center/docs/start-asset-discovery) (/migration-center/docs/start-asset-discovery), [assessment](/migration-center/docs/stratozone-assessment-overview) (/migration-center/docs/stratozone-assessment-overview), and [migration planning](/migration-center/docs/migration-planning-overview) (/migration-center/docs/migration-planning-overview).
- Complete the [foundation design](/foundation-toolkit) (/foundation-toolkit) and [landing zone design](/architecture/landing-zones) (/architecture/landing-zones).

Initial migration preparation

The **cloud migration project** is the major organizational effort that drives the migration of your workloads to Google Cloud.

Each migration project is divided into **waves**. A wave is a group of applications that share common characteristics or interdependencies, as identified by the workload discovery and assessment. Standalone applications and databases are typically good candidates for a first migration wave given their low external dependencies. On the other hand, workloads with significant interdependencies would constitute a complex migration wave that requires additional planning. In this case, you need to refine the migration plan to review the business impact of interdependencies, and remove the blockers that potentially prevent the migration.

Workloads within a migration wave are divided into **move groups** and migrated to Google Cloud in **sprints**. A move group is a group of infrastructure resources and workloads that you need to migrate together—these can be part of the same application or a group of applications that are interdependent.

During every sprint, you need to perform the following actions:

PRX155

1:23:cv-00108

- Prepare and integrate the tools required for migration.
- Develop a sprint plan.
- Execute the sprint plan.

Migration process and methodology

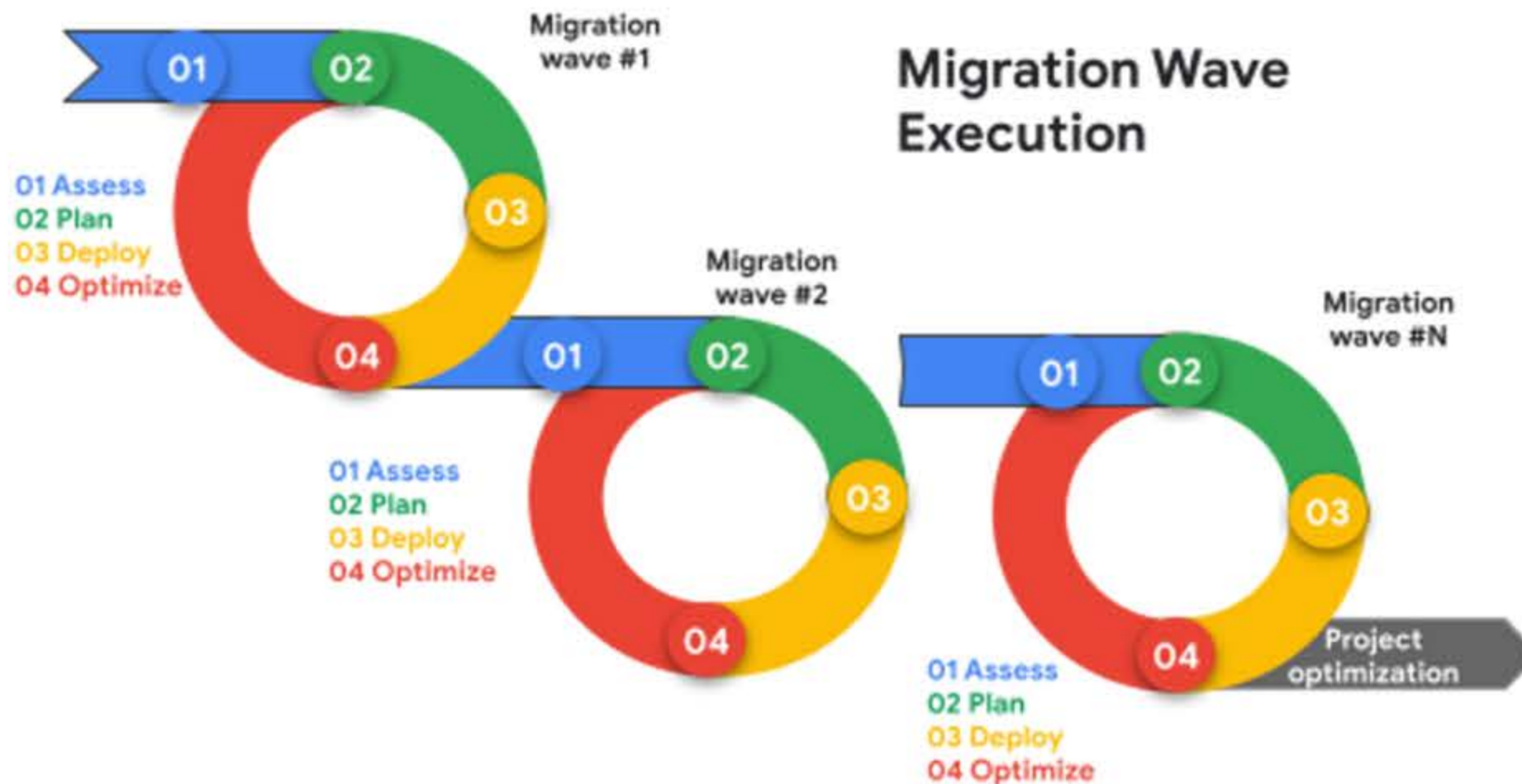
Sprint planning and playbook

In the sprint plan, define how you are going to execute the migration waves categorized into a sprint. Create a playbook by addressing the following building blocks.

#	Item	Description
0	Migration tools architecture	Architecture of tools constituting a migration factory (tools for ongoing assessment, wave plan refinement, workload-specific migration, build, testing, deployment, and monitoring)
1	Migration checklist	Checklist for use before and during the migration sprint
2	Inventory	List of workloads that will be migrated to Google Cloud
3	Sprint runbook	Execution guidelines for migrating each workload
4	Migration plan	Step-by-step migration plan (process) to be followed during the migration sprint
5	Network and security rules	List of all of the firewall rules for ingress and egress on Google Cloud DNS changes during migration to Google Cloud
6	Risks and mitigation	Possible risks during the migration sprint and mitigation steps
7	Testing and validation	Test plan to validate functional and non-functional requirements
8	Rollback plan	Rollback steps by workload
9	Team composition	Team composition and roster with contact details
10	Governance	RACI matrix of migration execution team, cadence and reporting, escalation resolution mechanisms

Migration execution

After you have completed the migration planning and preparation phase, this section describes how to perform repeatable migrations and validations to Google Cloud.



Assess

The first iteration of assessment happens during the migration planning phase, and produces data on dependencies between workloads and infrastructure components. You must continue to perform discovery and assessment throughout your cloud migration project to recalibrate and enrich data relating to the following aspects:

- Application and database mapping to infrastructure mapping (to identify all of the infrastructure and platform components for a business workload)
- Mapping between infrastructure to applications, databases, and services (to identify all of the business workloads attached to an infrastructure or platform component)
- Dependencies across business workloads
- Resource consumption by workloads
- Identification of any workloads that were not discovered in the initial wave of assessment
- Identification of new or changed landing zone requirements that have not been identified in the initial wave of assessment
- Identification of blocking issues that potentially prevent the migration

An ongoing assessment is critical to continuously calibrate and refine move group, identify and mitigate risks, and refine and optimize migration wave plans.

Plan

The planning phase in a migration wave aims at defining the final scope of the sprints within a wave, and consolidates the component-specific migration plans into a single plan. The outputs of this phase are the following:

- Move group(s) within the scope of the current sprint
- Migration sprint checklist
- Mitigations to remediate blocking issues
- Migrate, build, test, and deployment plan
- Rollback plan
- Scheduling of execution

Low-level detailed planning is critical for a successful deployment that follows.

Deploy

During the deployment phase, your migration team executes the migration plan and removes any critical issues. It is recommended to set up regular status meetings to track the execution plan. These status meetings, however, should not be used to troubleshoot issues. Instead, set up separate, dedicated sessions with respective technical experts.

The outputs of the deployment phase are the following:

- Migration plan updates (status per step, notes)
- Migration issue tracker updates
- Post-migration test results
- CMDB updates (if applicable)
- Migration results communication to stakeholders

If the deployment is unsuccessful, for example, if the migration plan fails, the tests fail or the fix is impossible within the defined migration timeframe, you need to execute the rollback plan.

It is advised to run application tests after the rollback and ensure any external changes that were part of the migration plan, such as upstream and downstream systems configuration, are rolled back as well.

Optimize

The optimization phase lets your project team regroup after completing the deployment phase to document the lessons learned and implement improvements for the following waves and sprints. For the scope that has already been migrated, the optimization phase can be used to resolve non-critical post-migration issues.

This phase is important as it enables continuous improvement throughout the project timeline.

The outputs of the phase are the following:

- Migration issue tracker updates
- Project knowledge base updates, if applicable

Migration tools

Automation tools play an important role in the migration lifecycle. During the execution phase of your migration, you need to create an architecture of automation tools based on a number of factors, such as the type of workloads to be migrated, geographical distribution and rollout strategy, and security requirements.

The following documents present several automation tools that address the following capabilities:

- [Migration and modernization tools](/migration-center/docs/migration-modernization-tools) (/migration-center/docs/migration-modernization-tools)
- [Build and CI/CD automation tools](/migration-center/docs/build-ci-cd-automation-tools) (/migration-center/docs/build-ci-cd-automation-tools)
- [Testing tools](/migration-center/docs/testing-tools) (/migration-center/docs/testing-tools)

What's next

- Learn more about the [tools for migration and modernization](#)

(/migration-center/docs/migration-modernization-tools).

- Learn more about [Google Cloud migration](/products/cloud-migration) (/products/cloud-migration).

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (https://developers.google.com/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2025-08-07 UTC.