**<Your company logo>**

**<Your company name>**

Application prioritization runbook  
for AWS large migrations



Template provided by the

AWS Large Migration Tiger Team

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved. See [AWS Site Terms](https://aws.amazon.com/terms/).

Table of Contents

[Overview 3](#_Toc92189710)

[Stage 1: Initialize 4](#_Toc92189711)

[Migration information and strategy 4](#_Toc92189712)

[Business and technical drivers 4](#_Toc92189713)

[Migration strategies 5](#_Toc92189714)

[Migration patterns 6](#_Toc92189715)

[Application prioritization 7](#_Toc92189716)

[Application complexity scoring criteria 7](#_Toc92189717)

[Application prioritization process 8](#_Toc92189718)

[Application prioritization rules 11](#_Toc92189719)

[Application deep dive 12](#_Toc92189720)

[Application workshop 12](#_Toc92189721)

[Application mapping 16](#_Toc92189722)

[Target application state 19](#_Toc92189723)

[Stage 2: Implement 20](#_Toc92189724)

[Prioritize applications 20](#_Toc92189725)

[Perform application deep dive 21](#_Toc92189726)

[Runbook revisions 22](#_Toc92189727)

[Contributors 22](#_Toc92189728)

# Overview

The objective of this document is to provide a data-driven process for portfolio assessment, including how to prioritize applications in the migration and perform deep dives to better understand the complexity of migrating the application. This runbook contains the information, rules, and processes used to repeat application prioritization tasks in an efficient and consistent manner.

For information about how to set up, use, and maintain this runbook, see the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

#### How to use this runbook template

Instructions for customizing this template are contained in blue boxes, such as this one. When you are finished customizing the template, we recommend deleting the blue boxes.

You should modify this template as needed to meet the requirements and use case for your large migration. You can add, modify, or remove steps to incorporate your project-specific processes and information. This template includes the following features:

* Examples – Examples are highlighted in gray and demonstrate how to use some tables. We recommend deleting the examples once you are familiar with the content item.
* Fields – Fields are highlighted in yellow, and you should enter information custom to your environment or use case in these fields. Once you edit a field, it reverts to the normal text color.

For information about how to customize and use this runbook, see the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html). The playbook contains detailed, step-by-step guidance for identifying the information and processes outlined in this template.

# Stage 1: Initialize

## Migration information and strategy

### Business and technical drivers

Update this section to reflect the migration drivers for your use case. When you are finished, remove the examples in the table. For examples of common drivers, see *Identify the business and technical drivers* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

The following are the business and technical drivers that must be considered when selecting migration strategies and mapping applications to migration patterns.

|  |  |  |
| --- | --- | --- |
| **ID** | **Type (business or technical)** | **Driver** |
| Example | Business | The data center contract is expiring at the end of the year. |
| Example | Technical | Your hardware or software is close to the end of its lifecycle, and you need to refresh it because the vendor no longer supports it. |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |

### Migration strategies

Update this section to reflect the how the applications are distributed to each migration strategy. For more information, see *Validate the migration strategies* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

The following is a high-level summary of how the application workload is distributed among the seven migration strategies.

|  |  |
| --- | --- |
| **Strategy** | **Percent of application portfolio** |
| Retire |  |
| Retain |  |
| Rehost |  |
| Relocate |  |
| Repurchase |  |
| Replatform |  |
| Refactor or re-architect |  |

### Migration patterns

Identify the migration patterns that will be used during your migration. When you are finished, remove the examples in the table. For more information and examples of common migration patterns, see *Validate the migration patterns* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

The following are the migration patterns identified for this migration.

|  |  |  |
| --- | --- | --- |
| **ID** | **Strategy** | **Pattern** |
| Example | Rehost | Rehost to Amazon EC2 using Application Migration Service or CEMF |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |

## Application prioritization

### Application complexity scoring criteria

Update this section to reflect the location of your scoring criteria. You might choose to use the provided score sheet template or to define criteria within your discovery tool.

For examples of common scoring criteria, see *About complexity scoring criteria* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

*Complexity scoring* is used to assess the difficulty of migrating an application, and it is a useful tool when prioritizing applications and planning waves. Refer to the business and technical criteria defined in the *Application complexity score sheet*. Ensure that you keep the scoring criteria current as you progress through the migration. The *Application complexity score sheet* is stored in the following location:

<file path or URL>

### Application prioritization process

This section contains three options for a complexity scoring process. Select one or more options, modify the process as needed for your use case, and then delete the options you will not to use.

For more information about the advantages and disadvantages of each process, see *Define the application prioritization process* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

#### Option 1: Manual complexity scoring

In this process, you use defined criteria to manually assess each application.

1. Open the *Application complexity score sheet template*, and create a new tab for this sprint. This document can be found in the following location:

<location>

1. In the **Application name** column, enter the name of the application you are scoring.
2. For the application, score each criterion from 1– 5, where one is the most complex and five is the least complex. Note the following as you score each application:
   1. For business criteria, we recommend that stakeholders complete this section unless you have full confidence that you have sufficient information to score. For reference, you can use readiness assessment data when scoring business criteria.
   2. For technical criteria, use the Scoring Guide sheet to help you score. The discovery results, CMDB, and asset inventory typically contain the data needed to score these criteria.
   3. If you don’t have the information needed to score, contact the person or team who might have it. For example, if you don’t have utilization data, contact the application owner or application monitoring team.
   4. If you cannot find the data needed to score a criterion, estimate the score. You can use the score from the closest application in terms of scale, architecture, size, and utilization. Alternatively, your cloud economics team might also be able to help provide data for an accurate estimate.
3. In the **Total Score** column, confirm that the score has been tallied correctly. A high score indicates low complexity, and a low score indicates high complexity.
4. Repeat steps 1–4 until all applications have a total complexity score. If an application is not scored, confirm it with the stakeholders. There are valid reasons that the application owners did not score the application, including intentionally excluding it or having insufficient data.
5. In the **Priority** column, enter a priority value for each application as follows:
   1. Deprioritize low-scoring applications, which are more complex.
   2. Prioritize high-scoring applications, which are less complex.
   3. For applications with the same complexity score, assign the same priority value. During the application deep dive, you might uncover more information that helps you break the tie. This is also good indication that these applications can be in the same wave or run in parallel if they are in separate waves.
   4. Use the priority rules defined in the Application prioritization rules section to determine the final priority for each application.
6. In the score sheet, sort the applications by the **Priority** column. This is the order of priority that you can use in wave planning.
7. Save the score sheet.

#### Option 2: Application nomination

In this process, application owners nominate applications for migration, based on their experience and knowledge of the applications.

1. Ask the application owners to nominate applications that can be easily migrated to the cloud. They should base this on their experience and knowledge of the application. A partial list of fewer than 10 applications is adequate as long as there are enough applications for 2–4 waves.
2. Open the *Application complexity score sheet template*, and create a new tab for this sprint. This document can be found in the following location:

<location>

1. In the **Application name** column, enter the names of the nominated applications.
2. In the **Priority** column, enter a priority for each application as follows:
   1. Prioritize low-complexity applications. When evaluating complexity, consider the migration strategy, the number of application dependencies, and the number of servers.
   2. Prioritize applications that don’t require much storage.
   3. Prioritize applications that have small a number of users.
   4. Prioritize applications that are not critical to the business.
   5. Use the priority rules defined in the Application prioritization rules section to determine the final priority for each application.
3. In the score sheet, sort the applications by the **Priority** column. This is the order of priority that you can use in wave planning.
4. Save the score sheet.

#### Option 3: Discovery tool

<Discovery tool name> has a feature for automated complexity scoring or application prioritization. This feature can be used after a minimum of <number> days of data discovery is complete.

1. Define the complexity scoring criteria and weights in <discovery tool name> as follows:
   1. <Insert instructions for your discovery tool>
2. Ensure that you have <number> or more days of discovery data available.
3. Generate a report in <discovery tool name> as follows:
   1. <Insert instructions for your discovery tool>

### Application prioritization rules

Update this section with your application prioritization rules. When you are finished, remove the example in the table. For more information and examples, see *Define the application prioritization rules* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

The following are the application prioritization rules for this migration.

|  |  |
| --- | --- |
| **Priority** | **Rule** |
| Example | Applications in the New York data center should have higher priority than applications in the Texas data center. |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |

## Application deep dive

### Application workshop

An *application workshop* is an efficient approach to an application deep dive. Using this process, you collaborate with the stakeholders, application owners, and the migration team to assess and analyze the application. The goal is to clearly understand the current state of the application, including its architecture, business purpose, dependencies, and environment.

#### Application workshop expected outcomes

Update this section with the expected outcomes of your application workshops. When you are finished, remove the example in the table. For more information and examples, see *Identify the expected outcomes* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

The following is the standard information that you should expect to collect in the application workshop. You might need to customize this table for each workshop, based on the information needed for the target application.

|  |  |
| --- | --- |
| **Priority** | **Expected outcome of application workshop** |
| Example | The application owner questionnaire is complete, and all key questions are answered. |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |

#### Application workshop rules

Update this section with the rules that govern your workshops. When you are finished, remove the example in the table. For more information and examples, see *Define the application workshop rules* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

The following are the rules that govern your workshop. Common rules include the workshop length, tools that may be needed in the workshop, and any scheduling considerations or deadlines that need to be considered.

|  |  |
| --- | --- |
| **Priority** | **Rule** |
| Example | Workshops should be scheduled for a maximum of 2 hours per session on Tuesdays and Thursdays. |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |

#### Application workshop process

This section includes a standard process for preparing, conducting, and finalizing an application workshop. Modify this process as appropriate for your use case. For more information and examples, see *Define the application workshop process* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

##### Prepare for the workshop

1. Establish an agenda for the workshop. This serves as a guide during the workshop to help keep the meeting on topic.
2. Copy the standard expected outcomes from the Application workshop expected outcomes section of this runbook, and then add expected outcomes that are specific to this application. Ask yourself the following questions when customizing the expected outcomes:
   1. **What information do you need for application prioritization?** Examples include the current application architecture and application source infrastructure information, such as CPU, disk, RAM, number of servers, location, and storage size.
   2. **What information do you need for wave planning?** Examples include application-to-application dependencies, application size, environment, and tier.
   3. **What information do you need for migration?** Examples include the target environment information, such as the AWS account, subnet, and AWS services.
   4. **Of the information identified in the previous questions, what can only be collected in a workshop?** Some of the application information can be collected from other sources, such as a CMDB or a discovery tool, but some information can be collected only in the workshop.
3. Identify the participants you need in the workshop, such as the application owners, a security representative, a support representative, or a DevOps representative.
4. Identify the appropriate participants from the portfolio and migration teams. These representatives should have knowledge of and experience with the application, based on their respective roles. Having the appropriate representatives helps the conversation flow smoothly during the workshop and minimizes the number of issues that must be resolved in subsequent meetings or as follow-up action items.
5. Identify any requirements, tools, or applications necessary to conduct the workshop. Examples include internet access or applications to record the meeting. To ensure the meeting is productive and efficient, install any tools or applications in advance of the workshop.
6. Collect any information about the application that you might need during the workshop. The following are some common artifacts that can help you prepare for the workshop and drive the discussion efficiently:
   1. Current reference architecture
   2. Application logical architecture
   3. Application playbook or runbook
   4. Application user manual
7. Schedule the workshop at least 7 days in advance, and include the following on the meeting invitation:
   1. Agenda
   2. Expected outcomes
   3. Workshop rules

##### Conduct the workshop

1. In the workshop, limit discussion to the content in the agenda and focus on the expected outcomes.
2. At the end of the workshop, ensure that you have a clear understanding of the application and have met all of the expected outcomes.

##### Finalize the workshop outcomes

1. If necessary, reprioritize the application based on the results of the workshop.
2. If time allows, draft the future state of the application based on the information you gathered during the workshop.
3. Update the Application workshop expected outcomes and Application workshop rules sections of this runbook with any new topics, issues, or questions that arose during the workshop.

### Application mapping

*Application mapping* is the process of assigning each application to a migration pattern.

#### Application mapping rules

Update this section with your application mapping rules. When you are finished, remove the example in the table. For more information and sample rules, see *Define the application mapping rules* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

The following are the application mapping rules for this migration.

|  |  |
| --- | --- |
| **Priority** | **Rule** |
| Example | Using your utilization data or monitoring tools, identify whether the application is a zombie or idle application. If the application is a zombie or idle application, choose *Pattern 8: Retire the application*, and then shut down the servers in the application stack. |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |

#### Application mapping process

This section includes a standard process for mapping applications to one or more migration patterns. Modify this process as appropriate for your use case. For more information and examples, see *Define the application mapping process* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

1. Request the discovery data for this application from the discovery team. Use the provided data when assessing the application.
2. Evaluate the business and technical drivers that apply to this application. For a comprehensive list of your migration drivers, see Business and technical drivers in this runbook. The business and technical drivers may lead you to select more than one migration pattern for this application. For more information, see *Define the application mapping rules* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).
3. Evaluate whether this application is an n-tier (or *distributed*) application. For these applications, you might need to select different migration patterns for different tiers.
4. Evaluate the application or application tier against the rules defined in this runbook, in the Application mapping rules section. Proceed through each rule, in order, until you find a matching migration pattern for the application. Ensure that the selected pattern meets the requirements of any relevant business or technical drivers.
5. Review the identified pattern with the application owner and ask them to verify the selection.
6. Record the application name, the identified migration pattern, and the applicable drivers in this runbook, in the Application mapping table.
7. Repeat this process for the next application.

#### Application mapping table

Record the results of the application mapping process in this table. When you are finished, remove the example in the table. For examples of application mapping table entries, see *Define the application mapping process* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

As you complete the application mapping process, record the application, pattern, and relevant drivers in this table. Ensure that you keep this table current as you progress through the migration.

|  |  |  |  |
| --- | --- | --- | --- |
| **Application name** | **Pattern ID** | **Pattern name** | **Business and technical drivers addressed** |
| Corporate website (example) | 1 | Rehost to Amazon EC2 using Application Migration Service or CEMF | * Exit data center * Reduce operational cost |
|  |  |  |  |
|  |  |  |  |

### Target application state

Update the criteria in this section to reflect when you want the portfolio team to document the target state of the application.

The *target state* is how the application operates in the target cloud environment after the migration.

#### When to define the target application state

Defining the target state of an application might not be necessary if the application is standalone, non-critical, or low-complexity, and it might not be necessary for low-complexity migration strategies. Complete this process if the:

* The migration strategy is replatform or refactor.
* The application is critical.
* The application has a low-complexity score.
* The application has multiple tiers.

#### Application target state attributes

Update this section to reflect the location of where you want users to document the application’s target state. You might choose to use the provided *Application target state worksheet* template, develop your own worksheet, or list the attributes and results in this runbook. For examples of standard attributes, see *Define the application target state attributes* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

Refer to the attributes defined in the *Application target state worksheet*.

#### Application target state process

This section includes a standard process for documenting the target state of an application. Modify this process as appropriate for your use case. For more information and examples, see *Define the application target state* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

1. Open the *Application target state worksheet*.
2. Duplicate an existing worksheet in order to start a new one for your application.
3. Name the duplicated worksheet tab with the application name.
4. Evaluate the application for each attribute in the worksheet, and then update the values.
5. Save the worksheet.

# Stage 2: Implement

## Prioritize applications

Update this standard process with changes to meet the needs of your use case and environment. For more information, see *Step 3: Finalize the application prioritization process* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

The following is the overall process for prioritizing applications in the implementation stage (stage 2) of a large migration. You perform this process multiple times in the migration, once for each sprint. When following this process, refer to the process sub-sections for more information.

1. Open a complete list of applications that will be migrated in your project. This could come from a discovery tool or a configuration management database (CMDB).
2. Determine the subset of applications that you will prioritize. We recommend that you select 2–3 times the number of applications or servers targeted for the wave. For example, if the wave will have 50 applications or servers, you should prioritize 100–150 applications or servers.

**Note:** There are multiple ways to select which applications to prioritize. The general rule is you should select low-complexity and non-critical applications first. For example, you can filter the list using application environment and choose applications in development or test environments first. You can also sort applications by the number of servers and select the applications that have the fewest number of servers.

1. For each application, complete the Application prioritization process in this runbook.
2. In your progress tracking sheet for portfolio assessment, update the following fields:
   1. In the **Application name** column, enter the name of each application.
   2. In the **Application score** column, enter the application priority score for each application.
   3. In the **Application prioritization** column, enter **Done**.
   4. In the **Application deep dive**, **Wave planning**, and **Data collection** columns, enter target completion dates for each application.
   5. In the **Ready for migration** column, enter **Pending**.

## Perform application deep dive

Update this standard process with changes to meet the needs of your use case and environment. For more information, see *Step 4: Finalize the application deep dive process* in the [Portfolio playbook for AWS large migrations](https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-portfolio-playbook/welcome.html).

The following is the overall process for performing application deep dives in the implementation stage (stage 2) of a large migration. You perform this process multiple times in the migration, once for each sprint. When following this process, refer to the process sub-sections for more information.

1. Collect a list of prioritized applications, based on the results of the Implementation stage: Prioritize applications process in this runbook.
2. For each application, send an application owner questionnaire to the owner of the application.
3. Plan and conduct an application workshop according to the instructions in Application workshop process.
4. Map each application to a migration pattern according to the instructions in Application mapping process.
5. If the application meets any of the criteria in When to define the target application state, define the application state according to the instructions in Application target state process.
6. Save the application data in the following location:  
   <file path or application>
7. In your progress tracking sheet, in the **Application deep dive** column, enter **Done**.

# Revisions

|  |  |
| --- | --- |
| Date | Change |
| Click or tap to enter a date. | Initial release |

# Contributors

The following individuals contributed to this runbook:

* <name>, <job title>