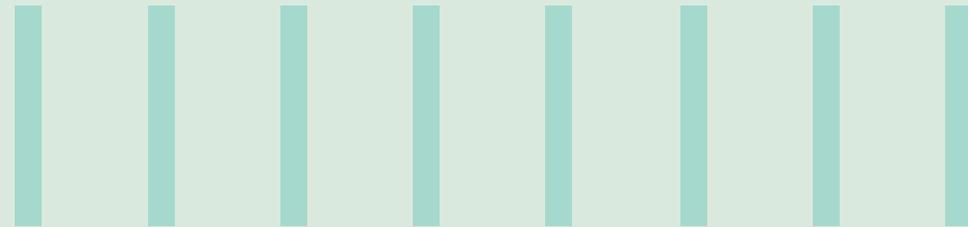




# HOW TO ENSURE A SUCCESSFUL SERVICE MAPPING IMPLEMENTATION

Key decisions and best practices



## Abstract

Service mapping is an aspirational tool for the modern IT department of an enterprise. It discovers underlying IT applications and their components and creates a map on how these connect to business services. Service mapping delivers several benefits like improved visibility, business continuity and compliance. However, implementing service maps is a complex process that requires several strategic decisions to be made based on the chosen approach. This white paper takes a look at these key decisions along the service mapping journey. It also discusses the advantages and disadvantages of various approaches with the aim of helping decision-makers make informed choices to ensure a smooth and successful service mapping implementation.

## Introduction

Service mapping is a process that establishes how underlying applications and IT infrastructure deliver business services. Using service maps makes it easier to identify the root cause of service outages and degradations and prioritize

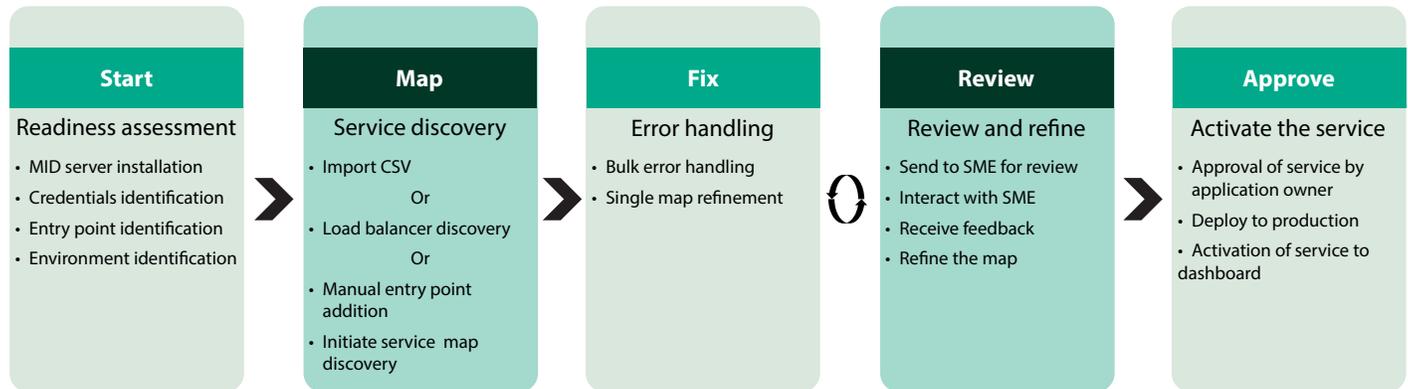
resolution of infrastructure issues based on business impact. It also plays an important role in business continuity, data center migration planning and compliance.

Service mapping applications discover foundational application services

and build a comprehensive map of all the components of an application service. However, due to its complexity, implementing service maps calls for strategic decisions to be made at the right time of the service mapping lifecycle.

## Service mapping lifecycle

Mapping a specific service to its infrastructure consists of five key stages:



Before determining the right implementation approach, some customer-specific decisions must be taken considering the different stages of the service mapping lifecycle. These decisions and their relevant stages are:

1. Environment identification during the 'Start' stage

2. Entry point identification during the 'Map' stage
3. Refining the service map during the 'Fix' stage
4. Maintaining the map during the 'Approve' stage

The following sections look at each stage-related decision and discuss several approaches along with their advantages and disadvantages. The aim of this exercise is to help organizations choose the optimal approach when implementing service mapping applications.



## Decision 1: Environment identification

Service mapping of an application can be done in non-production environments and, once approved, moved to production environments.

**Approach 1:** This process involves using a non-production ServiceNow instance to discover the non-production infrastructure of the service. Then, the solution is moved from the non-production instance to the production instance of ServiceNow. Once the non-production infrastructure of the service is discovered and approved, the production infrastructure of the service should be identified. This approach follows three steps:

**Step 1:** Non-production instance – Non-production infrastructure

**Step 2:** Non-production instance – Production infrastructure

**Step 3:** Production instance – Production infrastructure

### Advantages

- Most of the issues will be fixed before the production infrastructure is discovered
- The production ServiceNow instance will be more stable

### Disadvantages

- Connectivity from non-production instance to production infrastructure needs additional security approvals which would further delay the implementation
- Multiple ports must be opened from several servers to management, instrumentation and discovery (MID) servers
- Implementing the same solution in multiple environments takes additional time

**Approach 2:** This process involves a non-production ServiceNow instance that discovers the production infrastructure of the service. The solution is moved to the production instance of ServiceNow. After this, the production infrastructure of the service is discovered in the production

ServiceNow instance. The key steps here are:

**Step 1:** Non-production instance – Production infrastructure

**Step 2:** Production instance – Production infrastructure

### Advantages

- Service maps are created in lesser time as mapping is required for only one infrastructure environment

### Disadvantages

- Connectivity from non-production instance to production infrastructure needs additional security approvals which would further delay the implementation
- Customizations must be unit tested using the production infrastructure, which may lead to production issues

**Approach 3:** This process involves using the production instance of the ServiceNow to discover the production instance and non-production infrastructure. Once approved, the production infrastructure should be discovered. The key steps here are:

**Step 1:** Production instance – Non-production infrastructure

**Step 2:** Production instance – Production infrastructure

### Advantages

- Service maps can be created in lesser time

- The solution need not be migrated to a different environment

### Disadvantages

- Making changes in the production environment can affect the current discovery
- The production ServiceNow instance is less stable

**Approach 4:** This process involves using the non-production ServiceNow instance to discover the non-production infrastructure of the service. The solution is then moved to the production instance of ServiceNow. Then, the production infrastructure of the service is discovered in the production ServiceNow instance. The key steps here are:

**Step 1:** Non-production instance – Non-production infrastructure

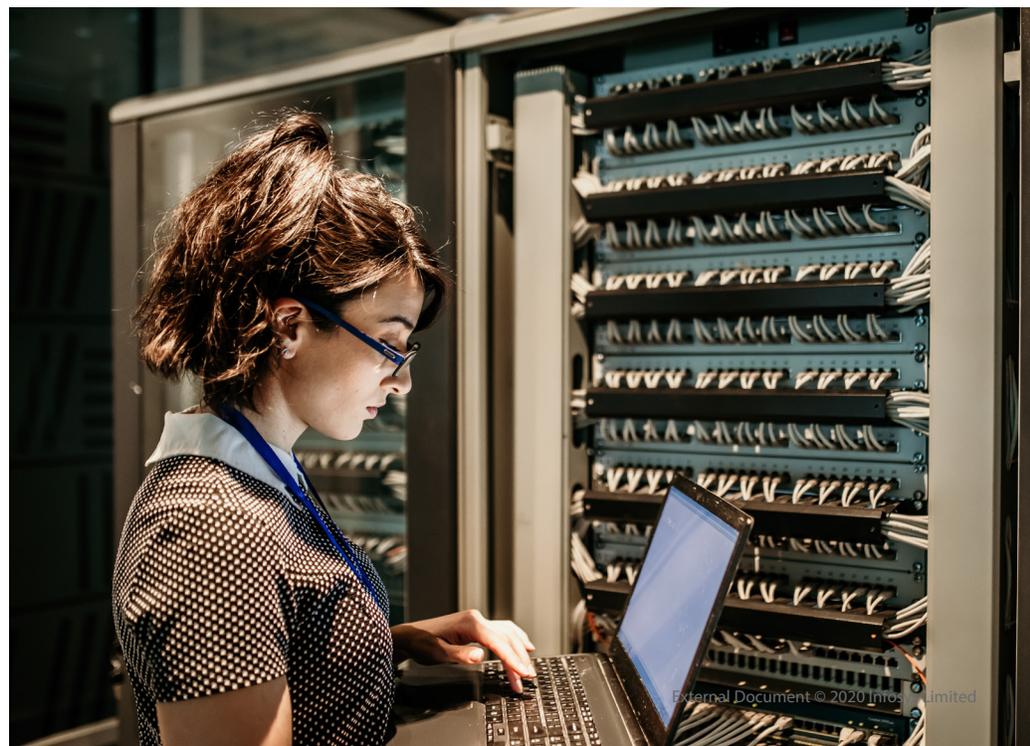
**Step 2:** Production instance – Production infrastructure

### Advantages

- As environments and instances are mapped appropriately, this approach aligns well with security considerations
- Customizations are tested in the non-production instance, making the production instance more stable

### Disadvantages

- New issues may be identified when changes are made to the production instance



## Decision 2: Entry point identification

The key components needed to begin service mapping are:

- a. Entry point
- b. Credentials
- c. MID server with ports enabled

Service mapping discovery begins with an entry point. An entry point is the first-tier application on a host configuration item (CI) where the end user receives the service. There are three approaches used to identify entry points, depending on how entry points are created or used:

- Sequential approach for entry points that are manually entered one at a time
- CSV import approach for entry points are imported from a comma separated variable (CSV) file, allowing for multiple services to be discovered in parallel
- Load balancer approach where entry points are used to run discovery against the management IP addresses of all load balancers in an enterprise. Here, service mapping will automatically provide a list of service candidates

**Sequential approach:** This approach is used for entry points that are manually entered one at a time. The key aspects of this approach are:

- The service map entry points must be created one after the other
- Issues and errors in each map are addressed

- Additional entry points will not be discovered

### Advantages

- Only the desired application infrastructure is discovered and mapped

### Disadvantages

- Requires greater manual effort
- Additional services are not discovered

In the case of bulk mapping, service mapping allows hundreds of services to be mapped parallelly. The two approaches for identifying entry points during bulk mapping are – CSV import approach and load balancer discovery approach.

**CSV import approach:** Entry points can be manually imported from a CSV file, allowing multiple services to be discovered in parallel. A standard CSV file can be downloaded and updated with the required entry points and imported back to ServiceNow.

### Advantages

- Multiple applications can be discovered once the entry points are imported
- Minimizes the potential of irrelevant application services

### Disadvantages

- Creating the entry points is still a manual process and needs knowledge about the application service
- The outcomes of bulk mapping are not precise

### Load balancer discovery approach:

Here, service mapping runs discovery against the management IP of load balancers and automatically provides a list of service candidates. The load balancer discovery approach is recommended for organizations with multiple applications to be mapped as it reduces the effort spent in identifying every entry point through automation. In this approach, service mapping performs the following tasks:

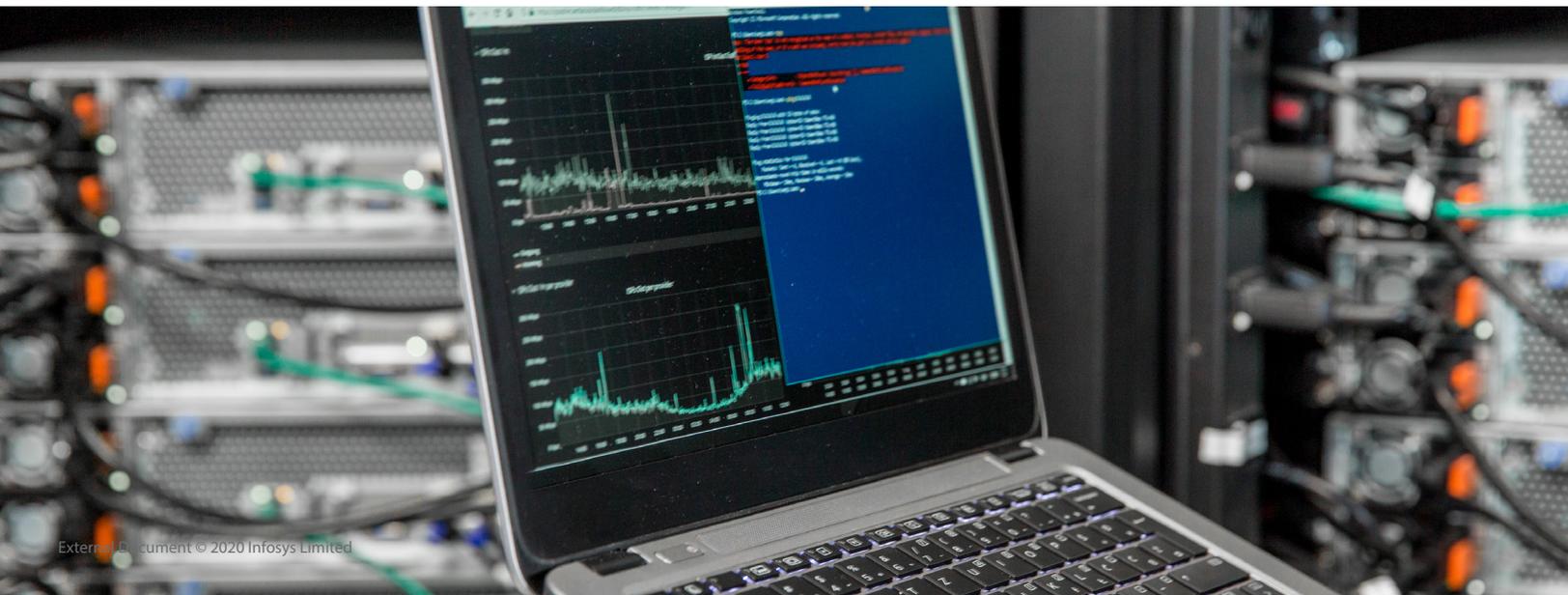
- When no credentials are available, it relies on traffic-based discovery by reducing traffic noise using machine learning
- Extracts entries directly from load balancers on the network
- Includes steps for prioritization, suggestions, task assignments, and better visibility into the overall picture

### Advantages

- Bulk mapping is an effective and quick way to create numerous application services
- Unlikely candidates will be filtered out
- Reduces discovery time
- Minimizes the number of potential irrelevant application services

### Disadvantages

- The results of bulk mapping are not precise
- False or duplicate applications may be created erroneously as the data from the load balancer is raw



### Decision 3: Refining the service map

The service mapping process consists of the following steps, which are further described below:

- Discovery of the service using patterns
- Discovery of the service using traffic
- Customizing patterns
- Fixing the service map
- Turning off discovery of the service using traffic
- Reviewing the service map
- Publishing the service map

Discovery through patterns is the recommended method to discover infrastructure and its inter-relationships. Nowadays, there are several out-of-the-box (OOTB) discovery patterns available. However, there are some scenarios where this approach may be unable to identify services.

In such cases, organizations can opt for traffic-based discovery. A caveat here, though, is that changes in the traffic will alter the service map. Hence, it is recommended to fix discovery patterns and use it for discovering the underlying infrastructure. Once the patterns have been fixed and the topology of the service has been refined, Infosys recommends turning off traffic-based discovery.

While most discovery patterns are suitable for infrastructure discovery, this may not be the case for application-related discovery. Sometimes, application teams do not provide access to OOTB commands and instead request different commands. In such cases, the discovery fails. The discovery pattern must then be fixed in order to successfully complete the discovery.

Once all the underlying infrastructure has been discovered, the service map gets populated automatically. In some cases, discovery may detect additional components that will need to be removed from the map. Similarly, some components

may be missing even after multiple fixes and may need to be created manually.

The next step is to send the service map for review and approval by the subject matter expert or the service owner.

Once approved, the service map will be published in ServiceNow.

In all cases, the above approach must be used to create a service map; there is no alternate approach. Published service

maps can be used along with the event management module to identify impact to the service during any infrastructure or network outage.

When traffic-based discovery continues to discover services without customized patterns, it can result in some negative impact like unstable service maps. This leads to greater effort being spent to maintain the map and keep it accurate.



## Decision 4: Maintaining the map

Service maps tend to change due to changes in the applications. To ensure that the service maps are updated, preconfigured discovery as well as custom schedules can be created.

**Discovery schedules:** Service mapping includes a preconfigured generic schedule for discovering all CIs of an application type. CIs of the application are derived from the configuration management database (CMDB) CI application table. The generic schedule triggers the discovery of all the organization's applications once daily. In addition to this, there is a preconfigured schedule for discovering load balancers derived from the CMDB CI load balancer service table.

Service mapping checks whether the device hosting the application CI exists in the CMDB. If not, service mapping triggers a discovery to detect the host. The information on the host is updated in the

CMDB during horizontal discovery.

Some CI types are more prone to changes and updates than others. It is easier to manage the load by adjusting the discovery schedule to match the nature of the CI type, thereby eliminating redundant stress on the infrastructure.

Discovery schedules can be created for different application CIs as listed below:

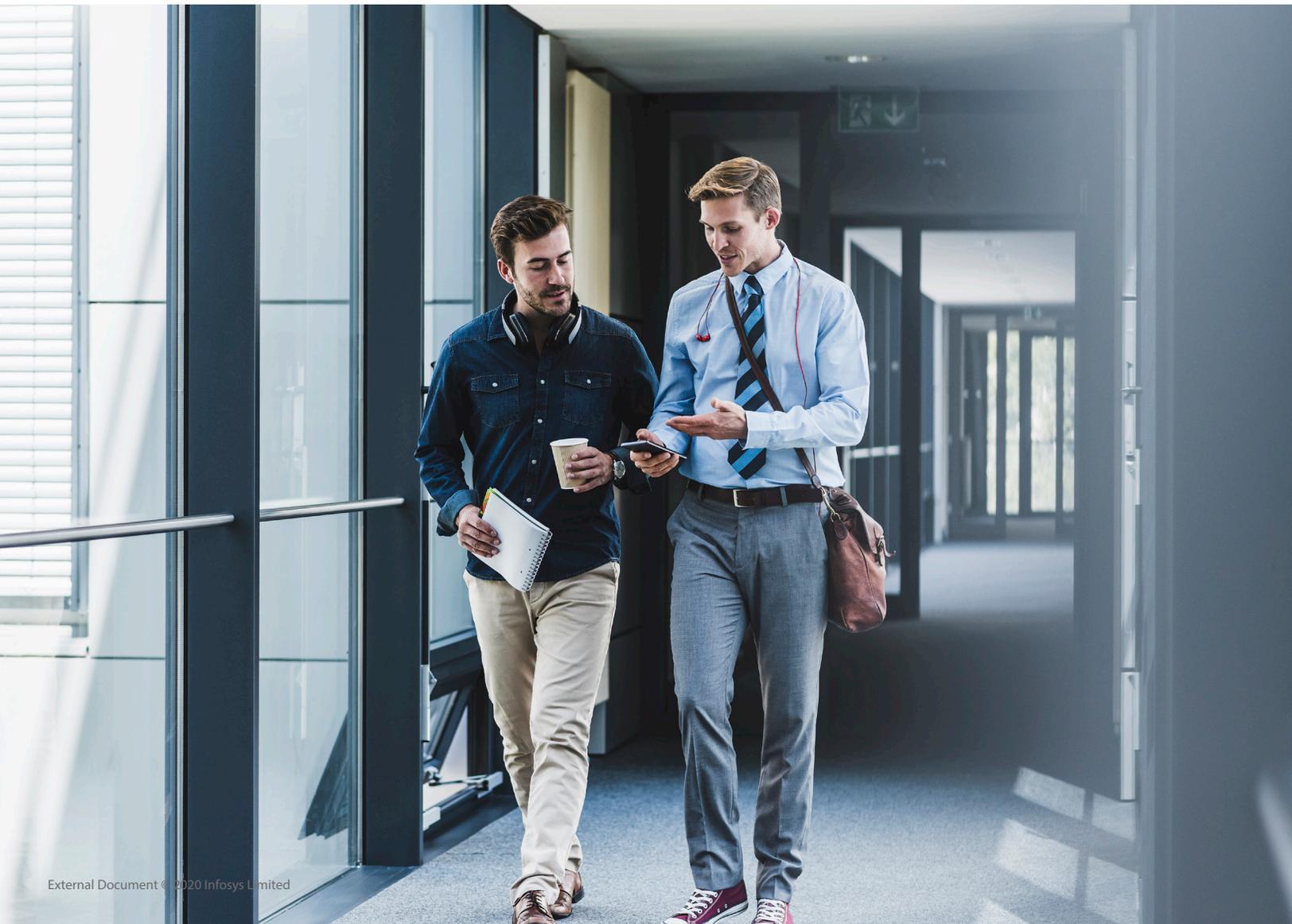
- For CI types – Service mapping will discover all the CIs that belong to a particular CI type
- For specific CIs – Service mapping will discover only the CIs specified for the schedule

**Being upgrade friendly:** When customizing OOTB patterns, ServiceNow automatically creates a copy of these patterns and does not delete the original version. While the original version keeps getting upgraded, the customized copy will serve as the active one. To extend the

pattern, a shared library (either existing or new) can be added to the discovery logic without changing the identification section. By extending the discovery patterns, additional information about the application service can be discovered.

**Checking and troubleshooting service maps:** Service mapping discovery patterns ensure that only the relevant CIs are discovered for each business service. It is important to regularly monitor critical service maps for transparency and accuracy. Some key monitoring tasks include:

- Hiding unwanted CIs
- Hiding topology cycles, i.e., connections returning to the same host
- Hiding traffic-based CIs
- Merging CI errors
- Setting map indicators to display related records





## Conclusion

Service mapping helps organizations better understand how IT infrastructure and components align with business services. When implementing a service map, there are some key decisions to be made at strategic points. Some of these decisions include how to identify environments and entry points and how to refine and maintain the service map. While each approach to these decisions has its own advantages and disadvantages, it is important to discuss these with the relevant stakeholders like application architects, security teams, product owners, and product sponsors before choosing the right approach. The above recommendations can act as a guide to facilitate key and timely decisions, a crucial differentiator for successful implementations.

## About the author



**Nagendra Swamy Paladi** – Senior Consultant, Infosys

Nagendra has over 13 years of experience working with Infosys clients across domains and geographies. His areas of expertise include architecture definition, evaluation, implementation of infrastructure and application monitoring, discovery, dependency mapping, service mapping, event management, and CMDB.

For more information, contact [askus@infosys.com](mailto:askus@infosys.com)



© 2020 Infosys Limited, Bengaluru, India. All Rights Reserved. Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.